MULTI-MODAL FAKE NEWS AND TAMPERED IMAGE DETECTION USING TRANSFORMER AND CNN-BASED ALGORITHMS

Research Project

MSc Data Analytics

Prashant Digambar Waghela

Student ID: x20207786

School of Computing

National College of Ireland

Supervisor: Prof. Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: PRASHANT DIGAMBAR WAGHELA

Student ID: x20207786

Programme: MSc. Data Analytics **Year:** 2021-22

Module: Research Project

Supervisor: Prof. Vladimir Milosavljevic

Submission Due Date: 15th August 2022

Project Title: MULTIMODAL FAKE NEWS AND TAMPERED IMAGE DETECTION USING TRANSFORMER AND CNN-BASED ALGORITHMS

Wordcount: 8892 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: PRASHANT DIGAMBAR WAGHELA

Date: 14th August, 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

MULTIMODAL FAKE NEWS AND TAMPERED IMAGE DETECTION USING TRANSFORMER AND CNN-BASED ALGORITHMS

Prashant Digambar Waghela

x20207786

Abstract

In this modern world, the use of computing devices and internet access has made it easier for any news article or post to spread among the masses, resulting in quick access to information but, sometimes these platforms are exploited to spread fake information. A customized multimodal algorithm that uses these news headlines to verify the authenticity and concurrently recognize if its corresponding image information is fabricated or not can reduce the spread of wrong information. The report provides a thorough literature review that helped in devising this multimodal technique for fake news classification. Moreover, the research contributes by developing three unique multimodal algorithms BERT+CNN, BERT+InceptionV3, and XML_RoBERTa+CNN that classify fake news text and related images simultaneously. The multimodal BERT+CNN model provided the best accuracy of 71% which was comparable to the unimodal BERT approach which achieved 72% accuracy. The study was crucial in understanding the impact of using multimodal text and visual features to classify fake news and the obtained results were analyzed to extract insights from the implemented multimodal technique.

Keywords: BERT+CNN, BERT+InceptionV3, and XML_RoBERTa+CNN

1 Introduction

The spread of false news has been rising substantially over the past decade and the major reasons include various social media platforms and news websites over the internet that can be easily accessed through mobiles and numerous computing devices. The posts and articles available on these platforms do assist people to access true information but, they often become a source that gets exploited by people to spread fake information to mislead a large population. These fake news can sometimes become so adverse that it can create major socio-economic and political chaos among people which can impact a nation's peace and harmony. As per the poll study led by an author at 'thejournal.ie'[1], in April 2022, the votes suggested that the three major reasons for the spread of fake news were social media platforms, individual internet users, and news organizations. These kinds of polls and research were critical in determining the issue of fake news that needs to be banished to reduce the spread of wrong information to some level. Moreover, a specific fact-checking survey among 22 participants was carried out by the European Broadcasting Union[2] to identify that 16 out of

[1] https://www.thejournal.ie/poll-stopping-misinformation-5731675-Apr2022/

[2] https://www.ebu.ch/media/fake-news

the 22 news journalists already implemented fact-checking initiatives to reduce the spread of fake information. However, such journalism initiatives if used in conjunction with machine learning algorithms could critically improve the identification of fake news. The major components of news or any social media posts include a news headline, its summary along with an image or video. A machine learning or a deep learning technique can be used to evaluate these data pieces to detect if the article posted is true or fake. This available information can be further simplified as text and image data which can be fed as inputs to these algorithms. However, the basic machine learning technique can only work on one type of data and hence, a unique methodology that can process image and text data simultaneously has been implemented in this study. A novel research question that merges text and image data processing for news classification was proposed and implemented in this study.

1.1 Research Question

How do multimodal Transformer-based text and CNN-based image classifier models detect fake news and fabricated photos simultaneously, thereby reducing the spread of misinformation?

1.2 Motivation & Aim of the Research Project

Fake news is a root cause for the spread of wrong information which creates misunderstanding among people. It can have a negative impact on society which can lead to disruption of peace and harmony. The fact-checking survey conducted by EBU[3] highlighted critical reasons for the spread of misinformation and it was identified that the majority motives to spread fake news included political elections or personal gains. These kinds of surveys and investigations served as an inspiration for the development of a Deep Learning Algorithm that can assist in lowering the issue of fake news identification. Every news article or social platform post consists of both written and visual data. A short list of the conducted research project's aim and objectives are written below.

- The use of an innovative multimodal algorithm was proposed and created that could handle the text data using a transformer-based algorithm and visual data using a CNN-based model concurrently and identify if the news is fake or true.
- The goal of the proposed model is to identify whether the addition of visual features enhances the classification performance if compared to the algorithms that only used textual parameters.

Hence, this research project contributes to the field of Data Analytics by creating a Transformer-based Deep Learning Technique that provides a two-factor verification based on the text and visual data of the news article.

2 Literature Review

The widespread internet usage for various purposes has provided everyone with quick connectivity and real-time data access. However, the major disadvantage is the spread of false knowledge that results in chaos and misinformation within the masses. The social media platform or the news agency posts uploaded on any website includes three important parameters; a news title, associated summary text, and a visual piece of data. These

[3] https://www.ebu.ch/media/fake-news

components consist of the majority of the critical information, and it can be put to use to evaluate if the news or the post is practically correct or not. A Deep Learning Technique that can compute the text and visual parameters at once can recognize the fake data more precisely. Multiple scholarly research papers were studied and identified to acknowledge various methodologies and algorithms that led to the proposal and implementation of a hybrid multimodal technique. The proposed and implemented research included 4 critical decision points and hence, the literature review is sub-divided accordingly.

2.1 Fake News Detection using Textual Data

The most important factor of any news or social media post is its main headline and related abstract information. The written data comprises lexical, sentimental, and semantic attributes that form the base of Natural Language Processing and it becomes applicable in news stories classification. This section cites research papers in which predictive models were implemented primarily on the features of the textual data. Based on the various benchmark methodologies, it is further divided into three sub-categories to come up with the best approach.

2.1.1 Supervised Machine Learning Techniques

In the initial days of the internet boom, when fake news detection was a newer problem, researchers tried to find solutions for it using different supervised machine learning methodologies. Reis, J.C., et al. (2019) aimed at detecting fake news stories from social media by proposing a new set of textual features. They used a bag of words, POS tagging, and text readability methods to extract various lexical, psycholinguistic, semantic, and subjectivity features from the text input. Moreover, numerous news source features such as bias, credibility, trustworthiness, news locations, and online platform engagement patterns were given as feature inputs for machine learning models. A total of 5 models were implemented on a benchmark BuzzFeed dataset with 2282 records and the XG Boost model provided the best results with an F1 score of 0.81. A similar but, more thorough machine learning approach was imbibed by Ahmad, I., et al. (2020) and they used four real-world news datasets from the World Wide Web. They proposed a unique text feature Linguistic Inquiry Word Count - LIWC and their primary aim was to identify various text patterns across datasets from 4 different domains. They implemented 4 machine learning, 6 ensemble learning methods, and 3 benchmark algorithms and identified that each domain had different critical text features i.e. it was difficult to train a single algorithm to perform well on the datasets from different domains. However, they achieved the best results with the Ensemble XG-Boost classifier and obtained an average accuracy of 94% across datasets. A nearly identical problem statement to Reis, J.C., et al. (2019) was researched by Khanam, Z., et al. (2020) wherein; they used a slightly bigger LIAR dataset with 12.8k records which consisted of factually verified news records from PolitiFact.com. They proposed using Python's sci-kit learn library to perform tokenization using count and TIFF vectorizer to extract textual data features. The lexical features such as word length, counts of adjectives, verbs, and nouns, and the average length of news articles were also evaluated to feed them as inputs for the machine learning algorithms. The best results were achieved by the Naïve Bayes method with an accuracy of approximately 70%.

The above-cited 3 research papers provided a brief understanding of the performance of various supervised machine learning approaches for the fake news detection problem. However, it was evident that these algorithms provided less accurate results when the dataset size and the individual article length increased. Hence, this approach was not considered in the implemented research project. The next sub-section focuses on understanding a unique and more accurate propagation-based technique using graph neural networks to classify fake news.

