People's Democratic Republic of Algeria
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



Yahia Fares University of Medea
Faculty of Sciences
Department of Mathematics and Computer Science



Graduation Project

Submitted for Master Degree in Computer Science

Specialty: System Engineering and Web Technology

Deep Learning and Transfer Learning Models-based Fake News Detection

---

Proposed and directed by:

Dr. Ahmed Cherif Mazari

Presented by:

Ms. Racha Mitidji

2022/2023

Acknowledgement

I would like to begin by expressing my profound gratitude and boundless appreciation to Allah for granting me the strength and perseverance to complete this report.

I extend my heartfelt thanks to my supervisor Dr.Ahmed Cherif Mazari for his unwavering patience, encouragement, constant support, and invaluable guidance throughout the course of this project.

Furthermore, I sincerely acknowledge and appreciate the professors who imparted their wisdom and shared their extensive expertise over the duration of my five-year academic journey.

I would like to express my sincere gratitude to the esteemed jury, who dedicated their time and expertise to evaluate my work.

I used ChatGPT for proofreading.

Dedication

With deep gratitude and heartfelt affection, I lovingly dedicate this humble work to my extraordinary family. Throughout my journey, they have always believed in me, offering unwavering support at every twist and turn.

Additionally, I extend my sincerest appreciation to Imene, Lyna, and Asma for their invaluable advice and guidance.

Lastly, I express my heartfelt thanks to Generative AI technologies. They have been more than just helpful tools; they have been exceptional brainstorming companions, sparking ideas and pushing the boundaries of my thinking.

Abstract

The spread of fake news in various online platforms has become a major problem that can influence public opinion and decision-making. Recent Artificial intelligence models using Deep learning architectures have shown promise solutions in detecting fake news. This report explores the effectiveness of deep learning and transfer learning models, including BERT, CNN, LSTM, Bi-LSTM, GRU, and Bi-GRU**.** The proposed approach combines the power of deep learning models with the transfer learning models to achieve better fake news detection accuracy. The evaluation metrics encompass Accuracy, Precision, Recall, and F1-score. The results show that the proposed approach on the ISOT dataset, outperforms existing methods and achieves high accuracy values in detecting fake news.

Keywords

Fake News detection, Deep Learning, Misinformation, Disinformation, Text Classification

ملخص

أصبح انتشار الأخبار المزيفة في مختلف المنصات على الإنترنت مشكلة كبيرة يمكن أن تؤثر على الرأي العام وصنع القرار. أظهرت نماذج الذكاء الاصطناعي الحديثة التي تستخدم بنيات التعلم العميق حلولاً واعدة في اكتشاف الأخبار المزيفة. يستكشف هذا التقرير فعالية التعلم العميق ونماذج نقل التعلم، بما في ذلك CNN , LSTM , Bi-LSTM BERT , Bi-GRU , GRU.النهج المقترح يجمع بين قوة نماذج التعلم العميق و نماذج نقل التعلم لتحقيق دقة أفضل في الكشف عن الأخبار المزيفة. تشمل مقاييس التقييم الدقة والدقة الإحصائية والاسترجاع ونسبة F1.تظهر النتائج أن النهج المقترح في مجموعة بيانات ISOT ، يتفوق على الأساليب الحالية ويحقق قيم دقة عالية في اكتشاف الأخبار المزيفة.

الكلمات المفتاحية

كشف الأخبار المزيفة، التعلم العميق، الأخبار الخاطئة، الأخبار المضللة، تصنيف النصوص

Résumé

La diffusion de fausses nouvelles sur diverses plateformes en ligne est devenue un problème majeur qui peut influencer l'opinion publique et la prise de décision. De récents modèles d'intelligence artificielle utilisant des architectures d'apprentissage en profondeur ont montré des solutions prometteuses pour détecter les fausses nouvelles. Ce rapport explore l'efficacité des modèles d'apprentissage en profondeur et transfert d'apprentissage, notamment BERT, CNN, LSTM, Bi-LSTM, GRU et Bi-GRU. L'approche proposée combine la puissance des modèles d'apprentissage en profondeur avec des modèles transfert d'apprentissage pour obtenir une meilleure précision de détection des fausses nouvelles. Les paramètres d'évaluation englobent l'exactitude, la précision, le rappel et le score F1. Les résultats montrent que l'approche proposée sur l'ensemble de données ISOT surpasse les méthodes existantes et atteint des valeurs de haute précision dans la détection des fausses nouvelles.

Mots Clés

Détection de fausses informations, Apprentissage approfondi, Désinformation, Mésinformation, Classification de texte

Table of content

List of Figures ..... **i**
List of Tables ..... **ii**
List of abbreviations ..... **iii**
General Introduction **1**
Chapter 1 : Fake News Detection Problem **2**
1.1 Introduction 3
1.2 What is Fake News? 3
1.3 Fake News Concepts 3
1.3.1 Propaganda 3
1.3.2 Rumor 3
1.3.3 Satire and parody 4
1.3.4 Clickbait 4
1.3.5 Conspiracy theories 4
1.3.6 Hoax 4
1.3.7 Spam 4
1.4 Fake news categories 4
1.4.1 Misinformation 4
1.4.2 Disinformation 5
1.4.3 Mal-information 5
1.5 Fake News: Broad and Narrow Definition 5
1.6 Adverse Effects of fake news 6
1.6.1 Impact on health 6
1.6.2 Impact on society and individuals 7
1.6.3 Financial Impact 7
1.6.4 Political impact 8
1.7 Previous solutions to combat fake news 8
1.7.1 Knowledge-based approaches 8
1.7.2 Features-based approaches 14
1.7.3 Learning-based approaches 16
1.7.4 Measures of social media platforms 16
1.7.5 Media and News Literacy 17
1.8 Challenges in fake news detection 17
1.8.1 Need for Accurate Dataset 17
1.8.2 Diversity 17
1.8.3 Accuracy & malignant purposes 17
1.8.4 Ever-evolving features of Fake News 18
1.8.5 Velocity of diffusion 18
1.8.6 Volume 18
1.9 Conclusion 18
Chapter 2 : Deep Learning-Based Models for Fake News Detection **19**
2.1 Introduction 20
2.2 Bad content 20
2.3 Applied NLP 21
2.4 Word embeddings 22
2.4.1 Collobert and Weston (C&W) model 22

2.4.2 Word2Vec 22
2.4.3 GloVe 23
2.4.4 FastText 23
2.4.5 Contextual String Embedding (Flair) 23
2.4.6 WordPiece Embedding 23
2.4.7 Graphs from LOw-level unit Modeling (GLoMo) 24
2.4.8 Universal Language Model Fine-tuning (ULMFiT) 24
2.5 Related work 26
2.5.1 Methods and models 30
2.5.2 Related datasets 36
2.6 Conclusion 37
Chapter 3 : Proposed Approach 38
3.1 Introduction 39
3.2 Methodology 39
3.3 Proposed Approach Steps 40
3.3.1 Preprocessing and Sentence Encoding 40
3.4 Proposed models 42
3.4.1 CNN 42
3.4.2 LSTM 42
3.4.3 Bidirectional LSTM 42
3.4.4 GRU 43
3.4.5 Bidirectional GRU 43
3.4.6 BERT 43
3.5 Metrics Used for Models Performance Evaluation 44
3.5.1 Accuracy 44
3.5.2 Precision 44
3.5.3 Recall 44
3.5.4 F1-score 45
3.6 Conclusion 45
Chapter 4 : Experimentation and Evaluation 46
4.1 Introduction 47
4.2 Environment and Development Tools 47
4.2.1 Python Modules Used 47
4.2.2 Google Co-laboratory 50
4.3 Data Used for Training and Test 51
4.4 Experimentation Steps 53
4.4.1 Data Preprocessing 53
4.4.2 Tokenization and Sentence Encoding 55
4.4.3 Train, Validation and Test Sets Partitioning 58
4.4.4 Data preparation 58
4.4.5 Models Building 58
4.4.6 Hyperparameter Tuning 65
4.5 Results and Discussion 69
4.5.1 Training and Validation Results 69
4.5.2 Performance Evaluation 72
4.5.3 Comparison of proposed approach with existing works 72
4.6 Conclusion 73
General Conclusion 74
References 75
Web Links 87

List of Figures

Figure 1.1. Main types of fake news with representative examples. 5
Figure 1.2. Automatic fact-checking process. 13
Figure 1.3. Approaches for fake news detection. 15
Figure 2.1. Illustration that shows hierarchy of Word Embeddings. 25
Figure 2.2. A nested pie chart depicting the proportion of both published articles and popular models for each year. 26
Figure 2.3. A pie chart of the benchmark datasets utilized in research on detecting fake news. 37
Figure 3.1. Proposed Approach Overview flowchart. 40
Figure 4.1. ISOT Dataset real news word-cloud. 52
Figure 4.2. ISOT dataset fake news word-cloud. 52
Figure 4.3. Preprocessing steps illustration. 54
Figure 4.4. BERT tokenization and sentences encoding illustration. 56
Figure 4.5. Python code for tokenizing and encoding sentences in DL models. 57
Figure 4.6. CNN architecture. 59
Figure 4.7. Python code for CNN model building. 60
Figure 4.8. GRU architecture. 60
Figure 4.9. Python code for GRU model building. 61
Figure 4.10. Bi-GRU architecture. 61
Figure 4.11. Python code for Bi-GRU model building. 62
Figure 4.12. LSTM architecture. 63
Figure 4.13. Python code for LSTM model building. 63
Figure 4.14. Bi-LSTM architecture. 64
Figure 4.15. Python code for Bi-LSTM model building. 64
Figure 4.16. Python code for BERT model building. 65
Figure 4.17. Python code for hyperparameter tuning using random search method. 67
Figure 4.18. Results of GRU Model's Hyperparameter Tuning Process. 68
Figure 4.19. Training and validation accuracy and loss graphs of GRU model. 69
Figure 4.20. Training and validation accuracy and loss graphs of Bi-GRU model. 70
Figure 4.21. Training and validation accuracy and loss graphs of CNN model. 70
Figure 4.22. Training and validation accuracy and loss graphs of LSTM model. 71
Figure 4.23. Training and validation accuracy and loss graphs of Bi-LSTM model. 71
Figure 4.24. Training and validation accuracy and loss graphs of BERT model. 71

List of Tables

Table 1.1. Details regarding popular fact-checking websites. 10
Table 1.2. A summary of the methodology and actions taken by several fact-checking websites and digital resources. 11
Table 2.1. A brief overview of the published papers focused on fake news detection. 26
Table 4.1. Description of ISOT dataset. 51
Table 4.2. Distribution of Data. 58
Table 4.3. Hyperparameters values chosen for optimization. 66
Table 4.4. Hyperparameters for proposed models 67
Table 4.5. Optimal Hyperparameters from Random Search for Deep Learning Models. 68
Table 4.6. The performance of suggested models. 72
Table 4.7. Comparison of proposed approach with previous contributions' results. 73

List of abbreviations

AE	Auto-Encoder.
ALBERT	A Lite BERT.
ASC	Adaptive Subgradient Clustering.
BART	Bidirectional Auto-Regressive Transformers.
BayesNet	Bayesian Network.
BERT	Bidirectional Encoder Representations from Transformers.
BiLSTM	Bidirectional Long Short-Term Memory.
CBOW	Continuous Bag of Words.
CNN	Convolutional Neural Network.
CvC	Class versus Cluster.
CVPS	Cross-Validated Probability of Success.
C&W	Collobert and Weston.
DeBERTa	Decoding-enhanced BERT with Disentangled Attention.
DistilBERT	Distillable Bidirectional Encoder Representations from Transformers.
DistilRoBERTa	Distilled Robustly Optimized BERT.
DSSM	Deep Structured Semantic Model.
DT	Decision Tree.
ET	Extra Tree.
GB	Gradient Boosting.
GLoMo	Graphs from LOw-level unit Modeling.
GloVe	Global Vectors for Word Representation.
GPT	Generative Pre-trained Transformer.
GRU	Gated Recurrent Unit.
IBk	Instance-Based Learner.
IR	Information Retrieval.
JRip	Repeated Incremental Pruning to Produce Error Reduction.
KB	knowledge base.
KG	knowledge graph.
KLR	Kernel Logistic Regression.
KNN	K-Nearest Neighbors Algorithm.
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm.
LMT	Logistic Model Trees.
LR	Logistic Regression.
LSTM	long short-term memory.
LSVM	Linear Support Vector Machine.
LWL	Locally Weighted Learning.
ML	Machine Learning.
MLP	Multi-Layer Perceptron.
MNB	Multinomial Naive Bayes.
NB	Naive Bayes.
NLP	Natural Language Processing.
OLM	Ordinal Logistic Model.
OneR	One Rule classifier.
PAC	Passive Aggressive Classifier.
PCA	principal component analysis.
RF	Random Forest.
RFC	Random Forest Classifier.

RNN	Recurrent Neural Network.
RoBERTa	Robustly Optimized BERT Approach.
SGD	Stochastic Gradient Descent.
SMO	Sequential Minimal Optimization.
SVM	support vector machine.
TF-IDF	term frequency-inverted document frequency.
ULMFiT	Universal Language Model Fine-tuning.
WIHW	Weighted Instance Hardness.
XGB	Extreme Gradient Boosting.
XLNet	Extra-Long Neural Network.
ZeroR	Zero Rule method.

General Introduction

The Internet has become an integral part of people's lives, with millions of users worldwide sharing and consuming content on a daily basis. While this has provided individuals with unprecedented access to information, it has also opened new avenues for the spread of misinformation and fake news. A recent study found that false news stories are 70% more likely to be retweeted on Twitter than true stories (Vosoughi et al., 2018), making it more difficult to distinguish between fact and fiction, especially given the increasing complexity and volume of data. The 2020 US Presidential election serves as a clear reminder of the importance of addressing these issues as Facebook had to flag and remove over 120,000 posts for sharing false or misleading information[i].

The spread of fake news online can have far-reaching consequences for individuals, communities, and even entire nations. Fake news, which refers to intentionally false or misleading information that is presented as factual news, can be used to manipulate public opinion, spread fear and hate, and undermine trust in institutions and democratic processes. This has resulted in an urgent need to develop effective fake news detection techniques.

In response to this need, researchers have developed computational methods to automate the process of fake news detection. While manual fact-checking and crowdsourcing can be effective, they are not scalable or cost-effective. Machine learning-based approaches, such as deep learning and transfer learning models, have demonstrated potential in recognizing patterns and categorizing news articles as authentic or deceptive. Nonetheless, more research is required to improve the accuracy and reliability of these techniques.

The focus of the study is to propose an approach for detecting fake news. The report will explore the effectiveness of various deep learning and transfer learning models, namely CNN (Convolutional Neural Network), LSTM (long short-term memory), BiLSTM (Bidirectional Long Short-Term Memory), GRU (Gated Recurrent Unit), BiGRU (Bidirectional Gated Recurrent Unit), and BERT (Bidirectional Encoder Representations from Transformers).

To achieve the report objective, the ISOT Fake News dataset will be used, which is a widely recognized resource in the domain of combating fake news. This dataset will be preprocessed and employed to train the selected models. To evaluate the performance of the models, various metrics, including Accuracy, Precision, Recall, and F1-score, will be utilized. Furthermore, the proposed approach will be evaluated against existing methods to assess its effectiveness.The significance of this report lies in its potential to contribute to the development of more effective and efficient methods for detecting fake news.

The subsequent sections of this report are organized as follows: Chapter 1 introduces the topic of fake news and covers its terminology, consequences, measures taken to address this issue, and associated challenges. Chapter 2 examines prior research on fake news identification. In chapter 3, the proposed architecture is explained, and information on the dataset and classification models employed is provided. The findings of the experiments are given in Chapter 4. Lastly, the conclusion and future possibilities are discussed.

Chapter 1 : Fake News Detection Problem

1.1 Introduction

The proliferation of fake news has become a significant concern in recent years. The spread of misleading and false information can have a severe impact on individuals, institutions, and societies. As online platforms have become the primary source of news and information for many people, it has become increasingly difficult to identify and combat fake news. This chapter introduces the problem of fake news detection and discusses its adverse effects, previous solutions, and challenges.

1.2 What is Fake News?

The problem of fake news has been present long before the emergence of modern technology. It dates back to the time following the development of the printing press in 1439, which facilitated widespread dissemination of news (Shu et al., 2017). In 2016, the term "fake news" gained widespread usage during the US presidential election and became a buzzword[ii]. Since then, research on fake news has grown in popularity, defining it as mixing of false news, propaganda, hoaxes, junk news, and rumors in information (Pierri & Ceri, 2019). Despite the negative connotation, there is still no agreement on how to define or classify fake news, even within the field of journalism (Zhou & Zafarani, 2020).

To enhance the analysis of fake news and evaluation of relevant research, In the following, we discuss related definitions to fake news that have gained popularity through various studies related to detecting fake news. we include also both a broad and a narrow understanding of what constitutes fake news definition.

1.3 Fake News Concepts

1.3.1 Propaganda

Propaganda refers to intentionally misleading information that aims to promote a particular social or political agenda or interests (Glenski et al., 2020). It involves the use of deceitful tactics and selective presentation of information to affect the emotions, attitudes, and behaviors of the targeted audience for ideological, political, or religious purposes (Tandoc et al., 2017).

1.3.2 Rumor

Rumor is an unverified or doubtful story that spreads from person to another (Shu et al., 2020). Its etymology can be traced back to a Latin term meaning "noise" (Kumar Singh Manish et al., 2022). Despite not being based on news events, rumors are often accepted by the public (de Oliveira et al., 2021) and can refer to various topics of public interest (Kaliyar et al., 2021). When proven false, rumors become a form of misinformation (Shu et al., 2020). some scholars categorize rumors as a type of propaganda (Kumar Singh Manish et al., 2022).

1.3.3 Satire and parody

Satires and parodies are forms of news that incorporate comedic elements to comment on present-day occurrences, utilizing sarcasm and irony (Abdali et al., 2020; de Oliveira et al., 2021), and can be easily mistaken for legitimate news, posing a risk when disseminated without proper context (Kumar Singh Manish et al., 2022).

1.3.4 Clickbait

Clickbait refers to the use of deceitful or exaggerated headlines, frequently accompanied by attention-grabbing images, to entice readers to click on them, leading to a different site. It is considered as low-quality journalism, intended to generate revenue through advertising by increasing the number of clicks on advertisements (Abdali et al., 2020; Y. Chen et al., 2015; Hangloo & Arora, 2022; Volkova et al., 2017).

1.3.5 Conspiracy theories

Conspiracy theories involve unproven or untrustworthy explanations for events or situations that lack clear evidence to support their truth or falsehood (de Oliveira et al., 2021). they are frequently employed to influence political or social establishment (Saeed, 2022).

1.3.6 Hoax

A hoax, otherwise referred to as a prank or jape, involves creating a deliberate falsehood with the aim of presenting it as truth. The motivation behind it can be either for amusement or to deceive specific individuals (de Oliveira et al., 2021; Kaliyar et al., 2021).

1.3.7 Spam

Spam is a term used to describe unsolicited messages sent to a vast number of recipients, including digital platforms such as email. These messages often contain irrelevant or inappropriate content and are usually associated with an unrequested advertising campaign (Shu et al., 2020; Zhou & Zafarani, 2020).

1.4 Fake news categories

All above related concepts of fake news fall under three categories:

1.4.1 Misinformation

Misinformation refers to false information that is disseminated without the intent to deceive (de Oliveira et al., 2021), by individuals who lack knowledge about the topic, have cognitive limitations (Fetzer, 2004; Kucharski, 2016), or Unintended Mistakes that were made by news gatherers (Thota et al., 2018).

1.4.2 Disinformation

Disinformation is the deliberate spread of false or misleading information with the intention of deceiving a target audience (Fetzer, 2004; Hangloo & Arora, 2022; Kshetri & Voas, 2017).

1.4.3 Mal-information

The term "mal-information" refers to legitimate information that is disseminated with the intention of causing harm to others (Larzillière et al., 2018; Shu et al., 2020).

Misinformation and disinformation can be converted mutually. Figure 1.1 illustrates the main types of fake news with examples (example: fake news is an example of disinformation).



Figure 1.1. Main types of fake news with representative examples. (Shu et al., 2020)

1.5 Fake News: Broad and Narrow Definition

Fake news is a type of propaganda or yellow and deceptive journalism that deliberately spreads false information through traditional or online media platforms, as defined by Leonhardt[iii] in their article in the New York Times. (Allcott & Gentzkow, 2017; Shu et al., 2017) further specify that news articles that are intentionally and verifiably false and, have the potential to mislead readers are considered as fake news. (Mavridis, 2018) summarize these definitions, adding that fake news can aim to spread propaganda, influence political views, or be created for entertainment purposes, or generate profits.

(Gelfert, 2018) identifies common features among various definitions, including the role of social media in creating and spreading fake news, scarcity of accurate information, and the deliberate misleading of readers. He discusses the term fake news in-depth and proposes his own definition, stating that Fake news is the intentional dissemination of false or deceptive statements presented as legitimate news, which are specifically crafted to mislead.

(Sharma et al., 2019) criticize these definitions for being too narrow and propose their own definition, which covers news or messages disseminated through the media, containing false information, without considering the methods or motives involved. (Zhou & Zafarani, 2020) provide simple and broad definition of fake news as "false news." It covers different forms of communication such as speeches, declarations, claims, articles, social media posts, and other materials related to a given subject (Zhou & Zafarani, 2020). Two key factors that define fake news are the intention of deceiving the reader and the authenticity of the information presented (de Oliveira et al., 2021; Zhou & Zafarani, 2020).

Overall, the term fake news refers to false and often sensational information presented as relevant news (de Oliveira et al., 2021).

1.6 Adverse Effects of fake news

1.6.1 Impact on health

The abundance of inaccurate medical information on digital platforms has resulted in irreversible harm to public health (Dai et al., 2020), causing confusion and risky behaviors. The World Health Organization has labeled this widespread false information as an "infodemic"[iv], which has resulted in significant consequences globally. Examples include the unwarranted suspicion towards vaccines during Zika and Ebola outbreaks (Ferrara et al., 2016), and the proliferation of faked COVID-19 claims through social media (Ferrara, 2020). Social media platforms have implemented some policy changes to restrict the dissemination of health misinformation in response to pressure from healthcare professionals, lawmakers, and advocates (Mridha et al., 2021). However, it continues to have a significant impact on people's lives (Waszak et al., 2018), as demonstrated by a cancer patient's death due to relying on an online advertisement as a trustworthy source of information for an experimental cancer treatment[v]. In 2020, the International Fact-Checking Network documented over 3,500 false COVID-19 claims within two months[vi], which led to significant harm[vii]. This included more than 800 deaths caused by the ingestion of dangerous substances, such as methanol or alcohol-based cleaning products, due to the promotion of false coronavirus cures (A. Coleman, 2020). With the rising number of individuals searching for health-related news online (Mridha et al., 2021), the danger of fake news is undeniable.

1.6.2 Impact on society and individuals

The dissemination of fake news can have a major impact on individuals, who may face harassment, insults, and threats on social media (Mridha et al., 2021). This is particularly concerning since studies have demonstrated that a significant portion of the population, including young Americans, have difficulty differentiating between authentic and fake news (Stanford History Education Group and Robert R. McCormick Tribune Foundation, 2016), and depend on social media as their primary source of information[viii].

False information often leads to a change in people's beliefs and attitudes, as seen in the Middle East (Spohr, 2017) during the Russian invasion of Ukraine, where fake reports were disseminated alleging that the war was a hoax, media invention, or exaggerated by the West[ix]. Similarly, in India, misinformation about COVID-19 resulted in a nationwide lockdown[x].

Fake news not only confuses and create social instability[xi] but also impacts political and financial decision-making (Rasool et al., 2019), altering how people respond to authentic news (S. Ahmed et al., 2022; Shu, Bernard, et al., 2019). A recent Harvard study conducted across 142 countries found that major of users are concerned about misinformation (Knuutila et al., 2022).

The 2016 U.S. presidential election case demonstrated that fake news can be more widely disseminated on social media platforms like Facebook than mainstream news[xii]. This highlights the need for individuals to not believe in invalid information on social media (Mridha et al., 2021) and for society to address the issue of fake news.

1.6.3 Financial Impact

Fake news has a significant impact on the financial aspect, as proven by various studies and real-life examples. According to research conducted by CHEQ and the University of Baltimore, fake news costs the global economy $78 billion annually, with a yearly loss of $39 billion in the stock market (CHEQ, 2019). The impact of fake news can be seen in various industries, including the cryptocurrency market, where the promotion of a particular coin through a fraudulent 'John McAfee' Twitter account caused its value to surge, leading to a lucrative buying and selling opportunity before dropping back to its original level[xiii]. The World Economic Forum identified the proliferation of digital misinformation as a critical technological and geopolitical threat in 2013 (Lee Howell, 2013). Other instances of fake news campaigns include a fake report about the effects of iodized salt on radiation from the Fukushima nuclear leak, which led to a sudden shortage of salt in Chinese supermarkets[xiv], and a $130 billion decline in the stock market caused by a fabricated news report that US President Barack Obama had been injured in an explosion[xv]. Furthermore, fake news has a negative impact on businesses, by damaging their brand reputation, decreasing the demand for their products and services (C. Zhang et al., 2019), and promoting an unethical business culture, as dishonest businessmen may propagate fake news or reviews to boost their profits. Therefore, the issue of fake news poses a critical concern in the business world and industries (Mridha et al., 2021).

1.6.4 Political impact

The phenomenon of fake news has gained significant attention from the media due to its impact on democratic processes, particularly during the 2016 US presidential election (Mridha et al., 2021). (Mavridis, 2018) discovered that teenagers in a small Balkan town had created over 100 websites to earn money from advertising by posting fake news about the election. This resulted in a considerable amount of misinformation, and some work suggests that such false stories, including those about the Pope's support for Donald Trump and Hillary Clinton's ISIS email leak, generated greater public interest than traditional news outlets[xvi], potentially lead the election in favor of Trump[xvii]. Moreover, recent research has shown that fake news has also affected other important events such as the UK Brexit referendum in 2016 (P. N. Howard & Kollanyi, 2016) and the French presidential elections of 2017 (Ferrara, 2017). The danger of fake news lies in its ability to influence people's opinions and beliefs, which can have real-world consequences. For instance, a three-month-long diplomatic standoff in the Gulf in 2017 was partly caused by a fake news report[xviii]. Additionally, False news stories have had a direct impact on policy, as President Trump tried to eliminate the first amendment after a false rumor circulated on WhatsApp led to mob violence in India (Thota et al., 2018).

1.7 Previous solutions to combat fake news

1.7.1 Knowledge-based approaches

Evidence-based or Knowledge-based Fact-Checking is well-known method for combating false news and disinformation. This approach involves fact-checking, a process that was originally created in journalism to confirm the accuracy of news articles by cross-referencing them with reliable external sources.

This section will examine traditional fact-checking, also known as manual fact-checking, and automatic fact-checking.

1.7.1.1 Traditional fact-checking

In general, manual fact-checking can be categorized into two types: fact-checking conducted by experts, and Crowdsourcing.

- **Expert-based manual fact-checking**

This type of fact-checking is usually carried out by a small group of highly reputable domain experts' fact-checkers, it is effective and produces highly precise outcomes (Zhou & Zafarani, 2020). However, due to the ever-growing volume of content on different platforms, this approach is not easily scalable and can be costly. Besides, the number of qualified journalists in fact-checking is decreasing, as pointed out by Jennings.

Lately, numerous websites have rised with a focus on expert-based fact-checking, aimed to serve the public (Zhou & Zafarani, 2020) by evaluating, rating, and debunking false news on online platforms (Hangloo & Arora, 2021). Examples of such websites include BSDetector, AltNews, APF Fact Check, Hoaxy, Snopes, and PolitiFact, which are listed in Table 1.1. In addition, we provide a summary of the approach and set of measures taken by several fact-checking websites and digital resources in identifying and countering false information on online platforms shown in Table 1.2.

Some of these websites go beyond evaluating information authenticity and provide additional details to assist in pinpointing topics that should be checked thoroughly to ensure their accuracy. For instance, PolitiFact offers a scorecard that displays statistics on the credibility of statements regarding a specific subject (Zhou & Zafarani, 2020), which can be used to verify the authenticity of a topic (J. Zhang et al., 2018). Similarly, HoaxSlayer classifies articles and messages into different types of misinformation such as hoaxes, spams, and fabricated news, providing valuable insights for comparative analysis. The Reporters Lab at Duke University maintains a comprehensive list of over 200 fact-checking websites across different countries and languages (Zhou & Zafarani, 2020).

Overall, these expert-based fact-checking websites can provide a reliable source for detecting fake news. In fact, PolitiFact and GossipCop have played a significant role in developing publicly available fake news datasets (Zhou & Zafarani, 2020), such as LIAR and FakeNewsNet.

Table 1.1. Details regarding popular fact-checking websites. (Zhou & Zafarani, 2020)

Website	Topics Covered	Content Analyzed	Assessment Labels
PolitiFact[1]	American politics	Statements	True; Mostly true; Half true; Mostly false; False; Pants on fire
The Washington Post Fact Checker[2]	American politics	Statements and claims	One pinocchio; Two pinocchio; Three pinocchio; Four pinocchio; The Geppetto checkmark; An upside-down Pinocchio; Verdict pending
FactCheck[3]	American politics	TV ads, debates, speeches, interviews, and news	True; No evidence; False
Snopes[4]	Politics and other social and topical issues	News articles and videos	True; Mostly true; Mixture; Mostly false; False; Unproven; Outdated; Miscaptioned; Correct attribution; Misattributed; Scam; Legend
TruthOrFiction[5]	Politics, religion, nature, aviation, food, medical, etc.	Email rumors	Truth; Fiction; etc.
FullFact[6]	Economy, health, education, crime, immigration, law	Articles	Ambiguity (no clear labels)
HoaxSlayer[7]	Ambiguity	Articles and messages	Hoaxes, scams, malware, bogus warning, fake news, misleading, true, humor, spams, etc.
GossipCop[8]	Hollywood and celebrities	Articles	0–10 scale, where 0 indicates completely fake news and 10 indicates completely true news

[1] https://www.politifact.com/
[2] https://www.factcheck.org/
[3] https://www.washingtonpost.com/news/fact-checker/
[4] https://www.snopes.com/
[5] https://www.truthorfiction.com/
[6] https://fullfact.org/
[7] http://hoax-slayer.com/
[8] https://www.gossipcop.com/

Table 1.2. A summary of the methodology and actions taken by several fact-checking websites and digital resources. (Hangloo & Arora, 2022)

Name	Tool/Extension	Methodology/Action
AltNews	Fact-checking website	Continuously monitors social media and mainstream media platforms for identifying incorrect information related mainly to Indian politics and entertainment, and evaluates the veracity of a claim by Manual Fact-checking
APF Fact Check	Fact-checking website	It uses many simple tools to verify online information. Fact-checking is carried out by editors and a worldwide network of journalists
BS Detector	It is an extension of Google Chrome, Mozilla	Identifies and marks fake and satirical news sites, as well as other suspected news sources. It puts a warning label to the top of potentially dangerous websites, as well as identifies fake links on Facebook and Twitter
Emergent	Fact-checking website	Emergent is a real-time rumor tracker that assesses news credibility and gives a True, False, or Unverified label
Fact-Checker	Fact-checking website	A project of The Washington Post, grades news articles from zero to four "Pinocchios" based on the factual accuracy of their content
FakerFact	A Chrome and Firefox extension	Distinguishes a fake news article from the real one and categorizes it as opinion, satire, agenda-driven, journalism, and sensationalism
InVid Verification Plugin	Use with Chrome, Firefox	A plugin to debunk fake images and videos. The tool uses reverse-image searching to debunk fake videos and also provide the users with metadata to take informed decision
PolitiFact	Fact-checking website	Tests the statements made on the Internet by political analysts and politicians and rate them. Its journalists evaluate original statements and each statement receives a "Truth-O-Meter" rating as "True", "Mostly True", "Half True", "Mostly False", "False", and "Pants on Fire"
Snopes	Fact-checking website	Conducts in-depth fact-checking research on hot issues, which are frequently picked depending on reader interest. "True", "Mostly True", "Mostly False", "False", "Unproven", "Miscaptioned", "Misattributed" are some of the annotations used to classify the content
Reverse Image Search (TinEye)	Browser extension	Can be used to see if the image has been taken from somewhere online. The tool comes with a Compare feature, which can be helpful to see how your image differs from the original one
BOOM	Fact-checking website	Manually checks the posts, debunks fake news, and prevents further spread
SurfSafe	Browser Plugin	Alerts users about misinformation by scanning images and videos on the web pages they're looking at. Performs reverse-image search by looking for the same content that appears on trusted source sites and flagging well-known doctored images
YouTube Data Viewer	A web-based video verification tool	Simple tool for extracting hidden data and metadata from YouTube videos which is particularly valuable for locating original content

- **Crowd-sourced manual fact-checking**

Collective intelligence from a large population of individuals serves as the foundation of crowd-sourced fact-checking, which can be facilitated through popular crowd-sourcing platforms like Amazon Mechanical Turk. It was also used to create CREDBANK, a publicly available, large-scale fake news dataset. This approach has difficulties in managing the credibility and accuracy of fact-checkers due to their political inclinations and the presence of contradictory annotations, with having superior scalability. Filtering non-credible users and resolving conflicting fact-checking results become increasingly important as more fact-checkers participate (Zhou & Zafarani, 2020).

Although it has limitations, crowd-sourcing platforms offer fact-checkers the opportunity to provide more elaborate feedback, such as their emotions or positions, which can be utilized in fake news studies. One example of an early-stage crowd-sourced fact-checking website is Fiskkit[9], which enables users to upload articles, rate sentences within articles, and choose tags that describe the articles. These tags can be categorized into multiple dimensions, helping to identify patterns across fake and non-fake news articles (Zhou & