Deep Learning Approaches for Fake News Detection: A Comprehensive Review

Abstract

In recent years, the proliferation of fake news or fabricated information with the intention of deceiving the public has become increasingly prevalent. The dissemination of such false information undermines social cohesion and well-being by promoting political division and eroding trust in government. With the sheer volume of news being shared through social media, it has become difficult for humans to verify the authenticity of each piece of news, leading to the development and deployment of automated methods for identifying fake news. Fake news publishers employ a variety of stylistic techniques to increase the popularity of their content, such as evoking strong emotions in readers. As a result, sentiment analysis, a technique that identifies the polarity and intensity of emotions conveyed in a text, is now being used in false news detection methods, either as the primary approach or as a supplementary component. This research aims to provide a comprehensive overview of fake news detection, including its characteristics, features, taxonomy, types of data used, categories of fake news, and various detection techniques. The study also examines the foundational theory of related research to provide a thorough comparative analysis of the literature that has contributed to this field. Additionally, a comparison of machine learning and deep learning techniques is conducted to evaluate their performance for detecting fake news. To accomplish this, three datasets are utilized in the research.

Keywords: Fake News, Deep Learning, Machine Learning, BERT, ROBERTA

1 Introduction

Fake news is the purposeful presentation of false or misleading statements as news when the assertions are purposely misleading. The phrase "by design" is then explained in terms of systemic elements of the news generation and distribution process [1]. Communication tools have developed over time, and so has the concept of news consumption. In this digital age, news readers now have many options at their fingertips.The dissemination of propaganda and disinformation is increasingly pervasive in cyberspace, calling into doubt the trustworthiness of social media platforms, news media platforms, and news publishers. When important events in future news are affected by false information, consumers are more likely to identify false information as important information and are less likely to notice true information. [2]. Much debate

and research has focused on the political impact of fake news after the 2016 US presidential election. However, fake news is already spreading across the spectrum and has begun influencing other aspects of life. In today's world, where the dynamics of the globe are continuously changing, false news has contributed to the wider economy by producing conflict among cultures and groups. It would be hazardous to imagine what potential fake news has to produce mayhem in unforeseen occurrences. The identification of fake news is becoming increasingly difficult as a large volume of incorrect material spreads rapidly through social media. To address this issue and manage massive amounts of data simultaneously, deep learning models have already established their identity and demonstrated significant results in identifying false news [3].
"Fake news" or misrepresented material has become an important phenomenon in the Internet-based news environment. Scientists are investigating the causes, properties and consequences of its production and transmission, and it has attracted great interest in many fields. Types of news:

Real news [4]: Real news is reliable, unbiased coverage of current affairs that is often created by qualified journalists working for renowned news agencies. Real news is based on confirmed facts and is meant to inform the public about significant local, national, and international events and issues. It tries to offer a fair and truthful portrayal of events and is not influenced by the journalists' personal prejudices or ideas. Fake news, on the other hand, is defined as news that has been purposefully produced or deceptive in order to mislead readers and distribute false information. Before assuming something as news, it's critical to verify the facts with reliable sources.

Misinformation and fake news [5]: Fake news, by contrast, refers to the deliberate spread of misinformation, often misleading information that spreads conspiracy theories. While the nature of these materials may seem polarizing and sensational content (discussed later in this article), the message can be seen as a whole and emotion. So now if we have false new/hoaxes and real news then what is fake news.

1.1 What is fake News?

Although the word "fake news" now has a clear definition in the dictionary as "a false story that appears to be news, is broadcast on the Internet or is used in other media, often designed to distort emotions or jokes" (Fake News, 2018), it can distinguish what is not from what is not. it is difficult to do. There is much debate over deciding which information should be included and which should be excluded. This is especially true because the term "fake news" has important political connotations and is often used as a buzz word to attack the truth of news organizations or an incredible word that disagrees with us. Also, recording something wrong requires establishing universal truth, a difficult task that requires the approval of all participants.

1.2 Types of Fake news

Fake news refers to intentionally fabricated or misleading information presented as if it were real news. Here are some different types of fake news [6]:

- **Clickbait** [7]: Clickbait is a type of fake news that uses sensational or exaggerated headlines to attract readers and generate clicks on a website. The content of the

article is often not accurate or even related to the headline. Clickbait headlines are designed to grab people's attention and get them to click on a link, often leading them to an advertisement or a low-quality article that provides little or no valuable information. Clickbait articles are often shared on social media, and they can spread quickly, leading to misinformation and confusion.

- **Satire or Parody**[8]: Satire and parody are forms of fake news that use humor or irony to comment on current events. Satirical news websites, such as The Onion or The Daily Show, use exaggerated or absurd stories to make a point or poke fun at a particular issue or person. While these stories are not intended to deceive, some people may mistake them for real news. Parody news stories, such as those created by the website Clickhole, imitate the style and format of real news stories but with a humorous twist.
- **Propaganda**[7]: Propaganda is the use of misleading or biased information to promote a particular political agenda or viewpoint. Propaganda can take many forms, such as biased reporting, false or misleading statistics, or selective use of facts. It can be spread through traditional media, such as television or newspapers, or through social media and other online platforms. Propaganda is often used to influence public opinion and can be difficult to detect because it is presented as factual information.
- **Hoaxes**[5]: Hoaxes are fake news stories or videos that are intentionally created to deceive people. They often contain false information or fake quotes from real people, and they may be spread through social media or email. Hoaxes can be used for a variety of purposes, such as to make money through advertising revenue, to discredit a political opponent, or to stir up controversy and gain attention.
- **Conspiracy theories** [7]: Conspiracy theories are explanations of events or situations that are based on unproven or unlikely claims of secret plots or hidden agendas. Conspiracy theories often involve the government, powerful corporations, or other groups that are believed to be involved in a cover-up or other nefarious activity.Solidarity ideas are spread online on social media, websites, etc. and can have real-world consequences, such as discouraging people from getting vaccinated or engaging in violence.

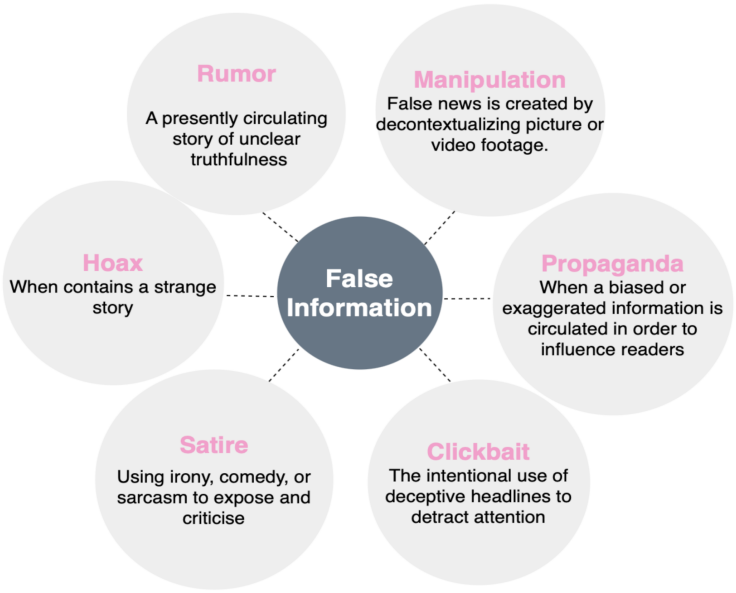


Fig. 1 Types of false/fake information [7]

1.3 Consequences of fake news

Fake news is, and still is, one of the biggest challenges facing the global economy, journalism and democracy.[9].

The following are some possible effects of fake news:

- **False information:** By disseminating false information, fake news can cause people to believe things that are false. Confusion, misconceptions, and a lack of faith in reliable information sources might result from this.
- **Political polarisation:** False news can exacerbate political polarisation and widen social gaps. It may foster an atmosphere of mistrust and hostility between various social groupings, resulting in instability and war.
- **Economic effects:** False news may also have financial effects. It may have an impact on consumer behaviour, stock prices, and even entire sectors. For instance, a false news report regarding a company's financial performance can drive down the price of its stock, costing investors money.
- **Manipulation:** The use of fake news to sway public perception and influence political outcomes. This can be especially risky when it comes to elections since inaccurate information can skew voter behaviour and threaten the democratic process.

In general, fake news is a significant issue with potentially serious ramifications. In order to prevent being mislead by erroneous information, it's critical to exercise caution and look for trustworthy sources of information.

1.4 Prevention of Fake News

In order to reduce the negative impact of fake news, the spread of false information through various channels, especially online channels, has not been completely stopped or reduced. This is because there is currently no mechanism to check for fake news that doesn't affect people. Given the first examples to train, experiments show that machines and learning algorithms can identify fake news. Using Machine Learning and Deep Learning Models to Detect Fake News. Deep learning techniques show great promise in fake news detection tasks. Many studies have shown the importance of neural networks in this field. In this project, we used some deep learning models and some machine learning models to detect fake news from different data and compare it to real news, and we also talked about the problems faced by data and models.

2 Fake News Detection

Fake news detection has been an major important issue in last few years, as the spread of false or misleading information has been a major problem in the age of interenet. Detecting fake news is critical for ensuring that people receive accurate and reliable information and can make informed decisions.

Source-based detection, content-based detection, social media-based detection, and network-based detection are some of the methods used to identify fake news. In source-based detection, the reliability of a news article's source, such as the publisher or author, is assessed. Content-based detection involves analyzing the language, style, and content of a news article to identify patterns or anomalies that may suggest it is fake. Social media-based detection involves monitoring social media platforms to identify and track the spread of fake news. Network-based detection involves analyzing the connections and relationships between different news sources and social media users to identify patterns of misinformation.

Effective fake news detection requires a combination of these approaches, as well as the use of advanced technology such as natural language processing, machine learning, and data analytics. However, it is important to remember that fake news detection is not a foolproof process, and there are always limitations and challenges to detecting and combating fake news.In addition to technological solutions, addressing the problem of fake news also requires a commitment to media literacy and critical thinking skills. Educating people about how to identify and evaluate reliable sources of information, as well as teaching them how to spot and analyze patterns of misinformation, can help to promote a more informed and media-savvy public. In this section we solely focus on fake news and different aspects of fake news detection.

2.1 Catagories of Fake News Detection

By categories of fake news detection, we mean the different approaches or methods used to detect and identify fake news. These categories represent different ways of analyzing and evaluating news articles to determine whether they are accurate and reliable or false and misleading.

There are several categories of fake news detection, including source-based detection, content-based detection, social media-based detection, and network-based detection. Each of these categories involves a different approach to analyzing news articles and identifying patterns or anomalies that may suggest that the news is fake.

By using a combination of these categories, researchers and analysts can develop more effective methods of fake news detection that can help to promote accurate and reliable information and protect against the spread of false or misleading news.

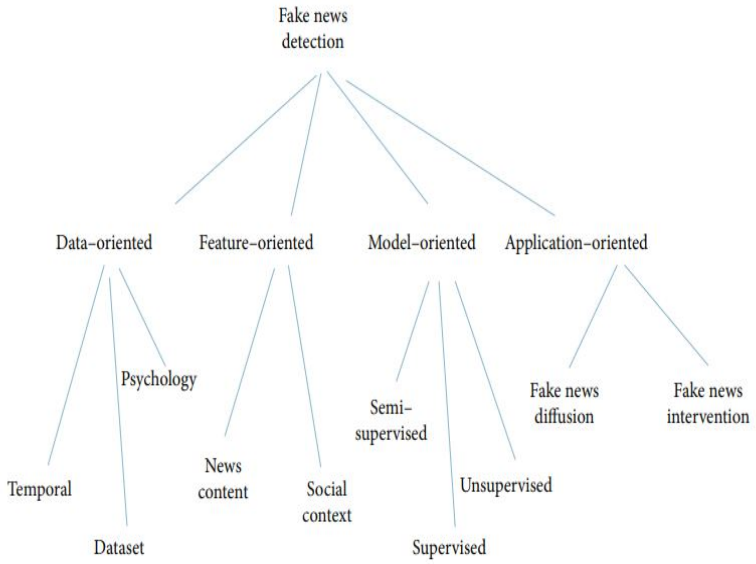


Fig. 2 Fake news detection methods approaches

2.2 Features of Fake News Detection

Fake news detection [10] involves the use of various techniques and technologies to identify and distinguish fake news from real news. The following are typical characteristics used in identifying fake news:

- **Source evaluation:** One of the most important features of fake news detection is evaluating the credibility of the source. This involves assessing the reputation of the website or media outlet that published the news, as well as investigating the author and their credentials. Researchers and analysts can use a variety of tools to evaluate the credibility of a source, such as checking the website's domain name and examining the website's history and content.
- **Fact-checking:** Fact-checking involves verifying the accuracy of the information presented in the news article. This includes checking the sources used in the article and comparing the information with other reliable sources. Fact-checking can be done manually, by researchers and analysts, or through automated tools that use

natural language processing and machine learning algorithms to identify false or misleading information.

- **Content analysis:** Content analysis involves examining the language and structure of the news article to identify any patterns or anomalies that may suggest that the news is fake. For example, fake news articles often use emotional language, exaggeration, and sensational headlines to attract attention. Content analysis can be done manually, by researchers and analysts, or through automated tools that use natural language processing and machine learning algorithms to identify patterns in respective articles langauge.
- **Social media analysis:** Social media analysis involves monitoring the spread of news articles on social media platforms and identifying any patterns or trends that may suggest that the news is fake. This can include analyzing the number of shares, likes, and comments on a post, as well as examining the profiles of the people who are sharing the news. Social media analysis can be done manually, by researchers and analysts, or through automated tools that use machine learning algorithms to identify patterns in social media activity.
- **Machine learning:** Machine learning involves using algorithms and statistical models to identify patterns and anomalies in large datasets. This can be applied to fake news detection by analyzing large collections of news articles and identifying common characteristics of fake news, such as language use and headline structure. Machine learning can be used to develop automated tools for fake news detection, such as algorithms that scan news articles and social media posts for false or misleading information.

Overall, fake news detection requires a combination of technical and analytical skills, as well as a deep understanding of the political and social context in which the news is being produced and disseminated. By using these features, researchers and analysts can help to combat the spread of fake news and promote accurate and reliable information.

2.3 Different Types of Data in News

Here type of data in news, we mean the different categories or formats of information that are used in news reporting to provide context, support arguments, and illustrate patterns and trends.

The use of data in news reporting can help to provide readers with a more complete understanding of the issues being discussed, and can help to support arguments and claims with empirical evidence[11]. However, it is important for journalists and news organizations to use data ethically and accurately, and to be transparent about the sources and methods used to gather and analyze the data.

y analyzing and presenting different types of data in news reporting, journalists and news organizations can help to promote more informed and nuanced discussions of the issues that shape our world. Various forms of data are commonly employed in news reporting, with the following being some of the most prevalent types.

- **Statistical data:** Statistical data includes numerical knowledge and is often used to support a news article or provide context for an issue. Examples of statistical

data include population figures, crime rates, economic indicators, and poll results. Statistical data can be presented in a variety of formats, such as charts, graphs, and tables.

- **Geographic data:** Geographic data includes information about the location and distribution of people, places, and things. This type of data can be used to provide context for a news story or illustrate patterns or trends. Examples of geographic data include maps, population density figures, and climate data.
- **Time-based data:** Time-based data includes information that is organized by time, such as historical events, timelines, and schedules. This type of data can be used to provide context for a news story or help readers understand the sequence of events that led to a particular situation.
- **Social media data:** Social media data includes information that is generated on social media platforms, such as Twitter, Facebook, and Instagram. This type of data can be used to track trends, analyze public opinion, and monitor the spread of news stories.
- **Financial data:** Financial data includes information about the financial performance of companies and economies. This type of data can be used to provide context for news stories about business, finance, and the economy.

Overall, the use of data in news can help to provide context, support arguments, and illustrate patterns and trends. However, it is important to use data ethically and accurately and to be transparent about the sources and methods used to gather and analyze the data.

3 Model-oriented detection methods

3.1 Machine Learning

3.1.1 SVM

The powerful yet flexible supervised machine learning approach known as a support vector machine (SVM) is utilised for both classification and regression. However, they are frequently used in classification-related problems.

Operation of SVM: -A multidimensional spatial hyperplane that represents many classes is all that an SVM model is. To cut down on error, SVM will create the hyperplane iteratively. Finding a maximum marginal hyperplane (MMH) is the goal of SVM, which classifies datasets.

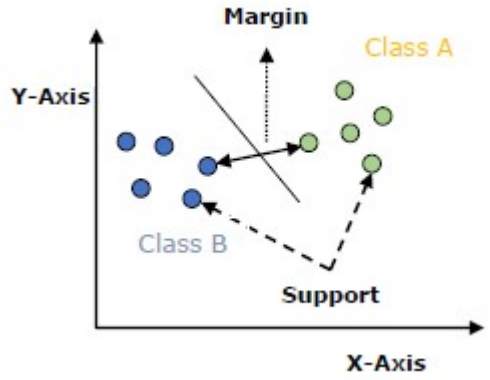


Fig. 3 Support Vector Machine [12]

Important concepts in SVM are given below:

- **Support Vectors:** Support vectors are the data points that are closest to the hyperplane. These data points will be used to define the separating line.
- **Hyperplane:** As seen in the diagram above, there are a number of objects with various classes divided up into this decision plane or space.
- **Margin:** Margin can be calculated as the perpendicular distance from the line to the support vectors, and it can be described as the space between two lines on the closet data points of different classes. A large margin is regarded as a good margin, and a little margin is seen as a bad margin.

SVM kernels: The SVM algorithm is really implemented with a kernel that modifies the input data space to take the desired form. The kernel trick is a method used by SVM, in which the kernel converts a low-dimensional input space into a higher-dimensional space.

1. **Linear kernel:** Between any two observations, it may be utilised as a dot product.
2. **Polynomial Kernel:** It distinguishes between input spaces that are curved or nonlinear and is a more generalised variant of the linear kernel.
3. **Radial Basis Function (RBF) Kernel:** The input space is mapped in an infinitely dimensional space by the RBF kernel, which is mostly utilised in SVM classification.

Table 1 : Kernel Function.

Kernel function	Mathematical formula
Linear kernel	$k(X_i, X) = X_i \cdot X$
Polynomial kernel	$k(X_i, X) = (X_i \cdot X)^e$
Normalized Polynomial kernel	$k(X_i, X) = [(X_i \cdot X)^e] / \sqrt{(X_i \cdot X_i)^e (X \cdot X)^e}$
Radial basis kernel (RBF)	$k(X_i, X) = e^{-(\lvert X_i - X \rvert^2 / 2\sigma^2)}$

3.1.2 Decision Tree

A decision tree[13] is a tool for supporting decisions that employs a graph or model that resembles a tree to represent options and potential outcomes, such as utility, resource costs, and outcomes of random events. There are nodes and branches in a decision tree. The decisions or events are represented by the nodes, while the potential outcomes or outcomes are represented by the branches.

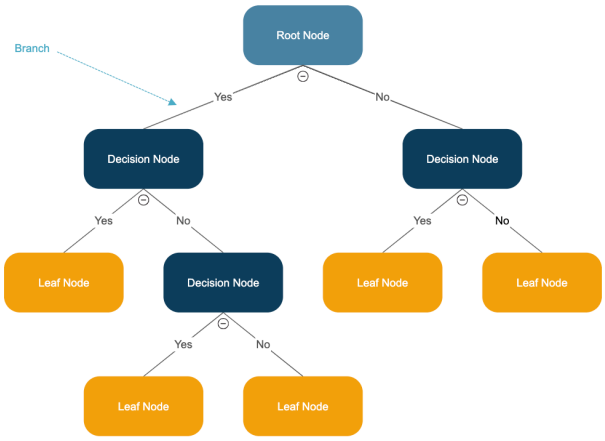


Fig. 4 Decision tree [14]

Each leaf node refers to a class or a prediction, whereas each interior node reflects a judgment made in light of the value of a certain characteristic or attribute.

The function can be represented as follows:

$$Y = f(X) \tag{1}$$

According to the values of the features, the data are iteratively divided into subgroups by the decision tree algorithm, which then produces the tree. The feature that maximises the information gain or the Gini index is the one the algorithm selects at each internal node as the one that delivers the best split, or the feature that offers the best split.

The Gini index calculates the likelihood that a random sample would be misclassified, while the information gain calculates the reduction in entropy or impurity of

the data after the split. The information gain is defined as follows:

$$IG(S, F) = H(S) - H(S \mid F) \tag{2}$$

$$IG(S, F) = H(S) - H(S|F) \tag{3}$$

where S is the set of samples, F is the feature, H(S) is the entropy of S, and H(S—F) is the conditional entropy of S given F. The entropy is defined as:

$$H(S) = -sum(pi * log2(pi)) \tag{4}$$

where pi is the proportion of samples in S that belong to class I. The conditional entropy is defined as:

$$H(S|F) = sum(pi * H(Si)) \tag{5}$$

where Si is the subset of S where feature F has value I. The Gini index is defined as follows:

$$Gini(S) = 1 - sum(pi^2) \tag{6}$$

where pi is the proportion of samples in S that belong to class i.

The decision tree approach keeps splitting data until all samples fall into the same class or until a stopping requirement, such as the maximum depth of the tree or the minimal number of samples needed to split a node, is satisfied.

3.1.3 Multinomial Naive Bayes

Assuming that the characteristics are conditionally not dependent given the class label, Bayes is a classification algorithm. It is frequently utilised in tasks involving text classification and natural language processing (NLP). It is based on the following formula:

$$P(A \mid B) = P(B \mid A) * P(A)/P(B) \tag{7}$$

Multinomial naive Bayes model[15] is mathematically represented as follows:

The fundamental premise of the model is that, given the class label, the features are conditionally independent, which means that the existence or absence of one feature is independent of the existence or absence of any other feature.

The multinomial naive Bayes algorithm is a well-known and popular classification algorithm for NLP and text classification jobs because it is straightforward yet highly successful.

3.1.4 Passive aggressive classifier

In text classification and data stream mining, the passive-aggressive (PA) method is a common online linear classifier. It is a straightforward but powerful algorithm that works well with big, noisy data and can swiftly adapt to shifting data distributions[15].

The PA algorithm's core principle is to continuously update the linear classifier's weight vector by adding new weights after viewing each instance. With a margin

limitation in place, the updates are intended to minimize the loss function. Although the margin restriction makes sure that the classifier maintains a minimum distance between the decision border and the training instances, the loss function calculates the difference between the predicted label and the actual label.

3.1.5 KNN

A non-parametric machine learning approach called K-Nearest Neighbors (KNN)[16] can be utilised for both classification and regression applications. In order to estimate the target value of the test point, it locates the K-nearest data points to the supplied test point. The KNN algorithm can be explained mathematically as follows:

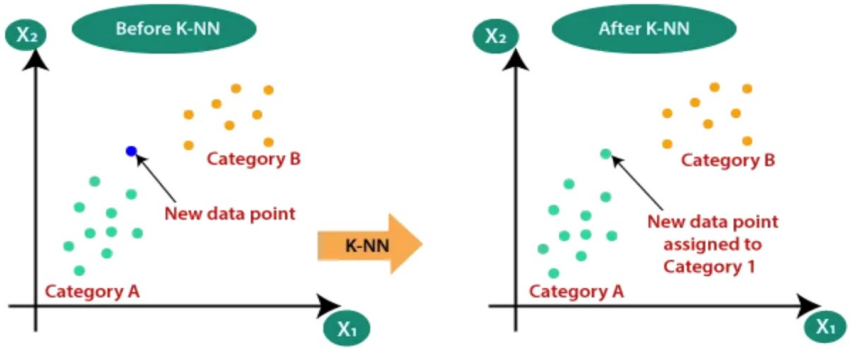


Fig. 5 KNN example[17]

1. Let Y be the equivalent set of target values and X be the set of input data points. Let yi be the appropriate goal value and x_i be a single input data point.
2. The KNN method operates in the following manner to forecast the target value of a fresh input data point, $x*$.
3. Measure the separation between each data point xi in X and x*. Several metrics, including the Manhattan distance, the cosine similarity, and the Euclidean distance, can be used to determine the distance.
4. Using the estimated distances as a guide, choose the K-nearest data points to x*. These informational points are x"neighbors." *'s.
5. The mean of xK-nearest *'s neighbours' goal values serves as the projected target value for regression tasks:
$$y* = (1/K) * \sum yi \tag{8}$$
6. The predicted target value of x* for classification tasks is the target class that received the most votes from its K-nearest neighbours:
$$y* = argmax \sum (yi = j) \tag{9}$$
where j encompasses all potential classes.

7. The KNN algorithm should output the expected target value y*. It should be noted that selecting the hyperparameter K's value can be adjusted to get the KNN algorithm to operate at its best on a particular dataset.

Formulas used for calculating distances:

1. **Euclidean Distance:**The square root of the sum of the squared differences between a new point (x) and an existing point is used to determine the Euclidean distance (y).

$$D = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{10}$$

2. **Manhattan Distance:**By adding up their absolute differences, two real vectors are separated by this amount.

$$D = \sum_{i=1}^{k}|x_i - y_i| \tag{11}$$

3. **Hamming Distance:**For categorical variables, it is employed. The distance D will be equal to zero if the values (x) and (y) are the same. If not, D=1.

$$D = \sum_{i=1}^{k}|x_i - y_i| \tag{12}$$

3.1.6 Random forest[13]

It is a well-liked machine-learning technique that is utilised for both classification and regression applications. It is technique that builds a large number of decision trees during training and produces a class that represents the mean of the predictions (regression) or classifications made by the individual trees. Here's how it works mathematically:

1. **Dataset:** Let's say we have a training dataset with N observations and p features (X1, X2, X3, ..., Xp) and a response variable Y.
2. **Bootstrap Sampling:** Random forest algorithm creates multiple subsets of the dataset by sampling with replacement. For each subset, it selects a random subset of the features.
3. **Decision Trees[13]:** A decision tree is built for each subset of the data. This algorithm selects the best feature to split the data at each node. The splitting criterion is typically based on the reduction in the impurity of the node.
4. **Ensemble of Trees:** The output of the random forest is the average prediction (regression) or the mode of the predictions (classification) of all the decision trees.
5. **Math:** For regression, the prediction for a new observation is the mean of the predictions of all the decision trees:

$$y_{pred} = (1/m) \sum_{1}^{m} (y_{Pred}, i) \tag{13}$$

For classification, the prediction for a new observation is the mode of the predictions of all the decision trees:

$$Y_{pred} = mode(Y_{pred}, 1, Y_{pred}, 2, ..., Y_{pred}, m) \tag{14}$$

where m is the count of decision trees in the forest.

The diagrammatical working representation of a random forest is shown below:

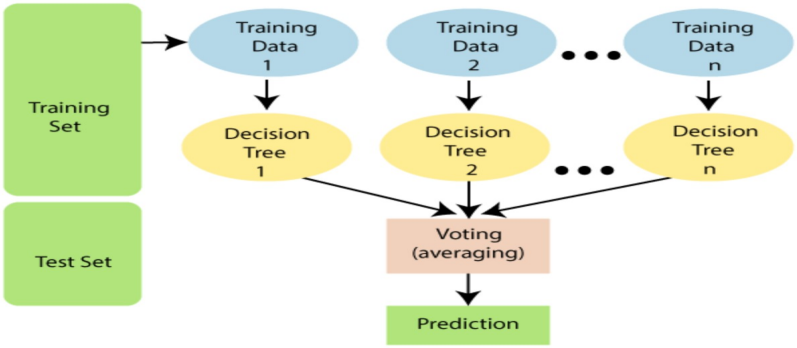


Fig. 6 Working of Random Forest[18]

3.1.7 Logistic Regression

A set of independent variables (predictors) and a binary outcome variable (response) with just two possible values, such as 0 or 1, are analysed using the statistical technique of logistic regression[19]. Searching for the optimal-fitting model that estimates the probability of the response variable given the values of the predictors is the aim of logistic regression.

The logistic regression model converts a linear combination of variables into a probability value that spans from 0 to 1 using the logistic function, commonly referred to as the sigmoid function. The logistic function is defined as follows:

$$p(y = 1|x) = 1/(1 + exp(-z)) \tag{15}$$

where p(y=1—x) is the probability of the response variable y=1 given the predictor variables x, exp() is the exponential function, and z is the linear combination of the predictors and their coefficients:

$$z = B0 + B1x1 + B2x2 + ... + Bpxp \tag{16}$$

where B0 is the intercept, B1, B2, ..., Bp are the coefficients for the predictors x1, x2, ..., xp, respectively.

To get the ideal coefficient values that maximise the likelihood of the observed data, the logistic regression model is trained using the maximum likelihood estimation approach. By estimating the likelihood that the response variable would equal 1 given the values of the predictors, the model can be used to make predictions.

For determining the likelihood that an event will occur based on a collection of predictors, logistic regression is frequently used in a variety of industries, including healthcare, finance, and marketing.

Below is the graphical representation of logistic function :

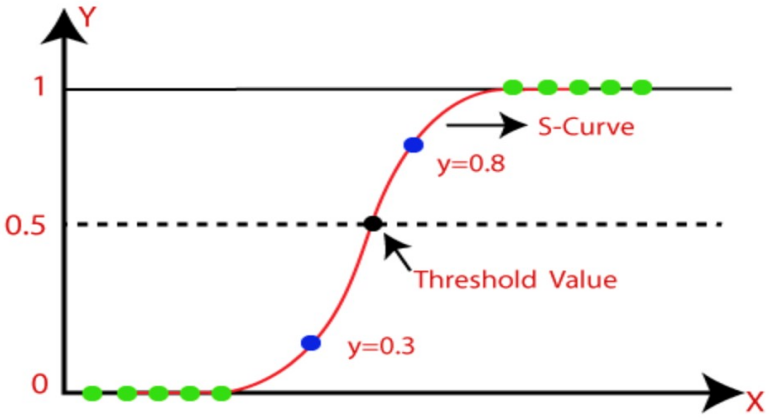


Fig. 7 Logistic Function [20]

3.2 NLP Methods for Feature Extraction

NLP stands for Natural Language Processing, which is a branch of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language[21]. NLP techniques involve the application of algorithms and statistical models to text and speech data, with the aim of extracting meaning and insights from them. NLP techniques can be implimented in a vast range of domains, including customer service, healthcare, finance, and social media analysis, among others. They are used to enable machines to communicate with humans in a more natural way, automate repetitive tasks, and gain insights from higher amounts of unstructured data. In this section we have discussed some NLP techniques which we have used and which are generally used for processing the data in ml and dl and deep learning models.

3.2.1 Tokenization

Tokenization[22] is the process of breaking down a piece of text into smaller units, called tokens. These tokens could be words, phrases, or even individual characters,

depending on the specific requirements of the task at hand. Tokenization is an essential step in many natural language processing (NLP) tasks, such as text classification, named entity recognition, and sentiment analysis. Mathematically, we can represent tokenization as a function that takes a string of text as input and produces a sequence of tokens as output. Let's denote the input string as S, and the output sequence of tokens as T. Then we can represent tokenization as follows:

$$T = tokenize(S) \tag{17}$$

The exact method of tokenization depends on the specific requirements of the task at hand. For example, if we're performing sentiment analysis, we may want to tokenize the text into individual words. On the other hand, if we're performing named entity recognition, we may want to tokenize the text into phrases or named entities.

Let's take an example to illustrate the process of tokenization. Suppose we have the following input string:

"Tokenization is the process of breaking down a piece of text into smaller units, called tokens."

To tokenize this string, we could use a simple rule-based approach that splits the string into individual words based on whitespace and punctuation:

$T = [$"$Tokenization$","is","the","$process$","of","$breaking$","$down$","a","$piece$","of","$text$","$into$","$smaller$","$units$",",",","$called$","$tokens$","."$]$

In this example, the function $tokenize(S)$ takes the input string S and produces the output sequence of tokens T, which consists of individual words and punctuation marks.

3.2.2 TF-IDF

TF-IDF (term frequency-inverse document frequency)[23] is a numerical statistic used to measure the importance of a term in a corpus of documents. It is calculated by multiplying two measures: term frequency (TF) and inverse document frequency (IDF). The term frequency (TF) of a term is the number of times it appears in a document divided by the total number of terms in the document. It can be calculated as follows:

$$TF(t,d) = count\ of\ "t"\ in\ "d"/number\ of\ words\ in\ "d" \tag{18}$$

An indicator of a term's rarity or prevalence throughout the full corpus of texts is its inverse document frequency (IDF). It is calculated as the logarithm of the total count of documents in the corpus divided by the count of documents that have the term. The formula for IDF is as follows:

$$IDF(term, corpus) = log(N/occurrences\ of\ t\ in\ N\ documents) \tag{19}$$

where N is the overall count of documents or corpus.

Once we have the TF and IDF values for each term in a document, we can calculate the TF-IDF score for that term. The TF-IDF score is the product of the term's TF

and IDF values. The formula for TF-IDF is as follows:

$$TF - IDF(term,\ document,\ corpus) = TF(term,\ document) * IDF(term,\ corpus) \tag{20}$$

This calculation results in a numerical score that reflects the importance of the term in the document and the corpus. Terms with high TF-IDF scores are considered important and relevant to the document, while terms with low scores are less important and less relevant.

3.2.3 Bag of Word

(BoW) Bag of Words [24] is a well known Natural Language Processing (NLP) technique used to represent text data as numerical vectors, which can be easily processed by machine learning algorithms. In this technique, a document is represented as a collection of its constituent words without taking into account their order or grammar. A vocabulary is first created by identifying all the unique words that appear in the corpus of documents. Then, each document is represented as a vector of word counts, where the value of each element in the vector represents the frequency of that word in the document. Here's a step-by-step explanation of the Bag of Words technique with maths:

- **Collecting the vocabulary:** Let's consider a corpus of documents, $D = d1, d2, ..., dn$. The first step is to collect all the unique words that appear in the corpus to create a vocabulary, $V = w1, w2, ..., wm$.
- **Document representation:** Each document is then represented as a vector of word counts, where the length of the vector is equal to the size of the vocabulary (i.e., m). Let's represent a document di as a vector $xi = [x1i, x2i, ..., xmi]$, where xji represents the frequency of the jth word in the ith document.
- **Counting word frequencies:** To calculate the frequency of each word in a document, we can simply count the count of times which a word appear in document. Let's denote the frequency of the jth word in the ith document as fji.
- **Computing the vector representation:** The vector representation of the i_th document can be computed as follows:

$$xi = [x1i, x2i, ..., xmi] \tag{21}$$

where $xji = fji$ This is the bag-of-words representation of the document. For example, lets assume we have a corpus of three different documents:
$d1$: "*The quick brown fox jumps over the lazy dog*"
$d2$: "*The dog chased the cat*"
$d3$: "*The cat climbed the tree*"
The vocabulary for this corpus is V = The, quick, brown, fox, jumps, over, lazy, dog, chased, cat, climbed, tree. Then, the bag-of-words representation of each document is:
$x1 = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]$
$x2 = [1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0]$
$x3 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1]$

In the above example, we can see that the bag-of-words representation of each document is a vector of word counts, where each element in the vector corresponds to a unique word in the vocabulary.

3.2.4 Named Entity Recognition

Named Entity Recognition (NER)[21] is a natural language processing (NLP) technique that involves identifying and classifying named entities (i.e., specific objects, people, places, organizations, and other types of entities) in unstructured text data. NER involves the use of machine learning algorithms and statistical models to analyze textual data and identify the named entities within it.

The goal of NER is to automatically recognize and classify entities within text, and to extract relevant information from these entities. For example, in a sentence like "I am going to New York to visit my friend John Smith", NER would identify "New York" as a location and "John Smith" as a person, and could potentially link this information to other relevant data sources.

NER can be used in a wide range of applications, including information retrieval, text classification, and machine translation, among others. By automatically identifying and classifying named entities within text, NER can help improve the accuracy and efficiency of natural language processing tasks, as well as facilitate the extraction of relevant information from large amounts of unstructured data.

3.2.5 Word Embedding

Word embedding[21] is a technique in natural language processing (NLP) that represents words as high-dimensional vectors in a continuous space, where each dimension represents a different feature of the word. The goal of word embedding is to capture the semantic and syntactic relationships between words and to represent them in a way that is practical for downstream NLP errand, like sentiment analysis, language translation, and named-entity- recognition.

The most commonly used method for word embedding is the Word2Vec algorithm, which uses a neural network to learn the vector representations of words from large amounts of text data. Specifically, Word2Vec trains a neural network to predict the probability of a word appearing in a given context, where the context is defined as a fixed number of words before and after the target word. The neural network is trained by optimizing the negative log-likelihood of the training data. This optimization is performed using stochastic gradient descent, which involves computing the gradient of the loss function with respect to the model parameters and updating the parameters in the direction of the negative gradient.

Once the neural network is trained, the learned weights of the input layer (which correspond to the word vectors) are used as the word embeddings. Using methods like principle component analysis (PCA) or t-SNE, these embeddings may be visualised in a lower-dimensional space or utilised as input features for subsequent NLP tasks.

3.2.6 Text Classification

Text classification[21], also known as text categorization, is a natural language processing (NLP) technique that involves categorizing a piece of text into one or more predefined categories based on its content. Text classification is a fundamental task in NLP, as it enables various applications such as sentiment analysis, spam filtering, news categorization, and content recommendation systems.

The process of text classification involves several steps, including:

- **Preprocessing:** The first step in text classification is to preprocess the text, which involves tokenizing the text, removing stop words, stemming or lemmatizing the words, and converting the text to a numerical representation such as a vector.
- **Feature extraction:** The next step is to extract features from the preprocessed text. This involves identifying important words or phrases that are indicative of the content and represent them in a numerical format.
- **Training a classifier:** Once the features have been extracted, a classifier is trained using a machine learning algorithm. The classifier learns to map the input features to the predefined categories.
- **Evaluation:** The performance of the classifier is calculated using a array of labeled data which has not been used for training. This calculation helps to determine the accuracy of the classifier and whether it is suitable for the intended application.

Some common machine learning algorithms used for text classification include Naive Bayes, Support Vector Machines (SVM), Decision Trees, and Neural Networks. These algorithms differ in their complexity and performance, and the choice of algorithm depends on the specific task and set of data.

3.3 Deep Learning

3.3.1 CNN

Convolutional Neural Networks (CNNs) are a type of deep learning model that is commonly used for image recognition, classification, and segmentation tasks.[25]hey are particularly effective at identifying patterns and features in images.

CNNs consist of several layers, each of which performs a different operation on the input data. The most common types of layers in a CNN are convolutional layers, pooling layers, and fully connected layers.

1. **Convolutional Layer:** It is the main building block of a CNN. It performs a convolution operation on the input data using a set of learnable filters. The output of this operation is a feature map that represents the presence of specific features in the input data.

   The convolution operation is defined as follows:

   $$z[i,j] = (f * x)[i,j] = sum(m)sum(n)f[m,n] * x[i-m, j-n] \tag{22}$$

   where f is the filter, x is the input data, z is the output feature map, and m and n are the indices of the filter.

2. **Pooling Layer:** This layer reduces the spatial dimensions of the input data by performing a downsampling operation. The most common type of pooling operation is max pooling, which takes the maximum value in each pooling region.

   The max pooling operation is defined as follows:

$$y[i,j] = max[x[m,n] : i * s <= m < (i+1) * s, j * s <= n < (j+1) * s] \quad (23)$$

   where x is the input data, y is the output of the pooling operation, and s is the stride.
3. **Fully Connected Layer:** This layer is a commonly and widely used neural network layer that takes the flattened output of the previous layers as input and produces a final output. This layer is used to classify the input data into different classes.

   TEach layer's output is sent via an activation function, such the Rectified Linear Unit (ReLU), which can be given as follows:

$$f(x) = max(0, x) \quad (24)$$

4. **Dropout:** A Dropout layer is another prominent feature of CNNs. The Dropout layer acts as a mask, eliminating some neurons' contributions to the subsequent layer while maintaining the functionality of all other neurons.

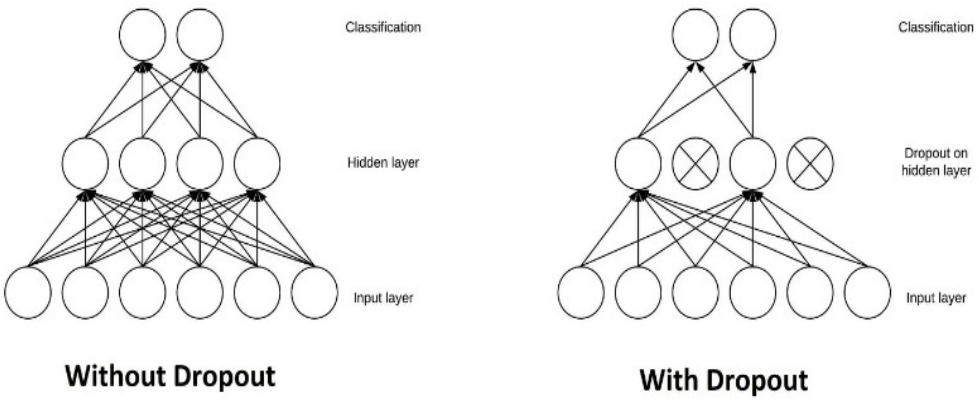


Fig. 8 Drpout[26]

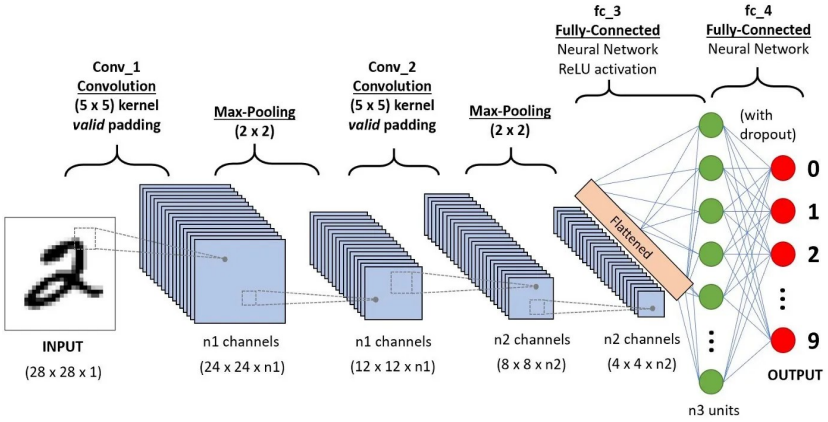


Fig. 9 Convolutional Layers [26]

The diagram shows a typical convolution neural network (CNN) architecture. The input image is processed by a convolution layer, which applies a set of filters to generate a feature map. The output of the convolution layer is then passed through a Rectified Linear Unit ReLU activation function to introduce non-linearity.

The subsequent pooling layer applies max pooling to the output of the convolution layer, which reduces the spatial dimensions of the feature map.

The final stage involves passing the output of the pooling layer through a fully connected layer, which is responsible for generating the network's final output.

3.3.2 RNN

Recurrent Neural Networks (RNNs) are a type of neural network that is well-suited for processing sequential data of varying lengths. They have proven to be highly effective for tasks such as speech recognition, machine translation, and language modeling, where both input and output data are sequences.

Unlike traditional feedforward neural networks, RNNs have the ability to maintain an internal state, which enables them to process sequences of variable lengths. At each timestep, the output of the network is influenced not only by the input at that timestep but also by the network's state from the previous timestep.

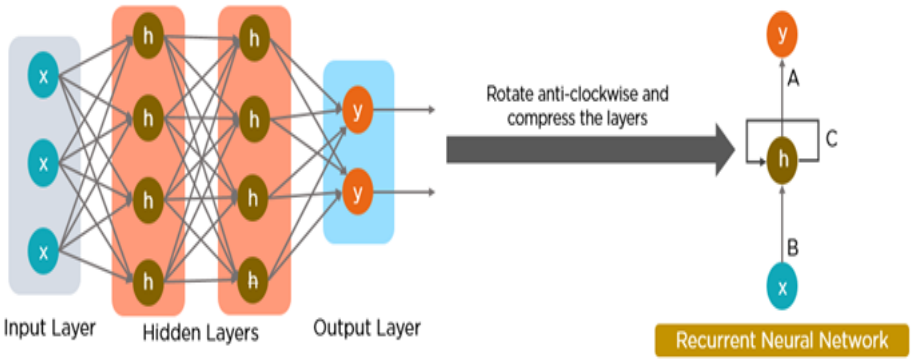


Fig. 10 Layers in RNN[27]

One of the key features of RNNs is that they can learn long-term dependencies in sequences. This is achieved through the use of a recurrent connection, this feature permits the transfer of data from one point in time to another.

Architecture of a Traditional RNN

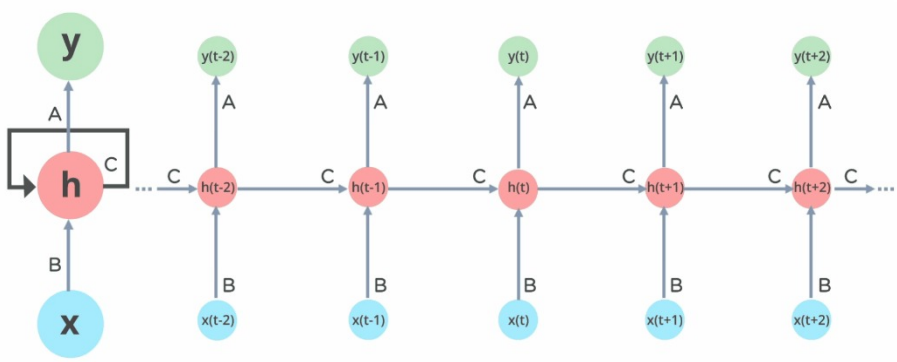


Fig. 11 Architecture of RNN [27]

The Recurrent Neural Network (RNN) is a specialized architecture intended to handle sequential data processing. Its structure comprises of three primary components: an input layer, a hidden layer, and an output layer.

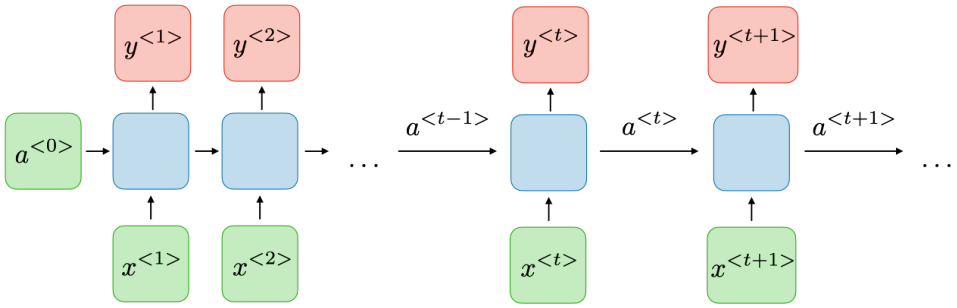


Fig. 12 Time specific architecture of RNN [27]

In the Recurrent Neural Network (RNN), the input layer receives the sequential input data. The network processes the input at each time step, and based on the input and the previous state of the network, it produces an output. The output layer, which can have various configurations depending on the task, usually generates the network's final output.

The hidden layer is a crucial component of the RNN, responsible for its ability to handle variable-length input sequences. The hidden state of the RNN is updated at each timestep, taking into account the current input and the previous state. This enables the network to remember the past inputs and utilize that information to influence the output of the network at the current timestep.

In mathematical terms, the computation in the hidden layer of an RNN can be represented as follows:

$$ht = f(Wh * ht - 1 + Wx * xt + b) \tag{25}$$

in which ht is the hidden state at given timestep t, f is a non-linear activation function (such as the sigmoid or tanh function), Wh and Wx are weight matrices, xt is the input at given timestep t, and b is a bias term.

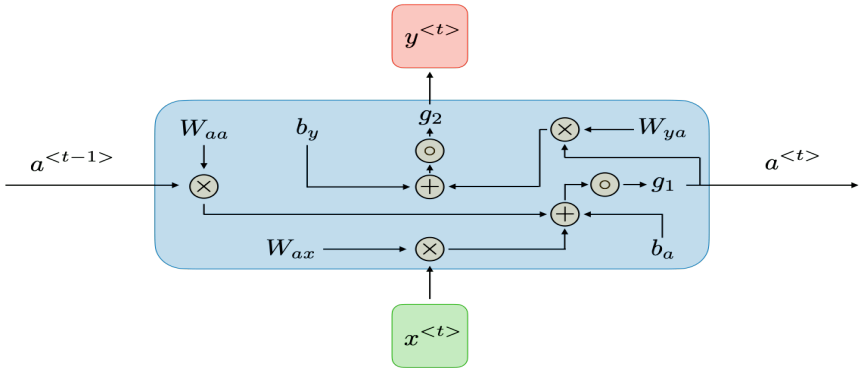


Fig. 13 Components of RNN [27]

The output layer of an RNN generates the final output of the network and can have different configurations depending on the specific task. For instance, in a language modeling task, the output layer could comprise a softmax function that produces a probability distribution over the next word in the sequence.

Backpropagation Through Time (BPTT)

Backpropagation Through Time (BPTT) is the conventional algorithm utilized for training Recurrent Neural Networks (RNNs) [28]. BPTT involves first unfolding the network through time, creating a sequence of interconnected layers that correspond to each time step in the input sequence. The weights of the network are then updated using the chain rule of differentiation, similar to the standard backpropagation algorithm. However, since the network is unfolded in time, the chain rule must be applied iteratively to all prior time steps, giving rise to the term "backpropagation through time."

A major challenge associated with Backpropagation Through Time (BPTT)[28] is the instability or vanishing of gradients as they propagate backwards through time. This problem is referred to as the vanishing gradient problem and can hinder the network's ability to learn long-term dependencies.

A method called Truncated Backpropagation Through Time (TBPTT) is commonly employed to mitigate the vanishing gradient issue in BPTT. TBPTT limits the number of time steps over which the gradient is backpropagated before truncation. This helps to alleviate the impact of the vanishing gradient problem and improve the network's ability to model long-term dependencies.

Two issues of Standard RNNs Standard RNNs[25] (vanilla RNNs) suffer from two main issues:

1. **Vanishing gradients:** The vanishing gradient problem arises when the gradients utilized to update the weights during training become excessively small, resulting in difficulties for the network to learn long-term dependencies. This happens because the gradients are backpropagated through multiple time steps, causing them to become increasingly smaller. As a result, the weights associated with earlier time steps are updated very slowly or not at all. This can result in the network being unable to remember information from earlier time steps, which is a significant problem when dealing with sequential data.
2. **Exploding gradients:** During the training of a neural network, the opposite of the vanishing gradient problem is the exploding gradient problem. This issue arises when the gradients become exceedingly large, leading to instability during training. This can cause the weights to update too much, leading to instability and difficulty in training. In extreme cases, the gradients can become so large that they cause the network to overflow and produce NaN values.

3.3.3 LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that is specially designed to effectively learn long-term dependencies. They have demonstrated excellent performance across a diverse range of problem domains and are currently a popular choice in machine learning applications.

LSTMs were intentionally specialized to overcome the issue of the long-term dependencies in RNNs. Unlike traditional RNNs, LSTMs can retain information for extended periods, making them capable of learning from sequences with gaps between relevant information. Recurrent neural networks, including LSTMs, are composed of repeating modules of neural networks. However, while the repeating module in traditional RNNs typically consists of a single tanh layer with a simple structure, LSTMs have a more complex internal structure that enables them to selectively choose an infromtaion to retain or discard over time.

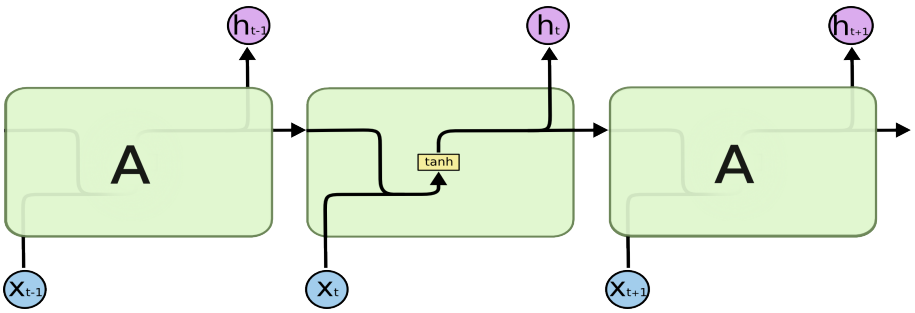


Fig. 14 LSTM [29]

While LSTMs share the chain-like structure of traditional RNNs, they are distinct in several key ways. LSTMs consist of four neural network layers, as opposed to just one in traditional RNNs, and these layers interact in a highly unique manner. This unique structure allows LSTMs to selectively retain or discard information over time, making them highly effective at handling long-term dependencies in sequential data.

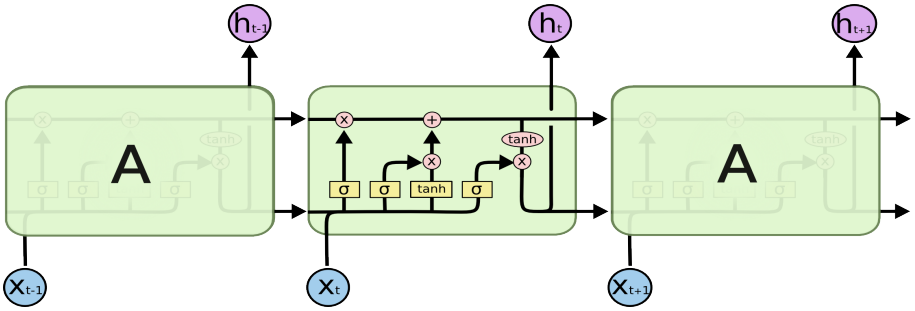


Fig. 15 Structure of LSTM [29]

The diagram depicts a neural network, specifically an LSTM, where each line denotes a vector passed from the output of one node to the input of another. The yellow boxes signify neural network layers that are learned during training, while pink circles indicate point-wise operations like vector addition. Merging lines represent vector concatenation while forking lines denote copying the content of a vector and sending it to multiple locations. The diagram helps visualize how information is processed and flows through the neural network.

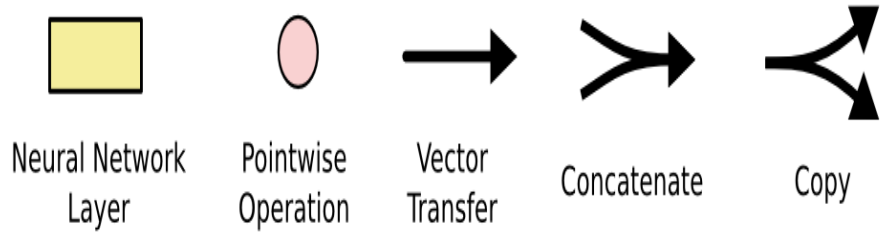


Fig. 16 Components of LSTM [29]

The cell state is a crucial component of the LSTM network architecture, and it is represented by a horizontal line that runs along the top of the diagram.

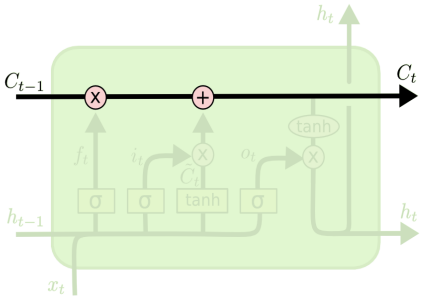


Fig. 17 cell state [29]

The cell state in an LSTM network can be thought of as a conveyor belt of information that passes through the entire network. With only a few linear interactions, the cell state remains relatively unchanged as it moves down the chain. However, the LSTM has the ability to modify cell state by selectively adding or removing information through the use of "gates." These gates comprise a point-wise multiplication operation and a layer of sigmoid neural networks that output values within zero and one, which control the amount of information that should be allowed through. Zero means to "not allow anything," whereas one means to "allow everything."

An LSTM has three gates that protect and control the cell state.

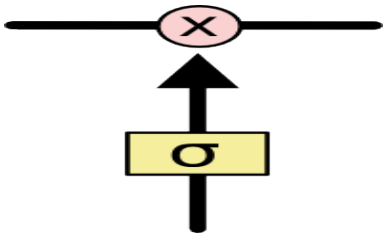
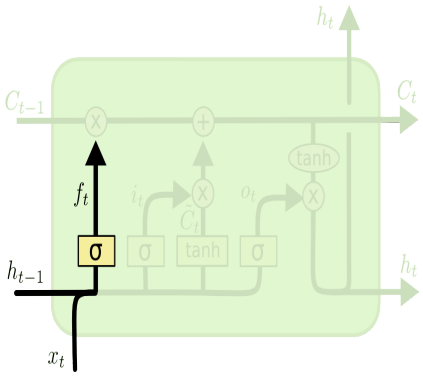


Fig. 18 Gates of LSTM [29]

Step-by-Step LSTM[30] Look Through

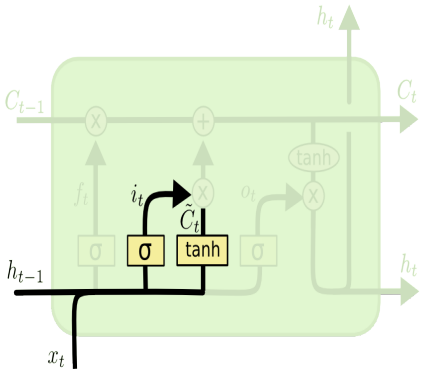
In the **initial step** of the LSTM process, the "forget gate layer" is employed to determine which information in the cell state should be removed. This layer uses a sigmoid function and inspects both the previous hidden state (ht-1) and the current input (xt) to produce an output value between 0 and 1 for each value in the cell state (Ct-1). A value of 1 indicates that the corresponding information should be retained, while a value of 0 indicates that it should be discarded entirely.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Fig. 19 Step 1 [29]

In the **subsequent steps** in the LSTM process, the focus is on determining what new information should be stored in the cell state. This process is composed of two stages. Firstly, a sigmoid layer known as the "input gate layer" is utilized to identify the values that need to be updated. Subsequently, a "tanh" layer is employed to produce a vector of new candidate values (Ct) that can be added to the cell state. Finally, the output of both stages is combined to create an update to the cell state.

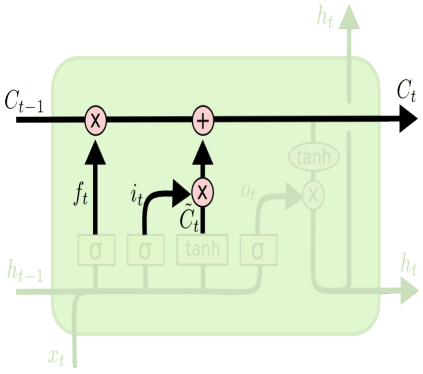


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Fig. 20 Step 2 [29]

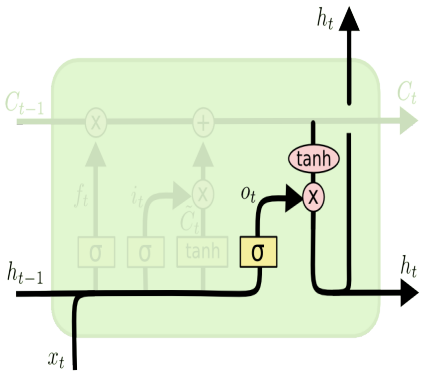
To create a new cell state (Ct) in the LSTM process, we update the previous cell state (Ct-1) using the decisions made in the previous steps. The old state is first multiplied by ft, which ensures that the values marked for removal in the previous step are forgotten. Next, it is added to the product of the "input gate layer" sigmoid function and C t, which represents the new candidate values determined by the earlier stages. The degree of update for each value in the cell state is based on the value outputted by the sigmoid function of the input gate layer.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Fig. 21 Step 3 [29]

The **final step**in the LSTM process involves selecting the relevant parts of the cell state to be outputted. The output is derived from the state of the cell, but filtered to include only necessary information. Firstly, the cell state is passed through a sigmoid layer that decides which parts of the state will be included in the output. Next, the output of the sigmoid gate is scaled by the cell state to ensure that only the selected portions are included in the output. This scaling operation is necessary to limit the output values between -1 and 1.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Fig. 22 Step 4 [29]

3.3.4 Bi-Lstm

The BiLSTM[31] model is a modification of the conventional LSTM model, intended to handle sequential data by retaining a memory state capable of capturing long-term dependencies. The key difference between BiLSTM and LSTM is that BiLSTM processes input sequences in both forward and backward directions simultaneously.

The BiLSTM architecture consists of two LSTM layers, where one processes the input sequence in the forward direction, and the other in the backward direction. By concatenating the output of each LSTM layer, the BiLSTM layer produces the final output. This allows the model to capture the past and future contexts of the input sequence simultaneously, resulting in better performance on tasks requiring a comprehensive understanding of the input sequence.

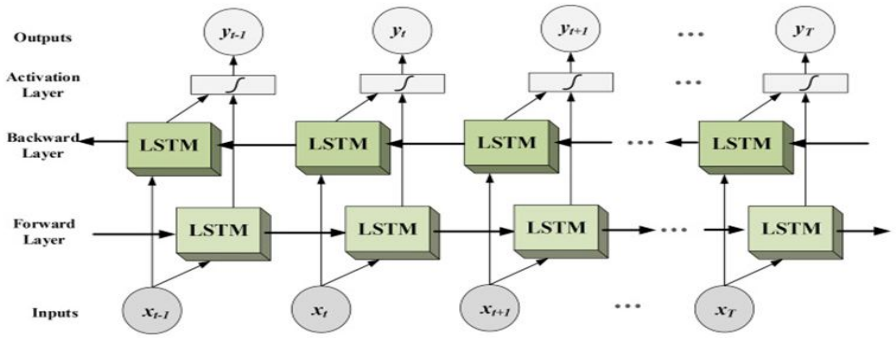


Fig. 23 Bi-LSTM flow [29]

The flow of information in BiLSTM goes both forward and backward layers, which is beneficial for tasks requiring sequence to sequence. This includes speech recognition, text classification, and forecasting models.

A significant advantage of BiLSTM is its capability to capture long-term dependencies in both directions of the input sequence. Therefore, it is useful in errands such as sentiment analysis, machine translation, and speech recognition. BiLSTM has also been widely used in NLP errands like entity name recognition, part-of-speech tagging, and text classification

3.3.5 BERT

According to Devlin et al. (2018), BERT is a machine-learning framework that utilizes the transformer architecture. The transformer architecture connects each output to each input and uses attention to weigh them and determine their relationship. BERT leverages pre-training on large datasets for unsupervised language modeling, enabling more sophisticated models to better understand sentence meaning. After pre-training, the model can be fine-tuned for specific data monitoring tasks to enhance performance.

During this stage, there are two strategies that can be employed: the top-down or bottom-up approach. In contrast, Elmo adopts a distinct design that combines multiple models and pre-trained models for language representation for each task.

BERT utilizes a bidirectional transformer encoder to understand text, hence its name. The model analyzes the context surrounding words to determine their meaning. By examining the terms before and after a given word, BERT establishes the relationship between them, a technique known as bidirectional attention.

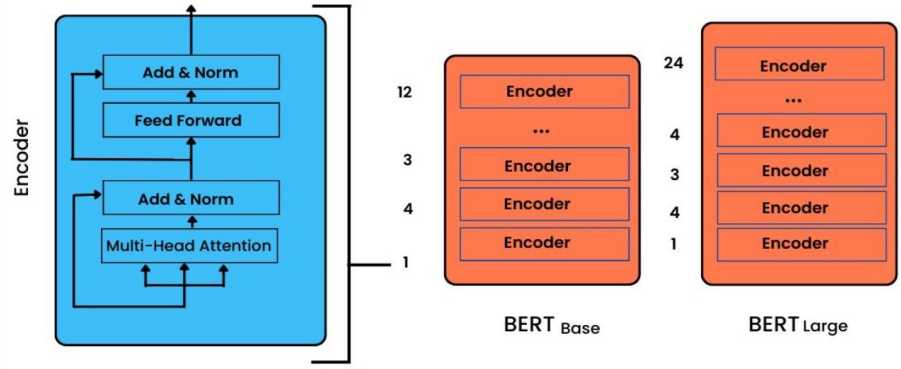


Fig. 24 Components of BERT [32]

The framework of the BERT NLP model

The BERT design consists of a set of transformer encoders, each of which has two sub-layers: a position-wise feedforward network and a multi-head self-attention mechanism. The model can concentrate on many input sequence segments concurrently thanks to the multi-head self-attention method. The position-wise feedforward network, meanwhile, performs an individual non-linear conversion at each place in the sequence.

In addition to the transformer encoders, BERT also includes special tokens -

1. CLS token[33], which is added to the beginning of each input sequence. This token is used to represent the entire input sequence in downstream tasks.
2. SEP token[33], SEP is a special token used to denote sentence separation at the conclusion of the input sequence. SEP is given the token representation of 102 in BERT. The model can now distinguish between the beginning of one phrase and its finish within the input sequence.

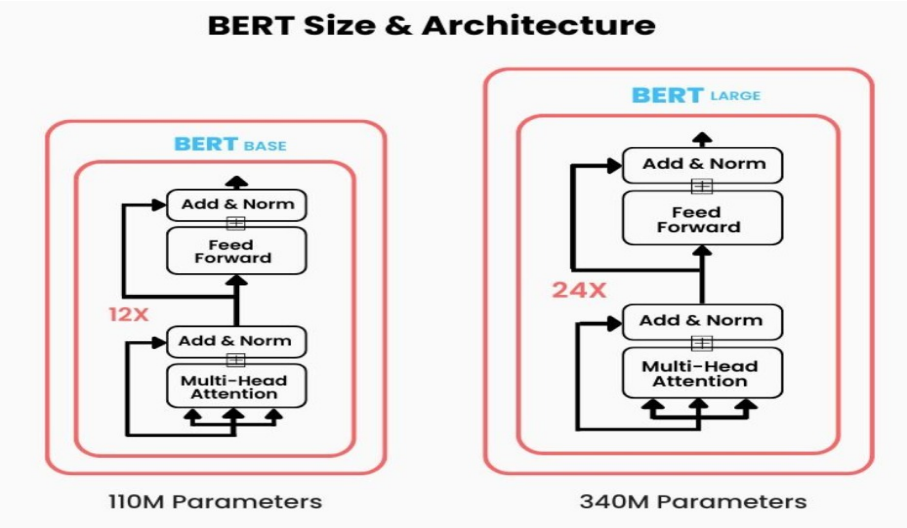


Fig. 25 Architecture of BERT [32]

The below table represents the components of BERT, their definitions:

Table 2 BERT Components and their definitions

Components	Definition
Parameters	Number of readable variables available in the model.
Hidden Size	It is the layers of different mathematical functions between input and output, assigns weight to produce the desired result.
Transformer Layers	Defines Number of transformer blocks. It will transform a sequence of word representations into a contextualized text or numerical representation.
Processing	Specifies the type of processing unit used for training the model.
Attention Heads	Specifies the block size of transformer .
Length of training	Time taken to train the model.

How does BERT work?
Following are thee steps that occurs in BERT -

1. **masked language:** By randomly masking part of the input sequence's tokens, masked language modelling trains the model to predict missing tokens by analysing the context around them. The model is compelled by this technique to get a deeper understanding of the relationships between words and their context in a phrase. By taking into account the words around the missing tokens, this method teaches the model to predict their absence.

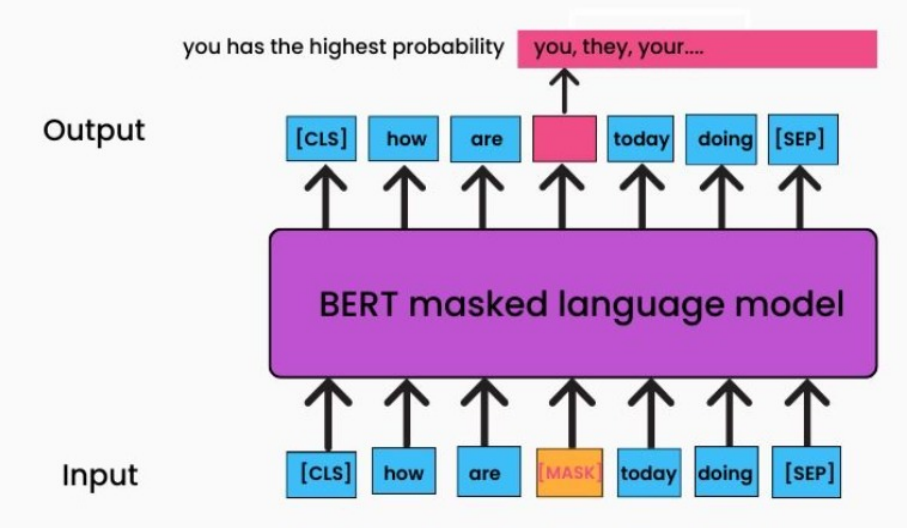


Fig. 26 Masking [32]

2. **Next Sentence Prediction:** The model has been trained to identify whether or not two input sentences follow one another. The model needs help from this assignment to understand the relationships between sentences and the context in which they are used. The model learns about the connections between sentences by completing this job.

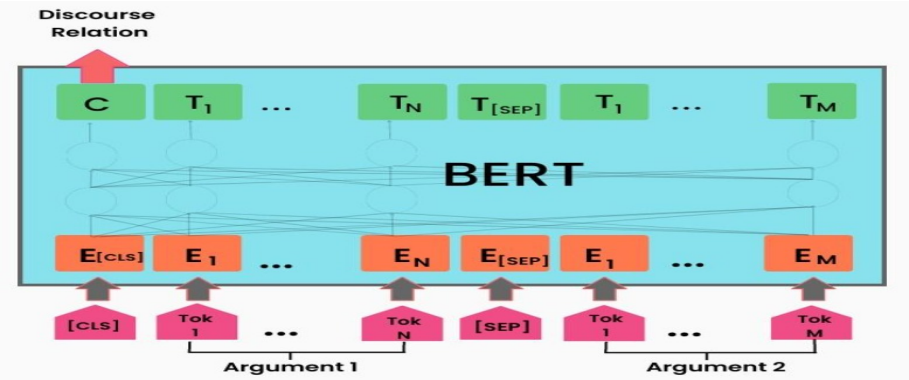


Fig. 27 Next Sentence prediction[32]

3. Using annotated data, the model may be fine-tuned for a particular NLP job after pre-training. BERT may be customised for a variety of tasks, including sentiment analysis, text categorization, and question answering. To fine-tune the model for the specific job at hand, the model's weights are changed based on the labelled data.
4. BERT can accept input sequences of different lengths and understand how the words in a phrase are connected because to the transformer architecture it uses. While multi-head attention helps the model to recognise many sorts of links between words, self-attention allows the model to prioritise different portions of the input sequence as appropriate.

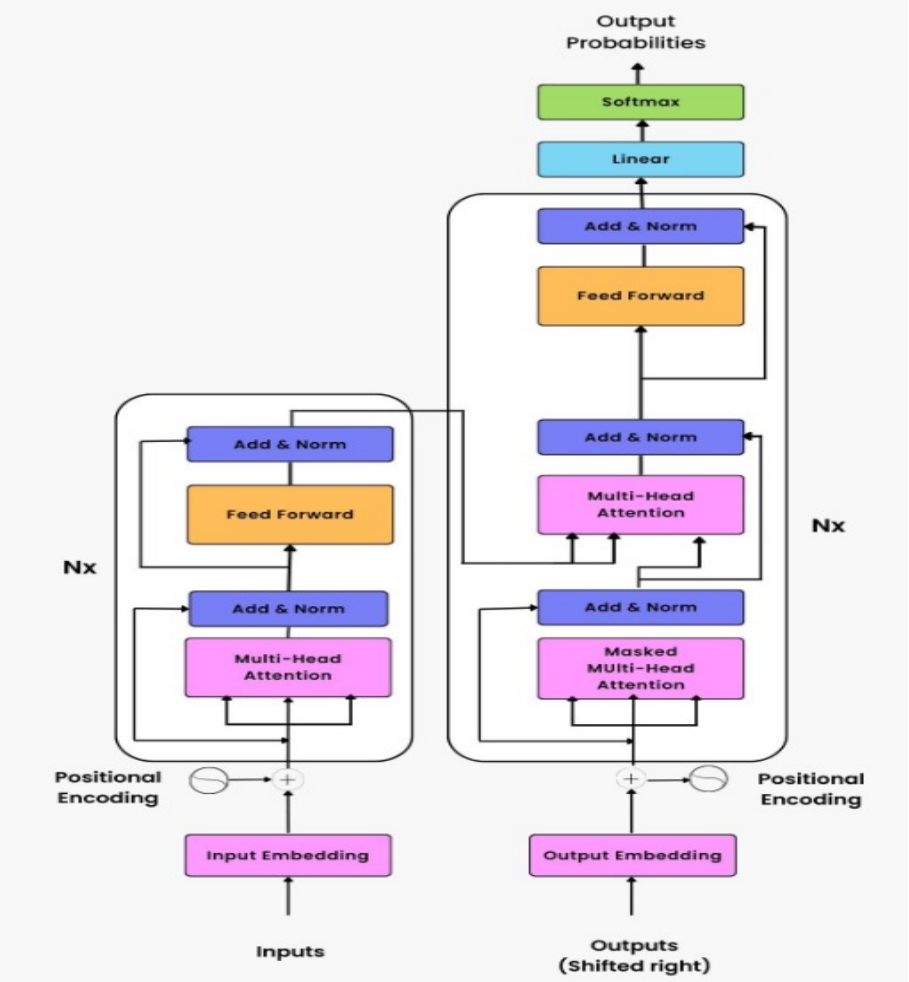


Fig. 28 Flow and Architecture of BERT [32]

3.4 RoBERTa

RoBERTa[34] (short for Robustly Optimized BERT approach) is a natural language processing (NLP) model developed by Facebook AI and released in 2019. RoBERTa is based on the BERT model, which Google released in 2018 and is also a very popular NLP model.

RoBERTa is a transformer-based model that pretrains on a lot of text material using self-supervised learning. The model is trained using a masked language modelling task and a next-sentence prediction task on a vast quantity of textual data, including books, Wikipedia, and web pages. This allows RoBERTa to learn to understand the meaning of words and the context in which they are spoken in a much more robust and accurate way than previous models.

On a variety of NLP benchmarks, including the GLUE benchmark and the SQuAD 2.0 dataset, RoBERTa has produced state-of-the-art results. Its improved performance is due to several optimizations made to the BERT architecture, including training on longer sequences, using dynamic-masking , and removing the task of predicting the next sentence.

Why it is better than BERT?

RoBERTa is considered an improvement over BERT because it incorporates several modifications to the original BERT model that have been shown to improve performance on a variety of NLP tasks. Here are a few reasons why RoBERTa is generally considered to be better than BERT.[35]

1. **Longer training:** RoBERTa was trained on more data for longer than BERT, allowing it to learn more from the vast amount of available text. Specifically, RoBERTa was trained on 160GB of text, while BERT was trained on 3.3GB.
2. **Dynamic masking:** In BERT, the masking of words was done statically, meaning that the same words were always masked during training. RoBERTa uses dynamic masking, which randomly masks different words at different times during training, making the model more robust to variations in the input.
3. **Larger batch size:** RoBERTa can handle a larger batch size than BERT due to the dynamic masking approach, which allows for more efficient training and improved performance.
4. **No next sentence prediction task:** RoBERTa doesn't use the next sentence prediction task that was used in BERT, which has been shown to be less useful in practice.
5. **Preprocessing optimization:** RoBERTa uses a byte-level BPE (Byte Pair Encoding) tokenizer, which is more efficient than the word-piece tokenizer used in BERT, allowing for faster preprocessing of text data.

4 Different data sets available for fake news detection [36]

Dataset	News domain	Application purpose	Type of disinformation	Language	Size	News content type	Rating scale	Media platform	Spontaneity	Availability	Extraction time
NELA-GT-2018 (NÃ¸rregaard Horne AdalÄ± 2019)	Politics	Fake detection	Fake news articles	English	713000 articles	Text	2 values	Mainstream	Yes	Yes	Yes(February 2018 - November 2018)
TW_info (Jang Park Seo 2019)	Politics	Fake detection	Fake news articles	English	3472 articles	Text	2 values	social media (Twitter)	Yes	Yes	Yes (January 2015 - April 2019)
FCV-2018 (Papadopoulou et al. 2019)	Society	Fake detection	Fake news content	English Russian Spanish Arabic German Catalan Japanese and Portuguese	380 videos and 77258 tweets	Videos and Text	2 values	social media (YouTube Facebook Twitter)	Yes	Yes	Yes (April 2017 - July 2017
Verification Corpus (Boididou et al. 2018)	Society	Veracity classification	Hoaxes	English Spanish Dutch French	15629 posts	Text images videos	2 values	social media (Twitter)	Yes	Yes	Yes (2012-2015)
CNN / Daily Mail summarization dataset (Jwa et al. 2019)	Politics society crime sport business technology health	Fake detection	Fake news articles	English	287000 articles	Text	4 values	Mainstream	Yes	Yes	Yes (April 2007 - April 2015)
Zheng et al.âs dataset (Zheng et al. 2018)	Society	Clickbait detection	Clickbait	Chinese	14922 headlines	Text	2 values	Mainstream + social media (Wechat)	Yes	Yes	No
Tam et al.âs dataset (Tam et al. 2019)	Politics technology science crime fraud and scam fauxtography	Rumor detection	Rumors	English	1022 rumors and 4 million tweets	Text	2 values	social media (Twitter)	Yes	Yes	Yes (May 2017 - November 2017)

Dataset	News Domain	Application purpose	Type of disinformation	Language	Size	News content type	Rating scale	Media platform	Spontaneity	Availability	Extraction time
BuzzFace (Santia Williams 2018)	Politics society	Veracity classification	Fake news articles	English	2263 news	Text	4 values	Social media (Facebook)	Yes	Yes	Yes (September 2016)
FacebookHoax (Tacchini et al. 2017)	Science	Fake detection	Hoaxes	English	15500 posts	Text	2 values	Social media (Facebook)	Yes	Yes	Yes (July 2016 - December 2016)
LIAR (Wang 2017)	Politics	Fake detection	Fake news articles	English	12836 short statements	Text	6 values	Mainstream + social media (Facebook and Twitter)	Yes	Yes	Yes (2007-2016)
FEVER (Thorne et al. 2018)	Society	Fact checking	Fake news articles	English	185445 claims	Text	3 values	Mainstream	No	Yes	No
FakeNewsNet (Shu et al. 2018)	Society politics	Fake detection	Fake news articles	English 422 news	Text images	2 values	Mainstream + social media (Twitter)	Yes	Yes	No	
Benjamin Political News Dataset (Horne Adali 2017)	Politics	Fake detection	Fake news articles	English	225 stories	Text	3 values	Mainstream	Yes	Yes	Yes (2014-2015)
Spanish fake news corpus (PosadasDurÃ¡n et al. 2019)	Science Sport Economy Education Entertainment Politics Health Security Society	Fake detection	Fake news articles	Spanish	971 news	Text	2 values	Mainstream	Yes	Yes	Yes (January 2018 - July 2018)

Dataset	News Domain	Application purpose	Type of disinformation	Language	Size	News content type	Rating scale	Media platform	Spontaneity	Availability	Extraction time
BuzzFeed News dataset (Horne Adal 2017)	Politics	Fake detection	Fake news articles	English	2283 news samples	Text	4 values	Social media (Facebook)	Yes	Yes	Yes (2016-2017)
MisInfoText dataset (Torabi Taboada 2019)	Society	Fact checking	Fake news articles	English	1692 news articles	Text	4 values for BuzzFeed and 5 values for Snopes	Mainstream	Yes	Yes	No
FNC-1 dataset (Riedel et al. 2017)	Politics society technology	Fake detection	Fake news articles	English	49972 articles	Text	4 values	Mainstream	Yes	Yes	No
Spanish fake news corpus (PosadasDurÃ¡n et al. 2019)	Science Sport Economy Education Entertainment Politics Health Security Society	Fake detection	Fake news articles	Spanish	971 news	Text	2 values	Mainstream	Yes	Yes	Yes (January 2018 - July 2018)
Fake_or_real_news (Dutta et al. 2019)	Politics society	Fake detection	Fake news articles	English	6337 articles	Text	2 values	Mainstream	Yes	Yes	No
TSHP-17 (Rashkin et al. 2017)	Politics	Fact checking	Fake news articles	English	33063 articles	Text	6 values for PolitiFact and 4 values for unreliable sources	Mainstream	Yes	Yes	No
QProp (BarrÃ³nCedeno et al. 2019)	Politics	Fact checking	Fake news articles	English	51294 articles	Text	2 values	Mainstream	Yes	Yes	No

5 Quantitative analysis of Datasets used in the Study and Associated Challenges

5.1 KAGGLE FAKE NEWS[37]

The Kaggle Fake News dataset is a collection of news articles that were used in a Kaggle competition to develop machine learning models for identifying fake news. The dataset contains over 20,000 news articles that are labeled as either "real" or "fake"[37], and it is intended to be used for training and testing machine learning models for identifying fake news.

Politics, entertainment, sports, and technology are just a few of the areas that are covered in the articles in the Kaggle Fake News dataset. A group of human annotators labelled the dataset after it was compiled from a variety of sources, including news websites and social media platforms.

The KFN dataset is publicly available on Kaggle, a platform for data science competitions and machine learning projects. It has been used by researchers and data scientists to develop and test machine learning algorithms for detecting fake news. However, it is important to note that the dataset may not be comprehensive or representative of all types of fake news, and caution should be used when interpreting the results of any analysis or model trained on the dataset.

The following characteristics or qualities are part of the Kaggle Fake News dataset:

id: A unique identification number assigned to each news article in the collection.
title: The heading for the news story.

author: The journalist who wrote the news story.

text: The news article's main body of text.

news article's "real" or "fake" status is indicated by a binary designation.

The id, title, author, and text features are all text-based and contain information about the content and structure of the news articles. The label feature is a binary variable that indicates whether the article is real or fake.

The text-based features in the dataset can be used to train and test natural language processing (NLP) models for identifying patterns or characteristics of fake news. These models can then be used to classify new news articles as real or fake, based on their content and structure.

5.2 ISOT[38]

The ISOT[38], which includes both authentic and false news pieces, is a publicly accessible dataset. The Canadian university of Victoria's research team produced the dataset, which will be used to study the issue of identifying false news. There are two CSV files in it.

The first file is "True.csv" which contains more than 12,600 articles from reuter.com.

The second file is"Fake.csv" which contains more than 12,600 articles from other different fake news outlet resources.

The dataset contains articles from a variety of sources, including mainstream news outlets and fake news websites. The articles are labeled as either real or fake and

are divided into two subsets: a training set and a test set. The ISOT Fake News dataset is one of the largest and most comprehensive datasets of its kind and has been used by researchers in a wide range of fields, including machine learning, natural language processing, and journalism. However, it is noteworthy that the data-set has been criticized by some for its lack of diversity in terms of news sources, as well as for potential biases in the labeling process. When using a dataset for study, it is crucial to carefully analyse its constraints and inherent biases.

5.3 liar liar.. Dataset[37]

The phrase "liar liar pants on fire" is often used as a playful taunt when someone is being ripped off. In recent years, however, In the domains of NLP and ML, the phrase now has a new meaning. thanks to the dataset of the same name. LLPOF[37] is a set of statements labeled with truth levels, from true to false, with various intermediate levels such as rarely true, mostly true, mostly false, etc. The LLPOF dataset was created to facilitate research on automatic fraud detection, a difficult and important task in natural language processing. Fraud detection involves analyzing cues and linguistic elements to identify misleading statements, which is useful in many situations such as detecting fraud, spam, and misinformation in social media, news, and advertising. However, automatic fraud detection is a difficult problem because people can cheat in very subtle ways and language is often ambiguous and context-dependent.

To create the LLPOF dataset, the authors collected statements from a variety of sources, such as PolitiFact, FactCheck.org, and other fact-checking sites, and then had human annotators assess the veracity of the statements based on their own research and knowledge. Annotators also assessed statement engagement, i.e., whether the statement supported, refuted, or was irrelevant to the topic. The resulting dataset contains more than 12,800 utterances Each of the statements was divided into six classifications, ranging from True to Pants on Fire, based on how true they were. The dataset consists of 14 columns with features like Statement Json ID, Labels, Statement - Title and Often the Statement, Subject representing the subjects of the statement, Speaker representing the source of the statement, Speaker's job, US state of residence, Party they are affiliated with, the number of truth values for the speaker including the current statement, and the statement context representing the place or location of the statement. The LLPOF dataset has become a popular benchmark for evaluating fraud detection algorithms, and several studies have used it to compare different approaches, such as rule-based models, machine learning, and machine learning. depth. For example, some studies use craft features such as the frequency of certain words or the use of negation and hedging, while others use neural networks to learn representations of statements and their context. Despite its usefulness, the LLPOF dataset has some limitations that researchers should be aware of. For example, the dataset is biased toward political speech and contains mostly speech from American politicians and the media. Moreover, the labeling of datasets is subjective and depends on the judgment of the annotators, which may vary depending on their background, beliefs and biases. Finally, the dataset does not capture the complex nature of deception, such as sarcasm or sarcastic statements, which may be scored differently by different annotators.

In summary, the LLPOF dataset is an invaluable resource for fraud detection and natural language processing researchers, providing a large and diverse collection of labeled sentences with varying degrees of authenticity and involvement. However, researchers should be aware of the bias and subjectivity of datasets and use them in conjunction with other datasets and assessment measures. Furthermore, the LLPOF dataset is a reminder that fraud is a complex and nuanced phenomenon that requires a multidisciplinary approach to understand and detect.

5.4 WELFake Dataset[39]

The WELFake dataset[39] is a collection of fake news articles, created by researchers at the University of Adelaide, Australia. The dataset contains around 8000 fake news articles, which were generated using various state-of-the-art text generation techniques, including generative adversarial networks (GANs), language models, and neural machine translation. The articles in the WELFake dataset cover a wide range of topics, including politics, science, technology, and entertainment. They are labeled according to their level of credibility, with some articles being more believable than others. The dataset was created to help researchers develop more effective methods for detecting and combating false news.

WELFake is publicly available and can be copied to your local system from the webiste of Adeliade university. However, since the articles in the dataset are fake, it is important to use caution when analyzing them, and to avoid spreading misinformation.

6 Literature Survey

Table 3 : Literature Survey for Machine Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[21]	Development of Fake News Model using Machine Learning through Natural Language Processing	The detection of fake news poses various challenges due to the nature of the content, which often contains misleading or false information	Passive Aggressive, Naïve Bayes, and Support Vector Machine	The study used a dataset of news articles collected from different sources, including reliable and fake news websites	the study identified specific features that distinguish fake news from real news, such as the use of emotionally charged language and sensational headlines	the study suggests the need for more efficient techniques to combat fake news as social media platforms evolve, and new forms of fake news emerge
[23]	Comparison of various machine learning models for accurate detection of fake news	Identifying the fake news source, distinguishing it from satirical news, and identifying the credibility of the news source are some of the challenges faced	Naïve Bayes, Random Forest, Support Vector Machine (SVM), and Logistic Regression	the "Fake News" dataset from Kaggle	study found that the Random Forest model performed best, with an accuracy of 0.93 and an F1 score of 0.93. The SVM model also performed well, with an accuracy of 0.92 and an F1 score of 0.92	the study only considers the detection of fake news in the context of news articles, and future research can explore its effectiveness in detecting fake news in other contexts, such as social media posts or images
[16]	Fake news detection on social media using k-nearest neighbor classifier	detection of fake news on social media poses unique challenges, including the vast amount of information shared on these platforms, the speed at which it is shared, and the potential for the spread of misinformation	KNN classifier	dataset used for this study consists of 10,000 tweets collected from Twitter	study found that the KNN classifier achieved an accuracy of 0.871 and an F1 score of 0.871 for fake news detection on Twitter	One of the limitations of this study is the small size of the dataset, which may limit the generalizability of the findings. Future research can explore the effectiveness of the KNN classifier on larger datasets, including those from other social media platforms

Table 4 : Continued Literature Survey for Machine Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[19]	Fake News Detection System using Logistic Regression and Compare Textual Property with Support Vector Machine Algorithm	the data is unstable is a limitation of this study, which means that every prediction model would have errors and make mistakes	logistic regression and SVM	dataset used in this study consists of news articles from different sources, including both real and fake articles	study found that the frequency of occurrence of specific words and the length of the text were the most effective textual properties for fake news detection.	study only considers the use of logistic regression and SVM algorithms, and future research can explore the effectiveness of other machine learning techniques for fake news detection on social media
[15]	Fake news detection using machine learning	Lack of discussion on ethical implications	Support Vector Machine(SVM), Naïve Bayes, Passive Aggressive Classifier	a news dataset based on internet findings	the SVM model outperformed the other models	future research could focus on improving the temporal analysis of fake news
[13]	An empirical comparison of fake news detection using different machine learning algorithms	used a limited set of features for detecting fake news and did not explore more complex features such as semantic or contextual information	decision trees, random forests, naive Bayes, and Neural Networks	LIAR LIAR.. fake news dataset	The Naïve Bayes algorithm demonstrates superior performance compared to other methods due to its ability to independently compute conditional probabilities for different events or classes based on text occurrences.	Future research could investigate the role of human factors, such as social context and cultural differences, in fake news detection
[40]	Improving fake news detection using k-means and support vector machine approaches	Unwilling to cope with fake news' massive quantity of unlabeled data	combination of k-means clustering and SVM	Buzz Feed News, BS Detector, LIAR	study found that the proposed approach of using k-means clustering and SVM outperformed traditional machine learning algorithms such as decision trees and naive Bayes for fake news detection	Future research could explore the use of other clustering algorithms, such as hierarchical clustering and DBSCAN, and other classification algorithms, such as decision trees and random forests

Table 5 : Literature Survey for Deep Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[25]	Fake news detection using different Deep Learning models	The proliferation of fake news in social media and its potential harm.	LSTM CNN BERT	Getting Real about Fake News FakeNewsCorpus TI-CNN dataset	The accuracy of LSTM is 91	Lack of explanation of the model's decision-making process and the need for further evaluation of other datasets
[35]	To introduce a new pre-training method called BERT for improving language understanding tasks.	Existing models did not consider the context of words in both directions of a sentence.	BERT uses a deep bidirectional transformer to pre-train large amounts of text data.	SQuAD GLUE CoNLL	BERT achieves state-of-the-art performance on several NLP benchmarks.	BERT has a large memory footprint and can be computationally expensive, and there is still room for improvement in handling certain types of language understanding tasks.
[33]	The aim of this paper is to propose an automatic fake news detection model called "exbake" based on the BERT technique.	The proliferation of fake news poses a significant challenge to the credibility of news sources and accuracy. Thus, there is a need for effective techniques to detect fake news automatically.	exbake model based on BERT and a dataset of news articles.	Fake News Corpus LIAR Dataset	The accuracy on the Fake News Corpus dataset is 94.9% and on the Liar dataset is 95.5%.	Limited evaluation of non-English news articles and potential bias in the dataset used. Further research is needed to improve the model's performance in different languages and address bias issues.
[41]	To review and summarize the state-of-the-art fake news detection methods that use machine learning techniques	The proliferation of fake news has become a significant problem that can have adverse effects on individuals, organizations, and society	Systematic review of research papers on fake news detection that use machine learning techniques	LIAR FakeNewsNet PolitiFact GossipCop	A comprehensive overview of the current state-of-the-art fake news detection methods using machine learning techniques	Lack of standardization in terms of datasets and evaluation metrics, and the need for more research in detecting fake news in non-English languages. Standardization of datasets and evaluation metrics, and the development of effective methods for detecting fake news in non-English languages

Table 6 :Continued Literature Survey for Deep Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[34]	The aim of the paper is to improve the performance and robustness of the BERT model, which is a widely used pre-trained language model for natural language processing (NLP) tasks.	BERT model is highly sensitive to small changes in input text, which can lead to significantly different output results. Additionally, training BERT is computationally expensive and time-consuming.	The authors introduced several improvements to BERT pretraining like "sentence order prediction," including a larger corpus, dynamic masking, and training with longer sequences.	BooksCorpus GLUE SQuAD SWAG.	The optimized BERT model, called Roberta, outperformed previous BERT models on a variety of NLP benchmarks, including GLUE and SQuAD.	The authors noted that further research is needed to investigate the impact of different hyperparameters and pretraining strategies on Roberta's performance.
[42]	To propose a news text classification approach using Bidirectional Encoder Representation from Transformers (BERT) model.	Traditional news classification methods face challenges such as poor accuracy and difficulty in handling text data's complexity.	The proposed approach uses the BERT model and its output is fed into a fully connected neural network for news classification.	AG News dataset Yelp Review Full dataset	The proposed approach outperformed traditional news classification methods, achieving an accuracy of 92.4	The authors did not provide an in-depth analysis of the BERT model's performance, and the evaluation was limited to only two datasets. Future work can involve evaluating the proposed approach on a more diverse set of datasets and analyzing the BERT model's performance.
[37]	To introduce a new benchmark dataset for fake news detection	Lack of comprehensive datasets for fake news detection	Utilized the PolitiFact website to collect statements and labeled them as true, false, or half-true using expert fact-checkers	12,836 statements from PolitiFact labeled as true, false, or half-true	The proposed dataset achieved higher performance than existing datasets, demonstrating its effectiveness	The dataset only focuses on political news and may not generalize to other domains. Future work includes expanding the dataset to cover multiple domains and different types of fake news.

Table 7 : Continued Literature Survey for Deep Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[36]	The aim of the paper is to provide a comprehensive survey of the existing datasets for evaluating fake news detection systems.	Lack of standardized datasets for evaluating fake news detection methods.	Systematic literature review of studies using 35 datasets for evaluating fake news detection.	The paper surveyed 35 datasets, including real-world and synthetic datasets, such as the BuzzFeed News Dataset, the Fake News Challenge, the LIAR dataset, and the RumourEval dataset.	The datasets varied significantly in terms of size, quality, and content, making comparisons difficult. The most commonly used dataset was LIAR.	More standardized and diverse datasets are needed to improve the reliability of fake news detection models, and to better reflect the real-world challenges.
[31]	To detect and classify fake news using a hybrid BiLSTM and self-attention model.	Traditional methods of detecting fake news are not effective, especially with the emergence of deep fake technology.	(BiLSTM) network and a self-attention mechanism.	Fake News Detection Dataset Kaggle Fake News dataset.	The accuracy on the Fake News Detection Dataset is 95% and on the Kaggle Fake News dataset is 97%.	The paper lacked a comparative analysis of the proposed model against other cutting-edge models in the field of fake news detection. Additionally, the study was confined to the English language. To enhance the model's efficacy, future research could investigate the incorporation of additional features, such as user profiles and network structure, into the analysis.
[30]	To develop a quantitative approach for classifying fake news in low-resource languages.	Limited research has been done on fake news classification in low-resource languages, which are particularly vulnerable to fake news due to a lack of resources for fact-checking.	LSTM Bi-LSTM Naive Bayes Passive Aggressive Classifier Random Forest	BanFakeNews (A Dataset for Detecting Fake News in Bangla)	The deep learning method BiLSTM shows 55.92% accuracy while Random Forest, in contrast, has outperformed all the other methods with an accuracy of 62.37%.	The authors acknowledge that their approach may not generalize well to other low-resource languages and that future work should explore the use of more advanced natural language processing techniques. They also suggest the need for larger and more diverse datasets to further improve the accuracy of fake news classification in low-resource languages.

Table 8 : Continued Literature Survey for Deep Learning Techniques.

Authors(year)	Aim	Issues	Methods	Datasets	Findings	Shortcomings/future work
[38]	To analyze machine learning enabled fake news detection techniques for diversified datasets.	The proliferation of false information on social media and other online platforms is an increasing issue, and identifying fake news poses significant difficulties due to its constantly evolving nature and the sheer volume of data that needs to be examined.	SVM Logistic regression Decision tree AdaBoost Naive Bayes k-NN CNN LSTM Bi-LSTM GAN	Buzzfeed Dataset Kaggle Dataset Corpus	Research in this field focuses on the theoretical foundations and related work, providing a comprehensive comparative analysis of existing literature contributions to the topic.	A potential limitation of the analysis is its focus on textual data. In the future, expanding the scope to include image data could lead to more comprehensive and diverse analysis results, considering a wider range of datasets.

7 Model–Oriented Performance Evaluation

7.1 Evaluation Parameters

Evaluation parameters refer to the criteria or metrics used to assess the performance or effectiveness of a system, process, or project. The selection of assessment criteria is based on the evaluation's goals, objectives, and the characteristics of the system under examination.
There exits multiple parameters used for evaluation purposes that can be used for the assessment of the performance of a fake news detection system. Some of the commonly used evaluation parameters for fake news detection are:

1. **Precision:** the proportion of true positives (correctly identified fake news) among all the news articles classified as fake news.
2. **Recall:** the proportion of true positives (correctly identified fake news) among all the actual fake news articles.
3. **F1 score:** A average based on weights of recall and accuracy that offers a single score for assessing the system's performance.
4. **Accuracy:** the proportion of correctly classified articles (both real and fake news).
5. **Confusion matrix:** a summary table detailing the system's performance in terms of true positives, true negatives, false positives, and false negatives.
6. **False positive rate:** the proportion of real news articles that are incorrectly classified as fake news.
7. **False negative rate:** the proportion of fake news articles that are incorrectly classified as real news.

We will use the following Notations:

1. True positive: TP
2. True negative: TN
3. Fale positive : FP
4. False negative: FN

In our project we have used the following evaluation criteria:

1. ***Accuracy:*** Accuracy is a commonly used evaluation parameter that measures the degree to which a system or process produces correct results or outcomes. In the context of evaluating a system for fake news detection, accuracy refers to the proportion of news articles that are correctly classified as real or fake news.
   To calculate the accuracy, the system is first trained on a labeled dataset of news articles where each article is labeled as real or fake news. The trained system is then tested on a separate set of unlabeled news articles to evaluate its performance. The accuracy is determined by dividing the percentage of properly categorised articles by the total number of articles in the test set. The output of the system is compared to the actual label of each news article.
   To calculate the accuracy, we can use the following formula:

$$Accuracy = \text{(Number of correctly classified articles) / Total number of articles} * 100 \tag{26}$$

2. ***Precision:*** Among all the news pieces that the system categorised as fake news, precision is an assessment metric that quantifies the percentage of accurately classified false news articles. In other words, precision measures how many of the articles that the system classified as fake news are actually fake news.
   To calculate precision, we need to use the following formula:

$$Precision = (TP/(TP + FP)) * 100 \tag{27}$$

   where False Positives are the number of genuine news pieces that the system misidentified as fake news, whereas True Positives are the number of fraudulent news articles that the system successfully recognised.
   example: Suppose we have a test set of 100 news articles, of which 20 are fake news articles. The fake news detection system correctly identifies 12 of the 20 fake news articles and incorrectly identifies 6 real news articles as fake news. Then, the number of True Positives is 12, and the number of False Positives is 6. We can calculate the precision as follows:

$$Precision = (12/(12 + 6)) * 100 \tag{28}$$

   articles the system classified as fake news, 66.67 percent of them were actually fake news. A higher precision score signifies that the system is comparitively good at correct identification of fake news articles. However, it is also important to consider other evaluation parameters like recall and F1-score to get a comprehensive understanding of the system's performance.
3. ***Recall:***Recall is an evaluation parameter that measures the proportion of true positives (TP) among all the actual positive samples ($TP + FN$). It is commonly used in classification problems, where we want to measure how well a model can identify all positive samples in a dataset. To calculate recall, we can use the following formula:

$$Recall = TP/(TP + FN) \tag{29}$$

   where TP is the number of true positive samples, and FN is the number of false negative samples.
   A higher recall score denotes that the system is better at correct identification of fake news articles. However, it is also important to consider other evaluation parameters like precision and F1-score to get a comprehensive understanding of the system's performance.
4. ***F1-Score:***F1 score is an evaluation parameter used in classification problems that combines both precsion and re-call. It provides a single measure of a model's performance, taking into account both false positives and false negatives. To calculate the F1 score, we use the following formula:

$$F1Score = 2 * (Precision * Recall)/(Precision + Recall) \tag{30}$$

   where Precision is the proportion of true positives

$$(TP)$$

among all positive predictions

$$(TP + FP)$$

, and Recall is the proportion of true positives

$$(TP)$$

among all actual positive samples

$$(TP + FN)$$

.
A higher F1 score denotes a model that performs better overall in terms of precision and recall. It is often used as a single measure of performance in classification problems, especially when the classes are imbalanced.

7.2 Machine learning-based models

Table 9 Results of Machine Learning Techniques for ISOT Fake News Dataset

Fake News Datasets	Machine Learning Models	NLP Techniques	Evaluation Metrics For Testing Dataset			
			Accuracy	Precision	Recall	F-1 Score
ISOT Fake News Dataset	Multinomial NB	BoW - (Bag of Words)	51.28%	54.34%	67.44%	60.18%
	Logistic Regression		51.11%	32.07%	18.88%	35.67%
	Decision Tree		49.73%	17.39%	13.33%	24.76%
	SVM		51.70%	30.03%	16.60%	13.04%
	KNN		51.91%	35.71%	15.05%	10.94%
	Passive Aggresive Classifier		51.11%	40.40%	44.44%	78.08%
	Random Forest		52.18%	50.98%	28.88%	54.67%
ISOT Fake News Dataset	Multinomial NB	TF-IDF	65.85%	67.74%	55.84%	61.22%
	Logistic Regression		55.45%	75.89%	11.30%	19.67%
	Decision Tree		51.66%	48.10%	18.43%	35.51%
	SVM		55.94%	74.45%	13.29%	22.56%
	KNN		53.16%	51.78%	42.94%	46.93%
	Passive Aggresive Classifier		56.65%	71.60%	16.88%	27.32%
	Random Forest		53.48%	82.60%	46.09%	18.73%
ISOT Fake News Dataset	Multinomial NB	Hashing Vectorizer	85.26%	83.34%	80.17%	81.20%
	Logistic Regression		96.55%	96.23%	96.75%	96.44%
	Decision Tree		98.07%	98.01%	98.01%	98.04%
	SVM		97.82%	97.53%	97.96%	97.74%
	KNN		84.16%	76.66%	96.60%	85.48%
	Passive Aggresive Classifier		97.84%	97.67%	97.86%	97.77%
	Random Forest		98.70%	98.92%	98.35%	98.63%

Table 10 Results of Machine Learning Techniques for LIAR Fake News Dataset

Fake News Datasets	Machine Learning Models	NLP Techniques	Evaluation Metrics For Testing Dataset			
			Accuracy	Precision	Recall	F-1 Score
LIAR Fake News Datasets	Multinomial NB	BoW - (Bag of Words)	60.76%	57.47%	88.47%	69.68%
	Logistic Regression		61.73%	72.10%	40.61%	51.96%
	Decision Tree		56.03%	59.20%	44.11%	50.55%
	SVM		60.24%	83.04%	27.62%	41.45%
	KNN		50.18%	76.13%	32.58%	16.25%
	Passive Aggresive Classifier		63.02%	71.69%	45.33%	55.54%
	Random Forest		70.07%	74.60%	59.58%	66.25%
LIAR Fake News Datasets	Multinomial NB	TF-IDF	61.14%	59.77%	72.56%	65.55%
	Logistic Regression		51.37%	69.26%	18.21%	14.70%
	Decision Tree		51.01%	55.21%	20.33%	29.71%
	SVM		51.42%	71.23%	18.23%	14.11%
	KNN		52.26%	57.30%	24.80%	34.62%
	Passive Aggresive Classifier		54.12%	61.76%	26.16%	36.76%
	Random Forest		50.28%	69.53%	14.32%	18.15%
LIAR Fake News Datasets	Multinomial NB	Hashing Vectorizer	89.24%	87.11%	84.46%	83.69%
	Logistic Regression		92.76%	94.05%	91.58%	92.80%
	Decision Tree		90.58%	90.60%	90.95%	90.77%
	SVM		94.59%	95.67%	93.62%	94.64%
	KNN		85.55%	81.46%	92.75%	86.74%
	Passive Aggresive Classifier		92.31%	92.75%	92.12%	92.43%
	Random Forest		94.75%	94.63%	95.03%	94.83%

Table 11 Results of Machine Learning Techniques for Kaggle Fake News Dataset

Fake News Datasets	Machine Learning Models	NLP Techniques	Evaluation Metrics For Testing Dataset			
			Accuracy	Precision	Recall	F-1 Score
Kaggle Fake News Dataset	Multinomial NB	BoW - (Bag of Words)	90.19%	87.42%	90.66%	89.01%
	Logistic Regression		93.42%	89.14%	96.78%	92.80%
	Decision Tree		91.61%	89.54%	91.57%	90.54%
	SVM		92.93%	87.26%	98.18%	92.40%
	KNN		79.85%	68.58%	99.73%	81.27%
	Passive Aggresive Classifier		92.12%	90.30%	91.91%	91.10%
	Random Forest		93.32%	89.11%	96.25%	92.55%
Kaggle Fake News Dataset	Multinomial NB	TF-IDF	88.10%	91.44%	80.38%	85.55%
	Logistic Regression		92.24%	86.47%	97.58%	91.69%
	Decision Tree		91.16%	89.11%	90.96%	90.03%
	SVM		92.87%	87.05%	98.37%	92.23%
	KNN		52.14%	47.79%	99.13%	64.49%
	Passive Aggresive Classifier		91.41%	90.64%	89.68%	90.15%
	Random Forest		93.27%	88.91%	96.67%	92.63%
Kaggle Fake News Dataset	Multinomial NB	Hashing Vectorizer	88.09%	91.24%	78.04%	84.10%
	Logistic Regression		92.46%	85.13%	97.58%	90.93%
	Decision Tree		90.40%	88.23%	90.13%	89.17%
	SVM		92.81%	86.34%	98.28%	92.03%
	KNN		72.23%	61.41%	99.77%	75.25%
	Passive Aggresive Classifier		90.10%	88.07%	89.56%	88.81%
	Random Forest		93.04%	87.89%	97.43%	92.41%

Table 12 Results of Machine Learning Techniques for WELFAKE Fake News Dataset

Fake News Dataset	Machine Learning Models	NLP Techniques	Evaluation Metrics For Testing Dataset			
			Accuracy	Precision	Recall	F-1 Score
WELFAKE Fake News Dataset	Multinomial NB	BoW - (Bag of Words)	86.85%	85.37%	89.56%	87.41%
	Logistic Regression		89.57%	89.15%	89.34%	90.41%
	Decision Tree		85.79%	84.99%	87.61%	86.29%
	SVM		87.62%	83.24%	87.88%	89.95%
	KNN		66.65%	61.29%	93.89%	74.16%
	Passive Aggresive Classifier		85.70%	87.65%	84.70%	85.65%
	Random Forest		90.14%	88.41%	91.22%	90.79%
WELFAKE Fake News Dataset	Multinomial NB	TF-IDF	87.88%	85.96%	89.44%	87.33%
	Logistic Regression		89.60%	88.64%	91.20%	89.84%
	Decision Tree		87.23%	87.10%	88.04%	87.57%
	SVM		88.25%	90.24%	91.45%	88.32%
	KNN		63.08%	58.56%	90.24%	73.39%
	Passive Aggresive Classifier		87.62%	91.18%	85.54%	87.27%
	Random Forest		89.50%	88.50%	91.22%	89.79%
WELFAKE Fake News Dataset	Multinomial NB	Hashing Vectorizer	86.68%	86.05%	87.25%	86.14%
	Logistic Regression		87.29%	86.82%	88.51%	87.66%
	Decision Tree		86.02%	85.86%	86.95%	85.40%
	SVM		89.22%	87.66%	88.20%	89.14%
	KNN		63.53%	58.40%	98.95%	74.45%
	Passive Aggresive Classifier		85.17%	89.82%	82.12%	85.50%
	Random Forest		90.12%	89.90%	89.83%	89.81%

7.3 Deep learning-based models

Models	Datasets							
	ISOT				Kaggle Fake News			
	Acc	Prec	R	F1	Acc	Prec	R	F1
CNN	98.74	99	99	99	94.87	95	95	95
LSTM	98.77	99	99	99	94.67	95	95	95
Bi-LSTM	98.84	99	99	99	94.95	95	95	95
BERT	99.76	100	100	100	98.43	98	98	98
RO-BERTA	99.84	100	100	100	98.28	98	98	98

Models	Datasets							
	WELFake				Liar Liar			
	Acc	Prec	R	F1	Acc	Prec	R	F1
CNN	94.65	95	95	95	56.51	51	50	50
LSTM	95.11	95	95	95	55.48	50	50	50
Bi-LSTM	95.03	95	95	95	53.67	49	49	49
BERT	99.04	99	99	99	66.89	62	63	62
RO-BERTA	99.64	100	100	100	67.95	64	65	64

Note: Acc: Accuracy; Prec: Precision; R-Recall: F1-F1score.

8 Discussions & analysis

The paper provides a comprehensive review of the fake news detection model, incorporating contributions from the previous year. It presents a survey that offers in-depth insights into fake news, including its various types. Furthermore, the paper discusses a range of machine learning and deep learning techniques employed in the field of fake news detection. It specifically examines different machine learning and deep learning approaches, outlines four datasets (ISOT dataset, the Liar fake news dataset, the Kaggle fake news dataset, WELFAKE dataset), explores various natural language processing techniques, and provides a detailed explanation of the utilized machine learning and deep learning techniques, along with their distinctive features and challenges.

Utilizing a variety of traditional machine learning algorithms, including SVM, logistic regression, KNN, decision tree, multinomial naive Bayes, and passive-aggressive classifier, in combination with NLP techniques such as bag of words, tf-idf, and hashing vectorizer, can offer numerous advantages for the detection of fake news.

Traditional machine learning algorithms offer the advantage of being more interpretable compared to deep learning models. This interpretability facilitates a better

understanding of the underlying processes through which these algorithms make predictions. Such transparency and accountability are particularly crucial in situations where clear explanations are needed.

One advantage of traditional machine learning algorithms is their computational efficiency and ability to operate with fewer resources compared to deep learning models. This characteristic becomes particularly significant when working with limited computing resources or handling small datasets. Of all the Machine Learning Techniques, using 3 NLP techniques(bag of words, tf-idf, hashing vectorizer) -

The Random Forest performed the best with an average accuracy of 68.6%, 76.66 precision, 57.33 Recall, and 56.66 F1-score in the ISOT dataset. For the Liar fake news dataset, Random Forest performed the best with an average accuracy of 71.6%, 79.55 precision, 66.33 Recall, and 62.43 F1-score. For the Liar fake news dataset, the Random Forest performed the best with an average accuracy of 93.33%, 88.35 precision, 97.72 Recall, and 92.43 F1-score. For the WELFake fake news dataset, the Random Forest performed the best with an average accuracy of 89.89%, 88.63 precision, 91.12 Recall, and 89.34 F1-score.

Overall, Machine learning is a powerful tool for detecting fake news, but it also has some limitations and challenges. Here are some of the main problems in the field of fake news detection using machine learning:

1. **Limited data availability:** Machine learning models rely on ample quantities of reliable training data to learn patterns and generate precise predictions. However, when it comes to identifying fake news, there is a scarcity of labeled data, which poses a challenge in training machine learning models effectively.
2. **Difficulty in defining and labeling fake news:** There is no universally agreed-upon definition of fake news, and categorizing news articles as "fake" or "real" can be subjective and open to interpretation. This subjectivity in labeling training data can introduce inconsistencies and potentially impact the accuracy of machine-learning models.
3. **Adversarial attacks:** Adversarial attacks can be used to deceive machine learning models by introducing small changes to the input data that can cause the model to misclassify the input. For example, an attacker can make slight changes to a news article to make it appear more authentic and bypass the machine learning model.
4. **Limited generalization ability:** Machine learning models can have limited generalization ability, meaning they may perform poorly on news articles that are outside of the training data distribution. This can be a problem in the case of fake news detection as new and novel types of fake news can emerge over time.
5. **Lack of transparency:** Some machine learning models can be difficult to interpret, which can make it challenging to understand how the model is making its predictions. This can be a problem in the case of fake news detection, as it can be difficult to understand why the model is classifying a news article as fake or real.

To cope with this problem we used various Deep Learning methods like CNN, RNN, LSTM, Bi-LSTM, BERT, and Roberta. In the field of fake news detection, deep

learning has several advantages over traditional machine learning methods, which I will explain in detail below:

1. **Ability to learn complex patterns:** Deep learning models possess the ability to comprehend intricate patterns within data, which is a challenging task for conventional machine learning techniques. This quality is especially valuable in the context of fake news detection, as fake news articles frequently exhibit subtle patterns and nuances that prove elusive for identification.
2. **Feature extraction:** Deep learning models have the ability to automatically extract significant features from unprocessed data. This characteristic is especially valuable in the context of fake news detection since the pertinent features might not be readily apparent.
3. **Improved accuracy:**Deep learning models have demonstrated superior accuracy compared to conventional machine learning approaches across various tasks, including the detection of fake news. This is attributed to their capacity to acquire intricate patterns and extract pertinent features.
4. **Ability to handle large datasets:**Deep learning models are capable of processing extensive datasets containing numerous features, making them highly valuable for the identification of fake news. This is crucial because fake news detection often requires considering a wide range of relevant features.
5. **Transfer learning:**Deep learning models have the ability to undergo pre-training using extensive datasets, followed by fine-tuning on smaller datasets for specialized tasks like identifying fake news. This approach enhances their accuracy and minimizes the data volume needed for training.

Overall, ***"Deep learning offers numerous advantages over traditional machine learning techniques in the domain of fake news detection. Its capacity to comprehend intricate patterns, extract pertinent features, handle substantial datasets, and apply transfer learning renders it an influential tool for identifying fake news."***

8.1 Model Structure and Parameters Used in Study for the Deep Learning Models

8.1.1 CNN

1. First is the embedding layer of the model. The embedding layer is used to convert the input sequences (integer-encoded words) into dense vectors of fixed size. The 5000 parameter specifies the vocabulary size, 128 is the dimensionality of the embedding space, and 100 is the length of input sequences.
2. Then we add a 1D convolutional layer to the model. The convolutional layer is used to apply a set of filters to the input and extract local features. The 64 parameter represents the number of filters, and 5 is the size of each filter. The activation function used is ReLU (Rectified Linear Unit).

3. Then we add a global max pooling layer. The global max pooling operation reduces the dimensionality of the input by taking the maximum value over each feature map. It helps to capture the most important features from the previous layer.
4. Then we add a dense layer to the model. The dense layer is a fully connected layer where each neuron is connected to every neuron in the previous layer. The 2 parameter specifies the number of neurons in the layer, and the activation function used is sigmoid. Since the problem seems to be a binary classification task, the output layer has 2 neurons, each representing the probability of belonging to one of the classes.
5. Then we compile the model, configuring the learning process. The optimizer used is Adam, a popular optimization algorithm. The loss function is set to binary cross-entropy, which is commonly used for binary classification problems. The metric specified is accuracy, which will be used to evaluate the model's performance.

8.1.2 LSTM

1. First is the embedding layer of the model. The embedding layer is used to convert the input sequences (integer-encoded words) into dense vectors of fixed size. The 5000 parameter specifies the vocabulary size, 128 is the dimensionality of the embedding space, and 100 is the length of input sequences.
2. Then we add an LSTM (Long Short-Term Memory) layer to the model. LSTM is a type of recurrent neural network (RNN) that can capture long-term dependencies in sequential data. The 64 parameter represents the number of LSTM units (or cells) in the layer. The dropout parameter specifies the dropout rate, which helps prevent overfitting by randomly disabling a portion of the LSTM units during training. We add a recurrentdropout parameter that specifies the dropout rate for the recurrent connections within the LSTM units.
3. Then we add a dense output layer to the model, similar to the previous example. It has 2 neurons, representing the probability of belonging to each class. The activation function used is sigmoid.
4. Then we compile the model, configuring the learning process. The optimizer used is Adam, a popular optimization algorithm. The loss function is set to binary cross-entropy, which is commonly used for binary classification problems. The metric specified is accuracy, which will be used to evaluate the model's performance.

8.1.3 BI-LSTM

1. First is the embedding layer of the model. The embedding layer is used to convert the input sequences (integer-encoded words) into dense vectors of fixed size. The 5000 parameter specifies the vocabulary size, 128 is the dimensionality of the embedding space, and 100 is the length of input sequences.
2. Then we add a Bidirectional LSTM layer to the model. A Bidirectional LSTM processes the input sequence in both directions, incorporating information from past and future timesteps. The 64 parameter represents the number of LSTM units (or cells) in each direction. The dropout parameter specifies the dropout rate, which helps prevent overfitting by randomly disabling a portion of the LSTM units during

training. Then we add a recurrentdropout parameter that specifies the dropout rate for the recurrent connections within the LSTM units.
3. Then we add a dense output layer to the model, similar to the previous examples. It has 2 neurons, representing the probability of belonging to each class. The activation function used is sigmoid.
4. Then we compile the model, configuring the learning process. The optimizer used is Adam, a popular optimization algorithm. The loss function is set to binary cross-entropy, which is commonly used for binary classification problems. The metric specified is accuracy, which will be used to evaluate the model's performance.

8.1.4 BERT

1. The BERT model for sequence classification consists of a BERT encoder with 12 layers, incorporating self-attention and feed-forward neural network layers. It receives input embeddings for words, positions, and token types, while utilizing layer normalization and dropout for regularization.
2. The BertEmbeddings module manages the word embeddings, position embeddings, and token-type embeddings. The word-embeddings embedding layer has a vocabulary size of 30522, producing embeddings of size 768. Both the position-embeddings and token-type-embeddings layers also generate embeddings of size 768. Layer normalization and dropout are applied to the embeddings.
3. The BertEncoder contains a list of 12 BertLayer modules, each representing a single encoder layer. Each BertLayer consists of a BertSelfAttention module for self-attention computations, a BertIntermediate module with a dense layer and GELU activation function, and a BertOutput module with a dense layer for the final output. The self-attention module applies linear transformations to the input and incorporates dropout. The intermediate module utilizes a dense layer with 3072 output features and the GELU activation function. The output module employs a dense layer with 768 output features, incorporating layer normalization and dropout.
4. The BertPooler module comprises a dense layer with 768 input features and 768 output features, followed by a hyperbolic tangent activation function.
5. The BertForSequenceClassification model includes a dropout layer with a dropout probability of 0.1. The final classification is performed by a linear layer with 768 input features and 2 output features for binary classification.

8.1.5 RO-BERT

1. The RobertaForSequenceClassification model is based on the RoBERTa architecture, designed for sequence classification tasks. It utilizes a RobertaEncoder, which consists of 12 RobertaLayers. Each layer includes a self-attention mechanism and feed-forward neural network layers.
2. The input to the model includes word embeddings, position embeddings, and token type embeddings. The word-embeddings layer has a vocabulary size of 50265, generating embeddings of size 768. The position-embeddings and token-type-embeddings layers also produce embeddings of size 768. Layer normalization and dropout are applied to the embeddings for regularization.

3. The RobertaEncoder contains 12 RobertaLayers, each comprising a RobertaSelfAttention module for self-attention computations, a RobertaIntermediate module with a dense layer and GELU activation function, and a RobertaOutput module with a dense layer for the final output. The self-attention module performs linear transformations on the input, applying dropout for regularization. The intermediate module includes a dense layer with 3072 output features and the GELU activation function. The output module consists of a dense layer with 768 output features, applying layer normalization and dropout.
4. The RobertaForSequenceClassification model also includes a classifier called RobertaClassificationHead. This classifier consists of a dense layer with 768 input features and 768 output features, followed by dropout. The final classification is performed by a linear layer with 768 input features and 2 output features, suitable for binary classification tasks.
5. Overall, the model consists of 12 encoder layers, embedding layers, attention mechanisms, feed-forward layers, layer normalization, dropout, and a linear classifier. It has a total of 125 million trainable parameters, making it a powerful model for sequence classification.

8.2 Result analysis for the Datasets

First Dataset is **ISOT dataset**. we can observe that the performance of the models is as follows: -
The CNN model has an accuracy of 98.74%, precision, recall, and an F1-score of 99. The LSTM model has an accuracy of 98.77%, precision, recall, and an F1-score of 99. The Bi-LSTM model has an accuracy of 98.84%, recall and F1-score of 99%, and precision of 98. The BERT model has an accuracy of 99.76%, recall, F1-score, and precision of 100. The RO-BERTA model outperforms all the other models with an accuracy of 99.84%, precision, recall, and F1-score of 100.

Overall, the ***"Bi-LSTM, BERT, and RO-BERTA models perform better than the CNN and LSTM models on the ISOT dataset. The RO-BERTA model outperforms all other models in all the metrics, indicating that it is the best model for this dataset."***

Next is **Kaggle Fake News Dataset**. we can observe that the performance of the models is as follows: -
The CNN model has an accuracy of 94.87%, precision, recall, and an F1-score of 95. The LSTM model has an accuracy of 94.67%, precision, recall, and an F1-score of 95. The Bi-LSTM model has an accuracy of 94.95%, recall, F1-score, and precision of 95. The BERT model outperforms all the other models with an accuracy of 98.43%, recall, F1-score, and precision of 98. The RO-BERTA model outperforms all the other models with an accuracy of 98.28%, precision, recall, and F1-score of 98.

Overall, the ***"The BERTA model outperforms all other models in all the metrics, indicating that it is the best model for Kaggle Fake News Dataset."***

Next is **WELFake**. we can observe that the performance of the models is as follows: -
The CNN model has an accuracy of 94.65%, precision, recall, and an F1-score of 95. The LSTM model has an accuracy of 95.11%, precision, recall, and an F1-score of 95. The Bi-LSTM model has an accuracy of 95.03%, recall and F1-score of 99%, and precision of 95. The BERT model has an accuracy of 99.04%, recall, F1-score, and precision of 99. The RO-BERTA model outperforms all the other models with an accuracy of 99.64%, precision, recall, and F1-score of 100.

Overall, the ***"The RO-BERTA model outperforms all other models in all the metrics, indicating that it is the best model for WELFake Fake News Dataset."***

Next is **LIAR FAKE NEWS DATASET**. we can observe that the performance of the models is as follows: -
The CNN model has an accuracy of 56.51%, a precision 51, a recall 50, and an F1-score of 51. The LSTM model has an accuracy of 55.48%, precision , recall, and an F1-score of 50. The Bi-LSTM model has an accuracy of 53.67%, recall ,F1-score, and precision of 49. The BERT model has an accuracy of 66.89%, recall 62, F1-score 63, and precision of 62. The RO-BERTA model outperforms all the other models with an accuracy of 67.95%, precision 64, recall 65, and F1-score of 64.

Overall, the ***"The RO-BERTA model outperforms all other models in all the metrics, indicating that it is the best model for LIAR FAKE NEWS DATASET."***

So overall ***"BERT and RoBERTa have outperformed other ML and DL techniques due to several key factors. Firstly, their pre-training on extensive datasets enables them to capture a wide range of linguistic patterns and nuances. Secondly, their capability to handle input sequences of varying lengths is advantageous in processing textual data. Additionally, these models excel at extracting more significant features from the input text, thus enhancing their ability to detect fake news. Lastly, their utilization of contextual information and word relationships makes them a superior choice compared to conventional deep learning methods when it comes to Fake News Detection."***

9 Current research challenges in Fake news detection methods

Fake news classification using machine learning (ML) and deep learning (DL) techniques is a complex and demanding task. While these methods have shown promise in accurately classifying fake news, several challenges need to be overcome to enhance the accuracy and reliability of the models. One significant challenge in fake news classification is the limited availability of large annotated datasets. Annotated datasets

play a crucial role in training and evaluating ML and DL models. However, creating high-quality annotated datasets requires substantial time and effort, which can impede the development of effective models. To address this challenge, researchers are exploring techniques to augment existing datasets and employ unsupervised learning methods to reduce dependence on annotated data. Another obstacle is the dynamic and ever-changing nature of fake news. New forms of fake news emerge regularly, posing difficulties in developing ML and DL models that can effectively classify all types of fake news.

To overcome this challenge, researchers are actively investigating methods to enhance the adaptability of models. They are exploring techniques such as active learning and domain adaptation to improve model performance. Deep learning models, such as LSTM, CNN, and BERT, are known for their complexity, which makes it difficult to interpret their decision-making process. This lack of interpretability hinders efforts to improve accuracy. To tackle this issue, researchers are working on developing new explainability methods and incorporating interpretability into the model architecture. Adversarial attacks pose a significant threat to ML and DL models as they can modify fake news articles to be misclassified, thus undermining the models' accuracy. To address this challenge, researchers are focusing on improving the robustness of models through adversarial training techniques and developing new defense mechanisms. The creation of fake news is not limited to any particular language or cultural context, making it challenging to develop ML and DL models that can accurately classify fake news across various languages and cultural contexts. Researchers are actively exploring ways to enhance the generalizability of models by developing new cross-lingual and cross-cultural techniques. One prominent concern is the potential bias in ML and DL models caused by biased training data. Biased models can result in inaccurate classification of fake news, perpetuating the spread of misinformation. To combat this issue, researchers are working on improving the fairness and inclusiveness of models through the development of new techniques for bias detection and mitigation.

ML and DL models, when trained on a specific dataset, often lack generalizability and struggle to perform effectively in different datasets or real-world scenarios. Consequently, their ability to classify fake news in novel contexts becomes limited. To tackle this challenge, researchers are actively investigating methods to enhance the transferability of these models. They are developing novel transfer learning techniques and integrating external knowledge sources into the models, aiming to improve their performance across diverse contexts and datasets.

"In summary, the classification of fake news using machine learning (ML) and deep learning (DL) techniques presents a complex and demanding task, necessitating the creation of novel methods and approaches to tackle the aforementioned challenges. Although there remains ample room for further research and development, the ongoing progress in ML and DL techniques offers hope for enhancing the precision and dependability of fake news classification models."

References

[1] A. Gelfert, Fake news: A definition. Informal Logic, 2018.

[2] P. B. J.-c. G. J. O. M. R. Melanie Freeze, Mary Baumgartner, J. Szafran., Fake claims of fake news: Political misinformation, warnings, and the tainted truth effect. (02 2020).

[3] Z.-H. Zhou., A brief introduction to weakly super- vised learning. national science review (2017) 44–53.

[4] Z. Zhao, J. Zhao, Y. Sano, O. Levy, H. Takayasu, M. Takayasu, D. Li, J. Wu, S. Havlin, Fake news propagates differently from real news even at early stages of spreading, EPJ data science 9 (1) (2020) 7.

[5] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, L. De Alfaro, Some like it hoax: Automated fake news detection in social networks, arXiv preprint arXiv:1704.07506 (2017).

[6] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, Journal of economic perspectives 31 (2) (2017) 211–236.

[7] E. C. Tandoc Jr, Z. W. Lim, R. Ling, Defining "fake news" a typology of scholarly definitions, Digital journalism 6 (2) (2018) 137–153.

[8] B. Horne, S. Adali, This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, in: Proceedings of the international AAAI conference on web and social media, Vol. 11, 2017, pp. 759–766.

[9] Y. Tsfati, H. G. Boomgaarden, J. Strömbäck, R. Vliegenthart, A. Damstra, E. Lindgren, Causes and consequences of mainstream media dissemination of fake news: literature review and synthesis, Annals of the International Communication Association 44 (2) (2020) 157–173.

[10] J. C. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, Supervised learning for fake news detection, IEEE Intelligent Systems 34 (2) (2019) 76–81.

[11] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: A data mining perspective, ACM SIGKDD explorations newsletter 19 (1) (2017) 22–36.

[12] V. Jakkula, Tutorial on support vector machine (svm), School of EECS, Washington State University 37 (2.5) (2006) 3.

[13] A. Albahr, M. Albahar, An empirical comparison of fake news detection using different machine learning algorithms, International Journal of Advanced Computer Science and Applications 11 (9) (2020).

[14] Y.-Y. Song, L. Ying, Decision tree methods: applications for classification and prediction, Shanghai archives of psychiatry 27 (2) (2015) 130.

[15] J. Shaikh, R. Patil, Fake news detection using machine learning, in: 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), IEEE, 2020, pp. 1–5.

[16] A. Kesarwani, S. S. Chauhan, A. R. Nair, Fake news detection on social media using k-nearest neighbor classifier, in: 2020 international conference on advances in computing and communication engineering (ICACCE), IEEE, 2020, pp. 1–4.

[17] Z. Zhang, Introduction to machine learning: k-nearest neighbors, Annals of translational medicine 4 (11) (2016).

[18] I. Reis, D. Baron, S. Shahaf, Probabilistic random forest: A machine learning algorithm for noisy data sets, The Astronomical Journal 157 (1) (2018) 16.

[19] N. L. S. R. Krishna, M. Adimoolam, Fake news detection system using logistic regression and compare textual property with support vector machine algorithm, in: 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), IEEE, 2022, pp. 48–53.

[20] M. P. LaValley, Logistic regression, Circulation 117 (18) (2008) 2395–2399.

[21] S. Ahmed, K. Hinkelmann, F. Corradini, Development of fake news model using machine learning through natural language processing, arXiv preprint arXiv:2201.07489 (2022).

[22] G. Grefenstette, Tokenization, Syntactic Wordclass Tagging (1999) 117–133.

[23] K. Poddar, K. Umadevi, et al., Comparison of various machine learning models for accurate detection of fake news, in: 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), Vol. 1, IEEE, 2019, pp. 1–5.

[24] R. Zhao, K. Mao, Fuzzy bag-of-words model for document representation, IEEE transactions on fuzzy systems 26 (2) (2017) 794–804.

[25] Á. Ibrain Rodríguez, L. Lloret Iglesias, Fake news detection using deep learning, arXiv e-prints (2019) arXiv–1910.

[26] T. Kattenborn, J. Leitloff, F. Schiefer, S. Hinz, Review on convolutional neural networks (cnn) in vegetation remote sensing, ISPRS journal of photogrammetry and remote sensing 173 (2021) 24–49.

[27] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh, B. Chapman, T. Amrhein, D. Mong, D. L. Rubin, et al., Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification, Artificial intelligence in medicine

97 (2019) 79–88.

[28] P. J. Werbos, Backpropagation through time: what it does and how to do it, Proceedings of the IEEE 78 (10) (1990) 1550–1560.

[29] R. C. Staudemeyer, E. R. Morris, Understanding lstm–a tutorial into long short-term memory recurrent neural networks, arXiv preprint arXiv:1909.09586 (2019).

[30] R. Jain, D. K. Jain, N. Sharma, Fake news classification: A quantitative research description, Transactions on Asian and Low-Resource Language Information Processing 21 (1) (2021) 1–17.

[31] A. Mohapatra, N. Thota, P. Prakasam, Fake news detection and classification using hybrid bilstm and self-attention model, Multimedia Tools and Applications 81 (13) (2022) 18503–18519.

[32] M. Koroteev, Bert: a review of applications in natural language processing and understanding, arXiv preprint arXiv:2103.11943 (2021).

[33] H. Jwa, D. Oh, K. Park, J. M. Kang, H. Lim, exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert), Applied Sciences 9 (19) (2019) 4062.

[34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: a robustly optimized bert pretraining approach (2019), arXiv preprint arXiv:1907.11692 364 (1907).

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

[36] A. D'Ulizia, M. C. Caschera, F. Ferri, P. Grifoni, Fake news detection: a survey of evaluation datasets, PeerJ Computer Science 7 (2021) e518.

[37] W. Y. Wang, " liar, liar pants on fire": A new benchmark dataset for fake news detection, arXiv preprint arXiv:1705.00648 (2017).

[38] S. Mishra, P. Shukla, R. Agarwal, Analyzing machine learning enabled fake news detection techniques for diversified datasets, Wireless Communications and Mobile Computing 2022 (2022) 1–18.

[39] P. K. Verma, P. Agrawal, I. Amorim, R. Prodan, Welfake: Word embedding over linguistic features for fake news detection, IEEE Transactions on Computational Social Systems 8 (4) (2021) 881–893.

[40] K. M. Yazdi, A. M. Yazdi, S. Khodayi, J. Hou, W. Zhou, S. Saedy, Improving fake news detection using k-means and support vector machine approaches, International Journal of Electronics and Communication Engineering 14 (2) (2020)

38–42.

[41] M. Choudhary, R. Jain, A. Gupta, R. Kumar, N. Prakash, A review of fake news detection methods using machine learning, in: 2021 2nd International Conference for Emerging Technology (INCET), IEEE, 2021.

[42] L. Deping, W. Hongjuan, L. Mengyang, L. Pei, News text classification based on bidirectional encoder representation from transformers, in: 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), IEEE, 2021, pp. 137–140.