2.1.2 Propagation-Based Approach with Graph Neural Networks

The propagation-based Graph Neural Network – GNN approach exceeds the performance of the basic machine learning techniques because these ML algorithms lack the recognition and understanding of the political and socio-economic context. This ideology was identified by Monti, F., et al. (2019) when they proposed and implemented a novel propagation-based GNN technique. They highlighted the fact that fake and true news spread differently on any social platform and creates different graph patterns and recognized that the propagation-based method provided better resilience and classification performance against newer unseen data. A novel method using classic Convolutional Neural Network-CNN based GNN was developed to produce graph patterns using heterogeneous tweet features such as user content, user profile, user activity, their social graphs, and news propagation. The dataset was accumulated from 3 benchmark websites Snopes, BuzzFeed, and PolitiFact to create a custom factually verified dataset. The proposed method provided better results without relying on any textual parameters and an accuracy of 92% was achieved. A similar technique was implied by Han, Y., Karunasekeran, S., and Leckie, C. (2020) for fake news detection wherein, they tried to differentiate the propagation patterns of fake and true tweets without using the text information i.e. tweet content or tweet replies. They aimed at identifying if the GNN-based method achieved superior results without text parameters in comparison to the regular state-of-the-art context-based algorithms. They applied their algorithm on a custom FakeNewsNet dataset which comprised fake and true tweet records from the PolitiFact and GossipCop websites. They devised a unique feature matrix that included the number of followers, friends, lists, favourites, statuses, and tweet timestamps to create their heterogeneous graph patterns. In order to handle adversarial attacks against new unseen data, novel Gradient Episodic Memory - GEM and Elastic Weight Consolidation - EWC methods were incorporated within their proposed GNN model. The individual weighted F1 scores of 0.80 and 0.82 were achieved for the PolitiFact and GossipCop data respectively. However, Ren, Y., et al. (2020) used a slightly different approach to the fake news detection framework by proposing a unique Adversarial Active Learning based Heterogeneous Graph Neural Networks – AA-HGNN. They created a novel hierarchical attention-based algorithm to produce node representations where the model learned using both the textual and structural graph data points simultaneously. Two benchmark datasets PolitiFact and BuzzFeed were used to train and test the developed AA-HGNN algorithm. Their primary aim was to create an algorithm that could be trained on minimum data and can still produce decent results for the unseen data points. Hence, they trained their AA-HGNN model on only 20% of their data and still were able to achieve accuracy metrics of 61% for the PolitiFact dataset and 73% for the BuzzFeed dataset.

In the three cited research studies, the GNN was applied to the data from the social media domain, and the node representations of the tweet accounts were required to form those CNN-based heterogeneous graph patterns. Hence, it becomes challenging to create these graph patterns for the non-tweet category data and it can lead to a difficult task of recognizing news URLs and tracking them over Twitter handles. Thus, its duplication is demanding in other domains. The other major limitation of this technique is the forgetting issue which is visible when the GNNs are trained on newer data to obtain continuous adaptation and learning. The research project, thus, eliminated the use of the propagation technique, as the used dataset consisted of news from different websites. The following subsection focuses on the Deep Learning algorithms that outperform the previously cited techniques.

2.1.3 Deep Learning and Transformer-Based Methodologies

The fake news detection on large-scale datasets became an easier task to accomplish when the Deep Neural Networks - DNN was used for this problem statement. This was initially recognized by Hiramath, C.K. and Deshpande, G.C., (2019) who proposed a novel DNN algorithm for fake news classification. The model was implemented on the benchmark LIAR dataset wherein the text data was initially pre-processed and cleaned so that the data can be fed as an input to the model. A stemming technique was applied to replace the adjective and adverb forms of words with their base noun form. Moreover, the non-significant stop words that do not add any specific meaning to a sentence were also removed from the input text. A total of 4 machine learning algorithms and a Deep Neural Network were applied to the dataset and a comparison study was implemented. Some unique graphical insights suggested that the DNN algorithm utilized maximum system memory during implementation and it also had the lowest execution time of 400 milliseconds. The highest accuracy of 91% was also achieved by the DNN algorithm. A more refined comparison study between Machine Learning and Deep Learning algorithms was implemented by Saleh, H., Alharbi, A. and Alsamhi, S.H., (2021) who aimed to achieve an optimal high-performance model. The developed Optimized CNN algorithm was compared with RNN and LSTM techniques where each of these methods was trained and tested against 4 standard datasets. The glove embedding technique was used for the DL algorithm. Additionally, Grid Search and hyper-opt optimization methods were implemented to identify optimal model parameters. The OP-CNN algorithm provided the best accuracy of 97%, 95%, 54%, and 99% for the four datasets respectively. The most advanced and novel architecture that outperformed the majority of the state-of-the-art Deep Learning techniques is discussed in the upcoming research papers. The Deep Learning algorithms used unidirectional word embedding techniques whereas the Transformer-based models use a bidirectional approach with self-attention layers for more precise results.

A Transformer architecture was created by Vaswani, A., et al., (2017) from Google. Their out-of-the-box algorithm included a set of self-attention modules that achieved higher computational speed using parallel input techniques. This algorithm was initially developed to perform text translation tasks which were further enhanced to create models like BERT and RoBERTa to handle text classification problem statements. The same pre-trained BERT model was used by Rodríguez, Á.I. and Iglesias, L.L., (2019) to solve the fake news detection task. The BERT pre-processor and encoders were used to obtain valid inputs and to fine-tune the algorithm. A total of 20k news records were used and they were collected from multiple reliable news websites to train and test the BERT model. In comparison to the applied LSTM and CNN-based algorithms, the BERT architecture outperformed them and obtained an F1 score of 0.97 with an accuracy of 98%. The Transformer algorithm was uniquely enhanced by Kaliyar, R.K., Goswami, A. and Narang, P., (2021) when they implemented a FakeBERT methodology that obtained an accuracy of 98% on the exact dataset as the last research paper. The FakeBERT included 3 additional parallel convolutional blocks which took the same-sized word embeddings as inputs. The output of these convolutional pieces was further merged using a 1D-convolutional block. The model was optimized by selecting hyperparameters like batch size, loss function, and the number of epochs. From the papers cited in this sub-section, it is clearly understood that the DNN and the BERT algorithms perform well in terms of handling news classification tasks. These techniques outperform the basic machine learning and propagation-based approaches. Hence, the optimum fake news detection using text features is best obtained by a Transformer-based methodology and it was evident from the implemented BERT model in this research project.

2.2 Fake Image Detection using Image Data

The research also consisted of the image data as an additional input other than its corresponding news text to add a 2-factor classification. Thus, this section cites numerous papers that use Deep learning and pre-trained algorithms for image classification tasks.

2.2.1 CNN Based Approach

A CNN-based image classifier was implemented by AlShariah, N.M., Khader, A. and Saudagar, J., (2019) to detect tampered images that spread on social media platforms. A total of 560 real images were used to create 1260 filtered or fake images manually to feed them as inputs to the CNN algorithm. The model was developed using multiple convolutional and max pooling layers along with a ReLU activation function which was lastly followed by a SoftMax layer to perform image classification. The proposed model achieved an accuracy of 65%. The data size used was small and thus, the results could not be directly correlated to the larger image datasets. A similar problem statement was implemented by Qi, P., et al. (2019) on a larger Weibo dataset to detect fake images. They proposed a unique approach to extract pixel and frequency domain features separately and pass them through parallel convolution blocks hence, they named this new architecture Multimodal Visual Neural Network – MVNN. The accuracy obtained by this algorithm was 84% which was better than the state-of-the-art models at that time. Diallo, B., et al. (2020) aimed at detecting modified or tampered images and did so by implementing a custom CNN architecture. They used 3 image augmentation techniques wherein they created a compressed and a resized version of the original images and passed all of them as inputs to the convolution layer. The architecture developed had 4 convolution layers with different filter sizes in combination with the Max Pooling layer. They trained this algorithm on the Dresden dataset which consisted of 13k images and obtained a weighted accuracy of 60%. From the three cited studies, it was understood that the CNN-based approach provided decent results in fake image detection tasks and hence, it was incorporated into this project.

2.2.2 Pre-trained Algorithms for Image Classification

To reduce the model-building process, pre-trained algorithms were developed and one such model VGG16 was used by Kuznetsov, A., (2019) to work on spliced image forgery detection task. Image patches of size 40x40 were generated from the original pictures and given as input to this pre-trained model. The VGG16 was customized by adding a dropout layer to avoid data overfitting. Moreover, a 3x3 kernel size and a pooling layer of 2x2 were added throughout the convolution layers. A CASIA dataset was selected and image augmentation using compression techniques was used to create 80% and 90% compressed images from their corresponding original pictures. The accuracy metric obtained for the original and compressed images were 96%, 67%, and 66% respectively which were significantly better than the already existing results. A similar problem statement was taken by Zhong, J.L. and Pun, C.M., (2019) wherein a CNN-based Dense-Inception Net algorithm was developed. The image features were extracted using Pyramid Feature Extractor, Feature Correlation Matching, and Hierarchical Post Processing. The initial 1098 forged images from the dataset were augmented using rotation and scaling techniques to obtain a massive data count of 131k forged images to train and test the algorithm. The feature extractions were used to create pixel level and image level information and the model was trained on both of them individually. An F1 score of 0.92 and 0.78 was achieved for the image and pixel level data correspondingly. Nguyen, T.T. and Huynh, K.T., (2021) proposed a solution for forged image detection using a pre-trained InceptionV3 algorithm to detect fake images with spliced nature. A benchmark Columbia Uncompressed Image dataset was used for the research and an accuracy of 93% was obtained. They proposed a unique approach to locating forged areas within the images to reduce the processing time by only passing the forged image data to the algorithm and this was achieved by using an edge extraction and image enhancement-based extraction technique. Moreover, the high-level undetectable splices were handled using contrast and enhance layers. The papers studied in this sub-section helped to realize the importance of pre-trained models that could save time and still produce benchmark results. Since Inception-based pre-trained algorithms could handle large datasets easily so, the InceptionV3 was specifically selected to be applied in the research project.

2.3 Multimodal Approach for Fake News Detection

The scope of the implemented research project included both image and text data and hence, it was identified that the multi-modal approach bridged the gap between individual text and image classifier algorithms to be combined and used for a single classification task. A unique multimodal TI-CNN architecture based on this ideal was developed by Yang, Y., et al. (2018) to detect fake news by utilizing both the text and visual features. The dataset used had 20k records and various NLTK steps like special character removal, stop words removal and specific cognitive and psychological perspective words were studied and evaluated to understand the use of deceptive language within a text statement. Moreover, sentiment analysis was applied to the text data and image resizing was implemented to standardize the image classifier input for the multi-modal algorithm. The best F1 score of 0.92 was achieved by the TI-CNN model. Jindal, S., et al. (2019) addressed the issue of scarcity of benchmark multimodal datasets. Hence, they created a combined NewsBag dataset using various benchmark datasets which resulted in a final data file with 215k records. Moreover, the target class imbalance was handled using the bag of words technique. The created dataset was tested against all the benchmark algorithms and the best results were obtained by the Multimodal Variational Auto Encoder and the Fake Detector algorithms wherein the accuracy of 71% and 70% was achieved for this new dataset. Similarly, a distinctive multimodal dataset was created using Weibo, MediaEval, Reddit, and many Indian news websites data by Sharma, D.K. and Garg, S., (2021) wherein they also proposed and created a combinatory LSTM and VGG16 model to handle text and image features simultaneously for the fake news identification. They used a different approach to train and test their algorithm on two different image sizes 224x224 and 32x32 to recognize the impact of image size on the overall multimodal algorithm accuracy. It was observed that the accuracy metric dropped from 74% to 66% when the standard input image size was reduced to 32x32. All three papers within this sub-section were critical in identifying the influence of multimodal algorithms to enhance the performance of fake news detection. There was still a scope to add a transformer-based text classifier to verify if it could elevate the results in comparison to the cited multimodal techniques. Therefore, a hybrid Transformer and CNN-based multimodal algorithm were proposed and implemented in this research project.

3 Process Flow Technique

The project aimed at developing a multimodal algorithm that could detect fake news using both the text and image data features simultaneously. The problem statement did not include any business perspective and it only consisted of knowledge extraction to implement a classification task on a comparatively large dataset. Hence, the research project is aligned with a Knowledge Discovery Database approach and the critical process-flow steps are carefully mapped within this 5-step methodology and it is as shown in Fig. 1.

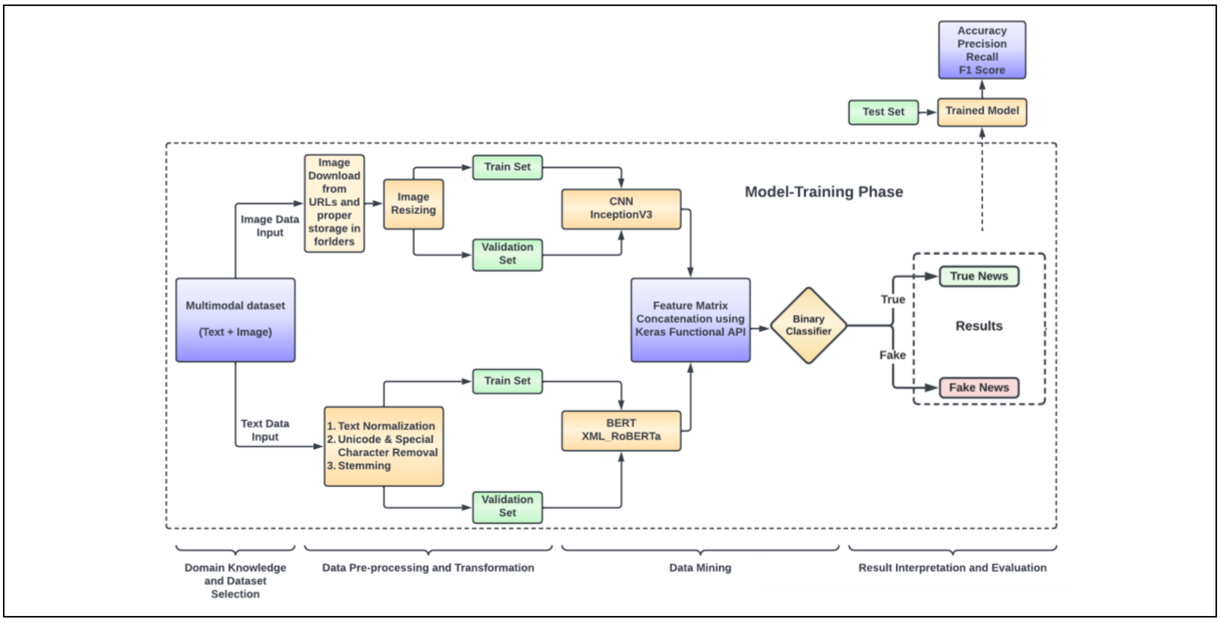

Fig. 1 KDD-mapped Process Flow

3.1 Domain Knowledge and Dataset Selection (Step 1 KDD)

From Fig. 1, it can be observed that the multimodal dataset selection comes under the domain knowledge and the data selection step of the KDD methodology. Fake news detection is an ever-growing issue in this technology dependant world. It is humanly impossible to factually validate every news or social media post to identify its authenticity and hence if these fact-checking initiatives are supported with the prediction algorithm, then, it could help in detecting if the news is true or false. Most of the research in this domain is based on individual text and image classifiers and hence, this project aimed at combining a transformer-based text and CNN-based image classifier to recognize fake news. The project had a unique approach of using a multimodal news dataset which included the news headline along with its corresponding multimedia image. Table 1 provides the details about the most important columns of the dataset. The raw multimodal dataset included 56k news records collected by Sharma, D.K. and Garg, S., (2021) and it was collated from multiple news websites like Reddit and PolitiFact. Proper permission from the dataset creator was taken via email to access the dataset provided via a private GitHub link. An Ethics Declaration Form for the same was submitted at the start of this Semester.

Table 1. Dataset Description

Columns	Data Type	Description
image_number	Numeric	Unique identifier for every news article present in the dataset.
news_title	String	A string with a news headline.
image_path	JPG	Image related to the corresponding news article.
target	Categorical	Target column with True and Fake labels as 1's and 0's.

3.1.1 Data Insights and Exploration

The multimodal dataset used in this project consisted of news headlines along with their corresponding image URLs. Various dataset insights were identified to extract features from them for better data analysis. The first visualization in any classification task is to identify the distribution of the target column and hence, a pie chart was plotted to identify this information.

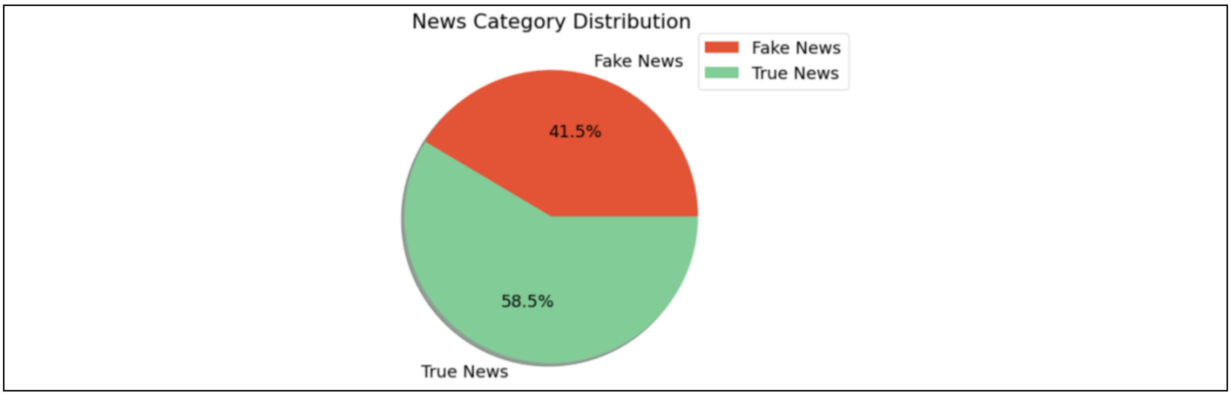

Fig. 2 Pie-Chart Distribution of Target Column

Fig. 2 represents a pie chart that gives the percentage distribution of the true and fake news within this dataset. It can be observed that, like in any real-world scenario, the amount of fake news would be comparatively low as compared to the true news and it remains true for this dataset as well wherein there are 41.5% of fake news and 58.5% of true news. It can be assumed that the dataset aligns with real-world scenarios and it is fairly balanced hence, no data augmentation technique is required to increase the count of fake news.

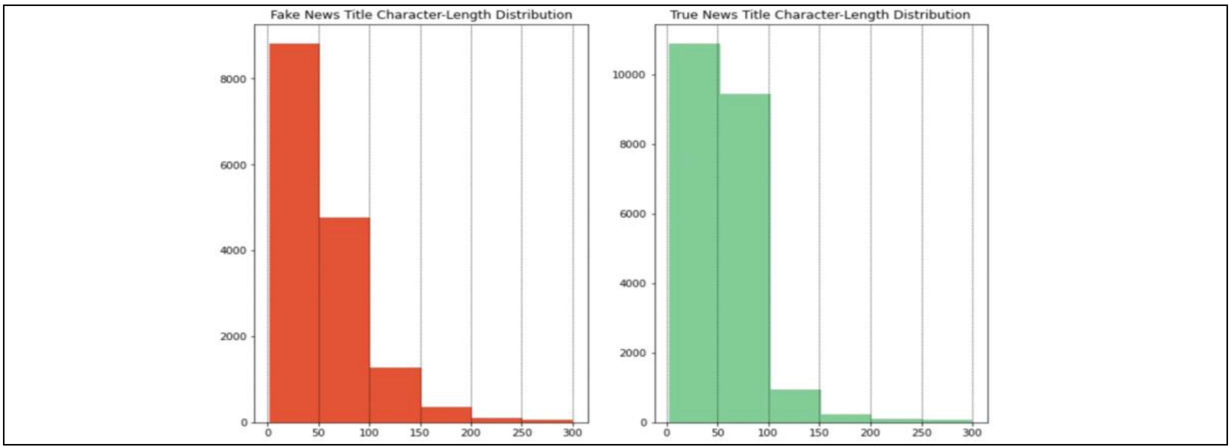

Fig. 3 News Title Character Length Distribution

Since there is a presence of text data it is necessary to identify if there are any notable differences in the character length distribution for the 'news_title' column for both the true and fake news. From Fig. 3 it can be observed that the true news had most of its news title within the 100-character length which could be roughly approximated to an average of 14 to 25 words per news title.

3.2 Multi-modal Data Pre-Processing & Transformation

The data pre-processing and cleaning are divided into two main parts because of their multimodal nature. However, before performing the individual text and image data pre-processing it was necessary to clean the combined data frame. The raw data file had numerous insignificant columns and hence, a total of 11 columns were eliminated from the data file. Moreover, there was a presence of N/A values in the initial dataset, and such rows were completely removed using the 'dropna()' functionality of the Pandas data frame. This resulted in the reduction of the data count from 56k to approximately 38k records.

3.2.1 Text Data Pre-Processing

The text data in the final data frame included the news title which could not be directly fed to any deep learning model. Hence, multiple text pre-processing and cleaning techniques like text normalization, special character removal, and stemming had to be implemented before the data could be given as input to the Transformer-based text classifier. Their detailed explanation is listed below.

1. **Text Normalization** – Any text classifier algorithm considers the uppercase and lowercase alphabets differently and hence, text normalization into lowercase was applied to the entire 'news_title' column to standardize the text input for the models.
2. **Unicode Removal** – The 'news_title' column had multiple data points wherein special characters were used. If these characters are not removed, then the algorithm converts them into Unicode. Hence, to avoid this scenario all the special characters within the text column were removed using a regex function that only included the alphabets 'a-z' and the numerals '0-9' which eventually reduced the processing time of the model.
3. **Stop words Removal** – A set of words in the English dictionary does not possess any significant meaning and hence, the text classifier algorithm cannot extract meaningful information from it for the classification task. Therefore, using the NLTK library this set of stop words was downloaded and then removed from the entire text column.
4. **Stemming** – The same word within the text is sometimes used in its adjective, adverb, past, or future form which could all be replaced by a single noun form of that word. This technique is called stemming and it was applied to the complete 'news_title' column. This method also helped in reducing the implementation timeframe of the final algorithm.

3.2.2 Image Data Pre-Processing

1. **Image Data Filtering** – The image data available in the dataset was in the form of URLs and hence, a custom Python script in the local Jupyter Notebook environment was developed to access those links and download the images and store them in appropriate folders within the local system. This step was necessary to identify the non-responsive websites and eliminate those types of data records out of the final usable data frame.
2. **Image Resizing** – The downloaded images had varied dimensions and they had to be standardized hence, the image size of 200 x 200 was selected so that, they could be given as input to the CNN-based image classifier. This Python script was also created and run on the Jupyter Notebook because the Google Colab platform could not access the local system path.

Lastly, the final resized images from the local system were permanently moved to a more secure Google Drive location because the further data splitting and data mining script were migrated on the Google Colab Pro Notebook. This step was initiated because of the high GPU requirements of the proposed multimodal algorithm. The image data filtering step resulted in more elimination of records and the final usable dataset count came down to 37,113.

3.2.3 Creation of NPZ Array

The above data processing steps provided the final useable data frame which required only one additional step before data splitting and the application of the chosen data mining techniques. An NPZ array had to be created which is a file extension by NumPy that gives storage of array data type with the help of gzip compression. Since the implemented algorithm was multimodal, it was necessary to store and access the text data, image path, and the target column class for every record through a single index. It provided an advantage of accessing the written and visual information along with its target column label simultaneously using NPZ indexing. The NPZ files for every record were created and stored in the secure Google Drive Location so that they could be accessed during the data transformation, splitting, and model implementation tasks.

3.2.4 Data Transformation

The data transformation for the text input of this project was implemented using a direct pre-processor link of the BERT and XML_RoBERTa pre-trained models which are later defined in the model building steps. It directly performed the necessary tokenization step to produce a vector representation of the text data. It is an important step because it provides a standardized vector input for the further encoding process within the transformer architecture. Moreover, the image data was adequate for model training and hence, no augmentation techniques were applied to increase their counts.

3.2.5 Data Splitting

After performing all the pre-processing steps, the data was ready to be utilized for the data mining techniques. Hence, it had to be split into training, validation, and test sets. The created NPZ array was used in this step to split data based on NPZ indexes. The complete data of 37,113 was split into 28,000 train records, 6000 validation records, and 3,113 test records. The final ratio of this splitting came out to be 75%, 16%, and 9% respectively.

3.3 Data Mining (Step 4 KDD)

The data mining step provides a brief explanation of the working of the proposed multimodal architecture. However, before moving to architecture, it is necessary to understand the design framework of this project. Fig. 4 represents a 3-tier design framework that consists of a data, application, and presentation layer.

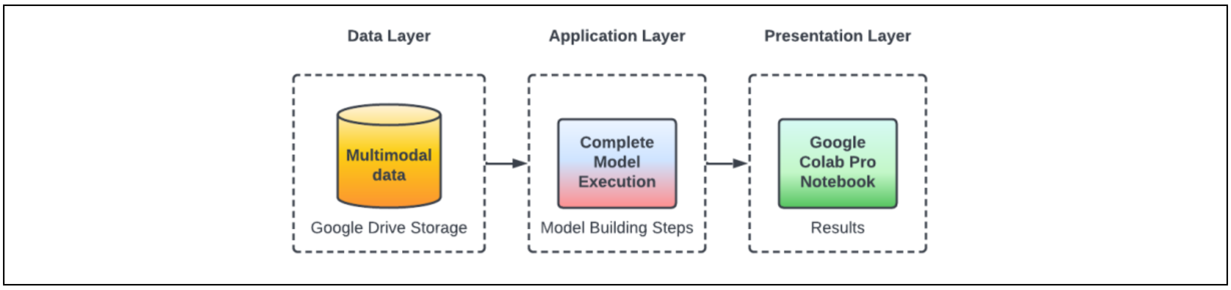

Fig. 4 Project Design Framework

The data layer in Fig. 4 includes of all the data storing steps where the multimodal data was stored in Google Drive. The application layer consists of all the data processing and model-building steps. Since no specific interface was created for this project hence, the results shown on the Google Colab Pro Notebook are included within the presentation tier.

3.3.1 Working of the Multi-Modal Architecture:

The complete multimodal architecture can be broadly divided into 3 main parts the Transformer-based text classifier, the CNN-based algorithm for image classification, and the concatenation layer using Keras Functional API for the final fake news classification. Fig. 5 depicts the custom multimodal architecture developed specifically for this research project.

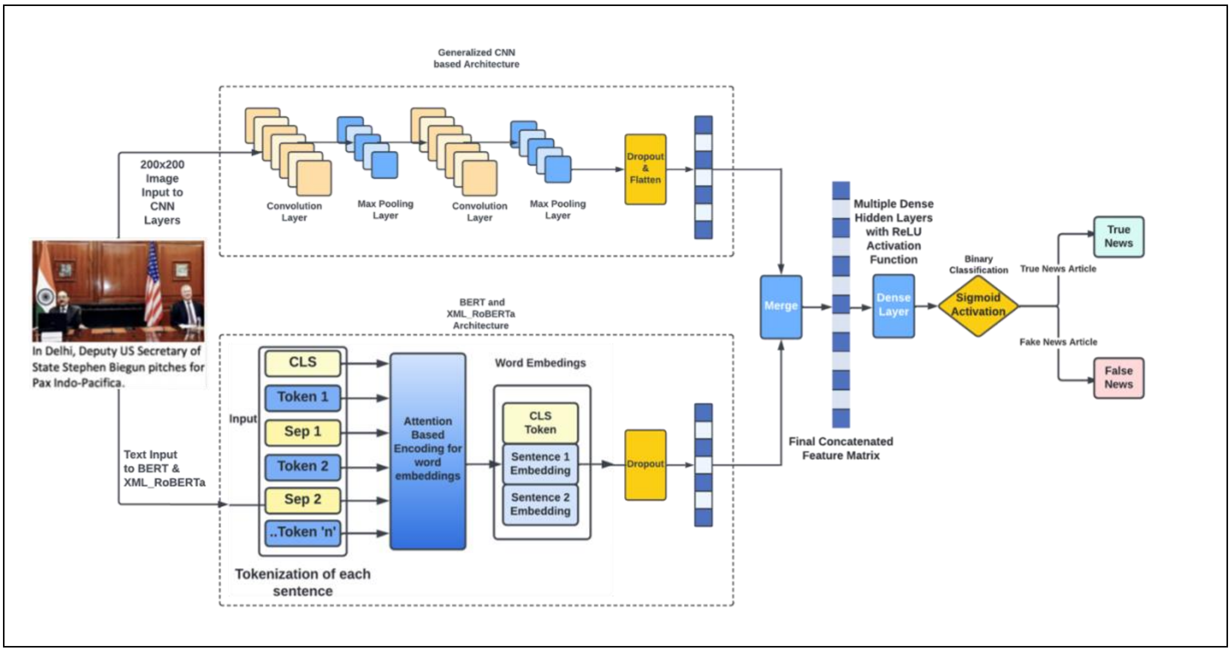

Fig. 5 Working of Multimodal Architecture

Transformer-based Algorithm Working:

The overall functioning of a Transformer can be broadly divided into two steps. Both the BERT and XML_RoBERTa models used in this project have similar working rules.

1. **Pre-processing Stage**

The BERT[4] and XML_RoBERTa[5] pre-trained models provided their custom pre-processor links which were used to generate standardized vector representations of the input news text. This process is also called the input text tokenizer which is then passed on to the attention-based encoder layer.

2. **Encoding Stage**

The output tokens are forwarded to this self-attention encoder layer wherein the pre-trained weight distributions were utilized to extract word dependencies from this new training data. This is done within the encoder layer to identify unique patterns among input text as per the labelled train data. The output of the encoder layer produced final word embeddings which were then passed through a fully connected dense layer to obtain a 1-dimensional tensor representation of the input news text.

CNN Working:

The CNN implementation had 2 major steps to obtain the final 1-dimensional matrix representation of the input image data.

1. **Convolution & Max Pooling Layers**

The corresponding input images of the news text were given as input to the convolutional layer to convolve the image using a specific filter size. A convolution output was then passed to the max pooling layer to down-sample the extracted feature matrix using a max filter. Multiple sets of these layers were added to create one custom CNN model. Additionally, in the case of the pre-trained InceptionV3 algorithm, these convolution and pooling layers were already pre-defined, and they were utilized by downloading the model using a URL and necessary libraries.

2. **Dense Layer**

The output of any CNN-based model consists of a 3-dimensional volume representation of the input image which has to be sent through a dense layer. In this project, this dense layer included a combination of dropout, flattening, and a dense layer of neurons to obtain the final 1-dimensional matrix representation of the input image.

Final News Classification using Concatenation Layer:

This was the custom layer that was developed specifically to satisfy the proposed research problem. After obtaining the 2 separate 1-D matrices of the text and image data it had to be concatenated and this was achieved using the functional API[6] of the Keras. Its working can be briefly defined in 2 important steps.

1. **Final Tensor Formation**

The CNN-based image classifier and the transformer-based text classifier were merged using the Keras ‘concatenate’ functionality. The concatenated 1-D matrix could be considered as a combined matrix representation of both the text and image algorithm outputs.

2. **Dense Layer**

The final concatenated matrix was used for the final news classification task and a unique dense layer which was the combination of dense neurons followed by a dropout layer, and the final sigmoid layer to classify the news as fake or true.

[4] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

[5] https://tfhub.dev/jeongukjae/xlm_roberta_multi_cased_L-12_H-768_A-12/1

[6] https://keras.io/guides/functional_api/

3.3.2 Model Implementation and Training Phase

The previous sub-section within the Data Mining step provided an overall understanding of the working of the implemented multimodal algorithm. In this step, the development environment and the model configuration details during the training phase will be explained in detail.

3.3.2.1 Development Environment

The complete project implementation had numerous challenges wherein the development environment had to be changed as per the requirement. Since the image data was provided in the form of URLs within the raw data frame, its downloading and pre-processing steps had to be scripted within the local Jupyter Notebook environment as explained in the above image data pre-processing section. Later, all the pre-processed images were completely moved to Google Drive so that they could be accessed during the model building stage. Apart from this, the text pre-processing, data insights, and explorations were carried out in the Google Colab Pro Platform. The primary reason to switch to this web-based IDE was the RAM and GPU requirements to train the multimodal algorithm where the 3 custom multimodal algorithms had to process approximately 37k text and image data records. Furthermore, using Google Colab Pro provided easy access to Google Drive to store all the code artifacts and the trained models in a secure location.

3.3.2.2 Model Configuration Details

In this research project, 3 multimodal algorithms and 2 unimodal text classifier models were developed, and numerous model-building steps were applied using specific parameter combinations to perform the training of these algorithms. This section provides a brief understanding of the selection of these parameters for all these techniques. Table 2 depicts the information about all the model parameters used while training the multimodal algorithms.

Table 2. Multimodal Text + Image Classifier Parameters

Model Parameters	BERT + CNN	BERT + InceptionV3	XML_RoBERTa + CNN
Trainable Parameters	74,71,745	1,73,69,601	3,54,34,945
Optimizer with Learning Rate	Adam (0.001)	Adam (0.001)	Adam (0.001)
Loss Function Used	Binary Cross Entropy	Binary Cross Entropy	Binary Cross Entropy
Early Stopping Patience	3	3	3
Batch Size	64	128	128
Epochs	10	10	10

BERT + CNN:
This technique had 3 model-building parts wherein the BERT stream was defined using the pre-processor and encoder link to obtain final pooled word embedding outputs. It was further passed to a Flattening layer to achieve a 1-dimensional feature matrix followed by the Dropout layer to avoid overfitting and lastly, a dense layer of 768 neurons using a ReLU activation function was added. The CNN stream was built using 4 sets of conv2d and max pooling layers with a kernel size of 3 x 3 and a max pooling filter size of 2 x 2. Like the BERT stream, these convolution and pooling layers were followed by Flattening and Dropout layers along with a dense layer of 512 neurons. The final part of the model building included a merging layer that concatenated the outputs of BERT and CNN streams using Keras. The

concatenated feature matrix was passed through a dense layer of 64 neurons followed by a dropout layer with a drop rate of 0.2. It was finally completed with a sigmoid layer which provided the final news classification. The created model had approximately 7.4 m trainable parameters and it was compiled using an Adam optimizer function with a standard learning rate of 0.001 and since the classification task was binary, a binary cross entropy loss function was chosen. Moreover, an early stopping with a patience value of 3 was selected during model training to avoid data overfitting. The batch size was selected as 64 with 10 epochs during the model training phase.

BERT + InceptionV3:

This multimodal algorithm had the same configuration for the BERT stream as the above-created model. The InceptionV3 image classifier stream was a pre-trained model which was directly imported using libraries and it consisted of 42 sets of convolution and pooling layers. The InceptionV3 output was sent through a Flattening and a Dropout layer followed by a dense layer of 512 neurons. The BERT and InceptionV3 output streams were passed through the exact concatenation stream as the BERT + CNN algorithm. The difference in the model features was the 17.3 m trainable parameters and a batch size of 128 which was selected to identify if it reduced the overall training time as compared to the 1st multimodal algorithm.

XML_RoBERTa + CNN:

The 3rd multimodal combination had a different selection of transformer-based text classifiers in the form of XML_RoBERTa. This pre-trained model had a higher general language understanding benchmark score than BERT and due to the elimination of next sentence prediction, it was specifically pre-trained to identify text patterns hence, it was selected to verify if it improved the overall multimodal fake news classification performance in comparison to the BERT + CNN model. The XML_RoBERTa had similar pre-processor and encoder URLs to obtain final word embedding outputs. It was then passed through the same flattening, dropout, and dense layer as BERT. The CNN-based image classifier stream and the concatenation layers had the same configuration as the previously explained BERT + CNN model. This multimodal algorithm had a higher 35.4 m trainable parameters and a batch size of 128, the early stopping patience of 3, and 10 training epochs were selected during the model training.

Table 3. Transformer-based Text Classifier Parameters

Model Parameters	BERT	XML_RoBERTa
Trainable Parameters	769	769
Optimizer with Learning Rate	Adam (0.001)	Adam (0.001)
Loss Function Used	Binary Cross Entropy	Binary Cross Entropy
Early Stopping Patience	3	3
Batch Size	32	32
Epochs	10	10

Table 3 shows the model parameter information of the two developed text classifier models BERT and XML_RoBERTa. They were selected to compare the classification performance of a unimodal algorithm with the above-explained 3 multimodal techniques. The BERT model only used the text data from the 'news_title' column wherein the pre-processor and encoder links were used to obtain final word embedding outputs. It was followed by a dropout layer with a drop rate of 0.1 along with the final sigmoid layer to classify the news as true or fake. Moreover, the trainable parameters for BERT were 769 and the model was compiled with an Adam optimizer with a learning rate of 0.001, and a binary cross entropy loss function was selected. The early stopping patience of 3 was defined during model training with 10 total epochs and a batch size of 32. Smaller batch size was chosen because of the low execution time of the text-based unimodal fake news classifier in comparison to the multimodal algorithms. The only difference in the XML_RoBERTa model was its different pre-processor and encoder links. All other model parameters were the same as the BERT algorithm.

3.4 Interpretation & Evaluation of Results (Step 5 KDD)

The first set of evaluations was to observe the behavior of the loss and accuracy values for the train and validation data sets while conducting the model training process over multiple epochs. It provided an initial understanding of whether the model was learning and extracting new patterns or was just overfitting the data.

BERT_CNN Multimodal Accuracy and Loss Plots

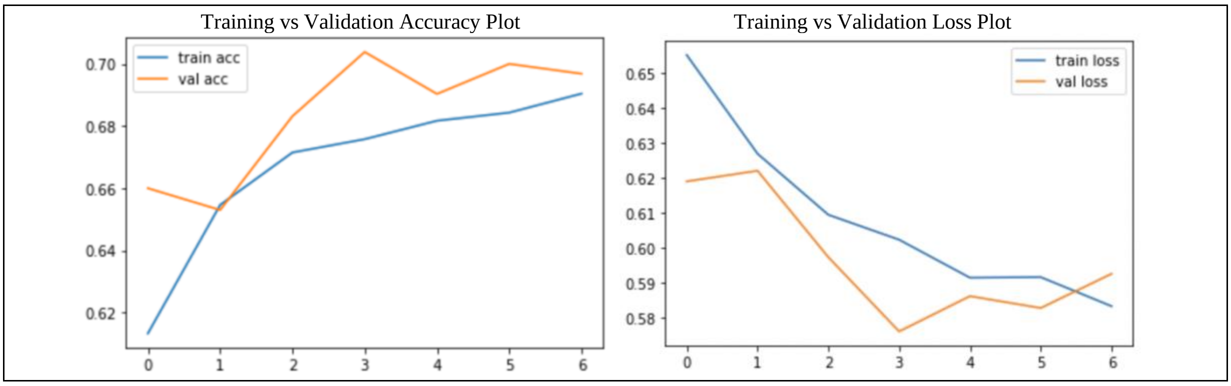

Fig. 6 BERT+CNN Accuracy and Loss Behaviour

Among the 3 multimodal algorithms, the BERT + CNN model provided the best accuracy and loss plots. The first graph in Fig. 6 showed that, as the epochs increased, so did the accuracy for both the train and validation datasets which was close to 69%. Moreover, the corresponding loss in the 2nd plot for both of these datasets decreased below 0.59, indicating that the model trained itself correctly. Because of the early stopping, the model stopped at the 6th epoch which indicated no new feature extraction from the available data.

BERT_InceptionV3 Multimodal Accuracy and Loss Plots

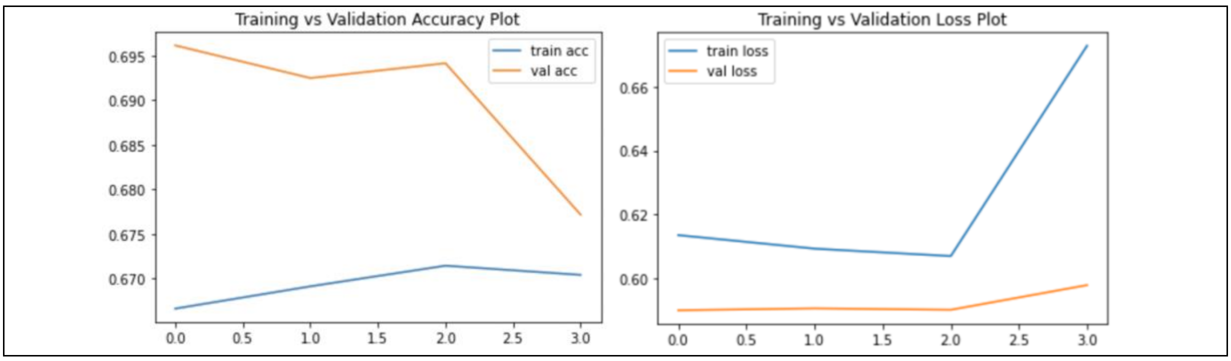

Fig. 7 BERT+InceptionV3 Accuracy and Loss Behaviour

The model training for the BERT + InceptionV3 model lasted for only 3 epochs because of the use of early stopping. The accuracy plot from Fig. 7 showed that the accuracy increased a bit for the first 2 epochs and then reduced and converged to approximately 67%

for both the train and validation sets. However, in the loss plot, it was observed that after the 2nd epoch, the model showed an unusual behavior where the loss increased for both the training and validation data.

XML_RoBERTa_CNN Multimodal Accuracy and Loss Plots

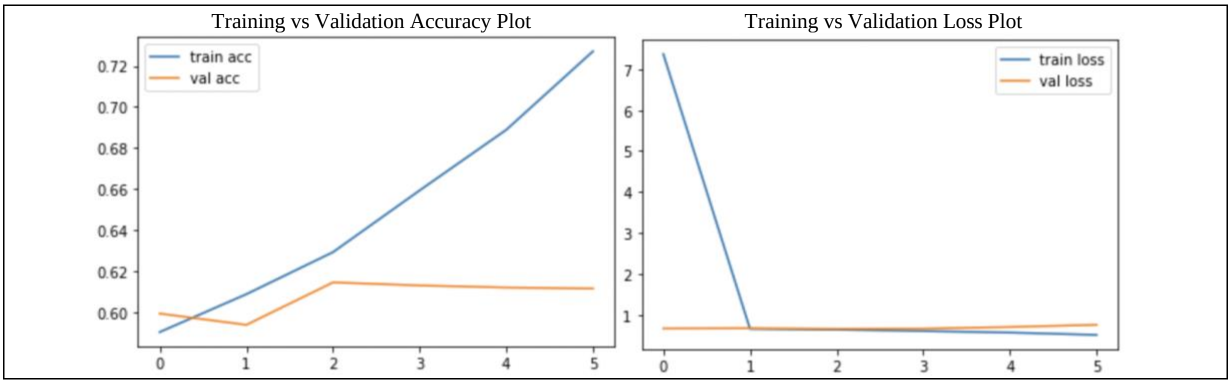

Fig. 8 XML_RoBERTa + CNN Accuracy and Loss Behaviour

Fig. 8 depicts the accuracy and loss plots for the final XML_RoBERTa + CNN multimodal algorithm. Here, the model training lasted for 5 epochs because of the early stopping patience defined in the model training. The accuracy values for the train and validation data diverged substantially after the 2nd epoch wherein the training accuracy reached close to 73% and the validation accuracy reduced below 62% which represented overfitting of training data. The loss plot depicted that it reduced throughout the model training and reached a value of 0.51 and 0.72 at the final epoch for the train and validation data respectively.

BERT Accuracy and Loss Plots

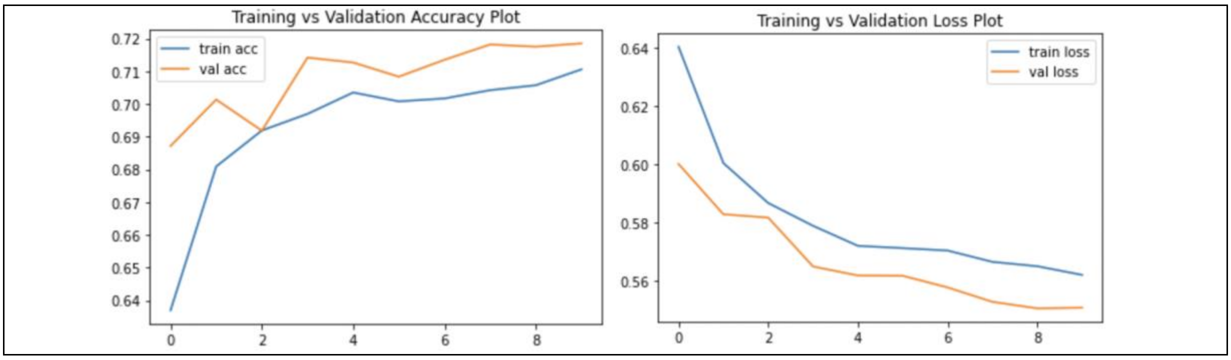

Fig. 9 BERT Accuracy and Loss Behaviour

The unimodal BERT model produced a better performance for the fake news classification in comparison to the XML_RoBERTa algorithm. Here the model training lasted for 9 epochs after which the early stopping kicked in to stop the model training process. The BERT model trained itself appropriately and this evaluation was supported by the continuous increase in accuracy of train and validation data within the accuracy plot in Fig. 9. Moreover, the loss values reduced below 0.57 for both the datasets.

XML_RoBERTa Accuracy and Loss Plots

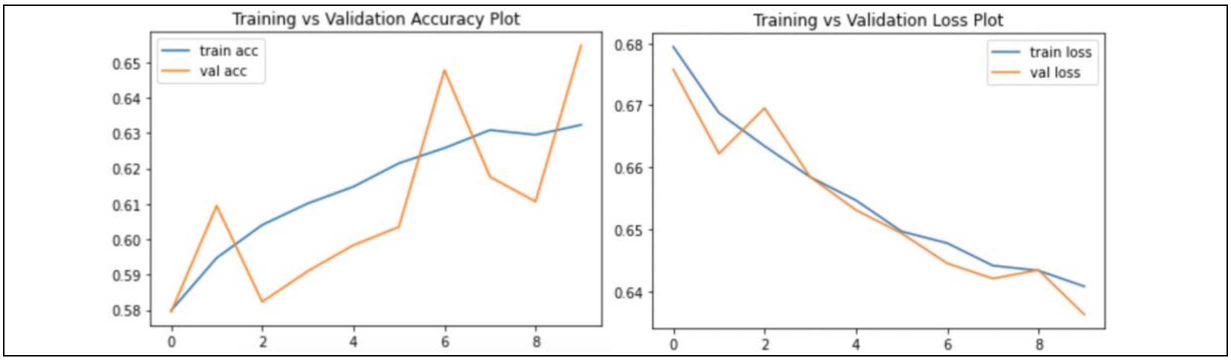

Fig. 10 XML_RoBERTa Accuracy and Loss Behaviour

The model training for the XML_RoBERTa text classifier lasted for 9 epochs, and it showed an ideal behavior for both the accuracy and loss plots which are shown in Fig. 10. The accuracy increased throughout the training phase and its values at the final epoch were close as well for both the train and validation sets. The loss plot showed ideal behavior too however, the loss values at the final epoch were close to 0.64 which was comparatively higher than the unimodal BERT model.

Performance of the Trained Algorithms on the Test Data

After evaluating the loss and accuracy plots during the model training phase, the developed techniques had to be experimented on the unseen test data to observe and assess the performance of all the algorithms for fake news detection. Fig. 11 shows two multi-bar chart representations of the Precision, Recall, and F1-score metrics for the two predicted classes - fake and true news individually.

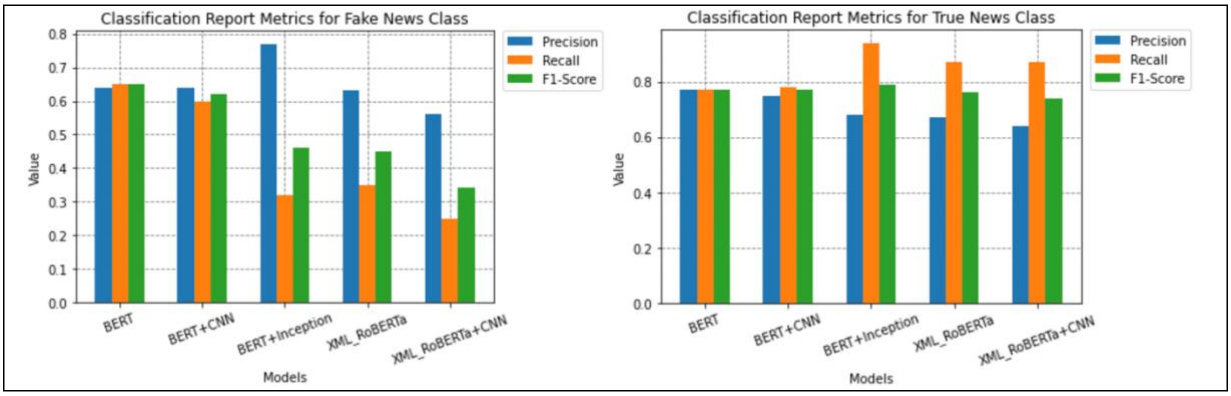

Fig. 11 Classification Metrics for the two predicted classes

From Fig. 11 it can be observed that the best overall performance in predicting both the news classes in a balanced manner was achieved by the unimodal BERT and the multimodal BERT+CNN models wherein the weighted F1-score was close to 0.71 for both these algorithms. Furthermore, the prediction of the true news category was done more correctly by all the developed models where the F1 score achieved for the predicted true news class was greater than 0.75 for all the implemented techniques. The BERT+Inception, XML_RoBERTa, and the XML_RoBERTa+CNN models performed poorly in predicting the fake news class and it was evident from their F1 score which was below 0.45 for all three models. This research experiment aimed at identifying whether the addition of image features with the text data did improve model performance in detecting fake news more accurately. However, from the test data results in Fig. 11, it could be observed that only one multimodal algorithm BERT+CNN provided similar results in comparison to the unimodal BERT model.

Confusion Matrix

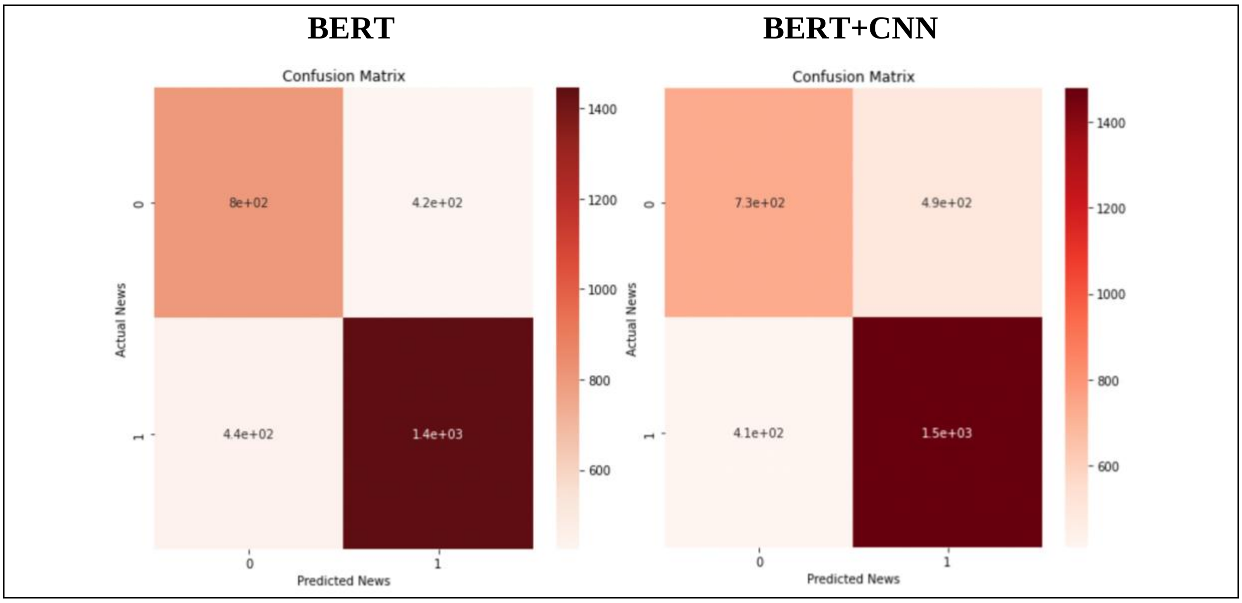

Fig. 12 Confusion Matrix of the BERT and BERT+CNN models

Fig. 12 represents the graphical representation of the confusion matrix for the best performing unimodal BERT and multimodal BERT+CNN algorithms. The BERT model predicted 802 news correctly for the fake news class which was slightly better than the BERT+CNN model's count of 735 for the same category. Moreover, the right prediction of the true news class was 1511 for the BERT+CNN model which was comparatively higher than the BERT model's tally of 1445. The performance of both these models was neck to neck in providing a balanced prediction of both the news classes accurately.

4 Discussion

The research project primarily aimed at implementing a unique multi-modal algorithm that could handle both the text and image data simultaneously to perform the fake news classification inspired by Sharma, D.K. and Garg, S., (2021). After conducting a thorough literature review it was identified that the transformer-based models provided more balanced results by utilizing the text data for fake news detection, which was evident in the research conducted by Rodríguez, Á.I. and Iglesias, L.L., (2019). Additionally, to consume the visual information various image classifiers were studied via literature to understand that the CNN-based models provided the best classification results for fake image detection (Diallo, B., et al., 2020). Furthermore, to diversify the research an additional pre-trained image classifier InceptionV3 was selected with the expectation to increase the performance of the multimodal algorithm which was influenced by the work of Nguyen, T.T. and Huynh, K.T., (2021). Hence, a research experiment to perform 3 custom multimodal combinations including a transformer-based text classifier and a CNN-based image classifier namely BERT+CNN, BERT+InceptionV3, and XML_RoBERTa+CNN were finalized to conduct this research. Moreover, the performance of these algorithms had to be compared to the two unimodal BERT and XML_RoBERTa algorithms that only used the textual features to classify the news as true or fake. This comparison was done to validate if the addition of visual data in the multimodal algorithms improved the fake news detection performance as compared to the unimodal text-based models. The final accuracy metrics achieved by each model are shown in Table 4.

Table 4. Final Accuracy Metrics

Models Implemented	Accuracy
BERT	72%
BERT + CNN	71%
BERT + Inception	70%
XML_RoBERTa	66%
XML_RoBERTa + CNN	63%

These results were crucial in analyzing the research goal of implementing a multimodal architecture to enhance the performance of fake news detection. However, it was observed that the best performing BERT+CNN multimodal architecture provided comparable results to a normal BERT model that only consumed text features. There could be multiple reasons for this occurrence but, as per the understanding, there was one factor why the multimodal accuracy did not exceed the normal BERT results. The dataset used was a subset created by Sharma, D.K. and Garg, S., (2021) which included news from a variety of domains hence, it could be difficult for the model to extract similar patterns from both the text and image data simultaneously to classify the news with high accuracy. In other words, this research work for fake news prediction could have been improved if the models were trained separately using single domain data to achieve better results. Moreover, in terms of execution time as well, the multimodal architecture approximately required 10 times the training time in comparison to the normal text-based model even though the web-based Google Colab Pro was utilized.

5 Conclusion

The modern world of the Internet and social media has made it easier to access any piece of information but, most of the time these websites and applications are used to mislead large masses by spreading fake information. It becomes a challenging task to factually verify each and every piece of news available on the internet. Hence, the project's objective was to determine whether adding visual data along with textual elements improved the accuracy of fake news detection using a multimodal algorithm. Hence, a total of three multimodal techniques BERT+CNN, BERT+InceptionV3, and XML_RoBERTa+CNN were designed and implemented in this study and the best result of 71% accuracy was achieved by the BERT+CNN model. This multimodal result was compared with the unimodal BERT model that obtained an accuracy of 72% on the same dataset by only utilizing text features. This research experiment signified that the addition of visual features did not elevate the performance of the multimodal algorithm to detect fake news instead, it provided similar results as the text-based model. This comparison also indicated that the majority of the critical information in the fake news classification task is present within the text data.

In conclusion, the use of generalized data which included news from multiple domains posed a difficulty in similar feature extraction from the concurrent supply of varied text and image data during the model training phase. A potential future solution to this limitation would be training models on domain-specific data to identify if it achieved higher detection accuracy than the generalized multi-domain data. Moreover, an introduction of a pre-trained Vision Transformer for image classification may yield interesting results for this multimodal fake news detection task. Another possible research area for this project would be an addition of text-related metadata, which could be useful in improving the classification performance of the multimodal algorithms.

6 Acknowledgement

I feel appreciative and would want to express my gratitude to Prof. Vladimir Milosavljevic, my research project supervisor, for his encouraging assistance and direction in helping me complete my research thesis successfully. I would also like to thank the NCI Library facility for its easy access to various resources required for the thesis project. Lastly, I am also grateful for the support my family and friends gave me to accomplish my goal.

References

Ahmad, I., Yousaf, M., Yousaf, S. and Ahmad, M.O., 2020. Fake news detection using machine learning ensemble methods. *Complexity*, *2020*.

AlShariah, N.M., Khader, A. and Saudagar, J., 2019. Detecting fake images on social media using machine learning. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, *10*(12), pp.170-176.

Diallo, B., Urruty, T., Bourdon, P. and Fernandez-Maloigne, C., 2020. Robust forgery detection for compressed images using CNN supervision. *Forensic Science International: Reports*, *2*, p.100112.

Han, Y., Karunasekera, S. and Leckie, C., 2020. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*.

Hiramath, C.K. and Deshpande, G.C., 2019, July. Fake news detection using deep learning techniques. In *2019 1st International Conference on Advances in Information Technology (ICAIT)* (pp. 411-415). IEEE.

Jindal, S., Sood, R., Singh, R., Vatsa, M. and Chakraborty, T., 2020. Newsbag: A multimodal benchmark dataset for fake news detection. In *CEUR Workshop Proc.* (Vol. 2560, pp. 138-145).

Kaliyar, R.K., Goswami, A. and Narang, P., 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia tools and applications*, *80*(8), pp.11765-11788.

Khanam, Z., Alwasel, B.N., Sirafi, H. and Rashid, M., 2021, March. Fake news detection using machine learning approaches. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1099, No. 1, p. 012040). IOP Publishing.

Kuznetsov, A., 2019, November. Digital image forgery detection using deep learning approach. In *Journal of Physics: Conference Series* (Vol. 1368, No. 3, p. 032028). IOP Publishing.

Monti, F., Frasca, F., Eynard, D., Mannion, D. and Bronstein, M.M., 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Nguyen, T.T. and Huynh, K.T., 2021, November. Spliced Image Forgery Detection Based on the Combination of Image Pre-processing and Inception V3. In *International Conference on Future Data and Security Engineering* (pp. 308-322). Springer, Cham.

Qi, P., Cao, J., Yang, T., Guo, J. and Li, J., 2019, November. Exploiting multi-domain visual information for fake news detection. In *2019 IEEE international conference on data mining (ICDM)* (pp. 518-527). IEEE.

Reis, J.C., Correia, A., Murai, F., Veloso, A. and Benevenuto, F., 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, *34*(2), pp.76-81.

Ren, Y., Wang, B., Zhang, J. and Chang, Y., 2020, November. Adversarial active learning based heterogeneous graph neural network for fake news detection. In *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 452-461). IEEE.

Rodríguez, Á.I. and Iglesias, L.L., 2019. Fake news detection using deep learning. *arXiv preprint arXiv:1910.03496*.

Saleh, H., Alharbi, A. and Alsamhi, S.H., 2021. OPCNN-FAKE: optimized convolutional neural network for fake news detection. *IEEE Access*, *9*, pp.129471-129489.

Sharma, D.K. and Garg, S., 2021. IFND: a benchmark dataset for fake news detection. *Complex & Intelligent Systems*, pp.1-21.

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

Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z. and Yu, P.S., 2018. TI-CNN: Convolutional neural networks for fake news detection. *arXiv preprint arXiv:1806.00749*.

Zhong, J.L. and Pun, C.M., 2019. An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, *15*, pp.2134-2146.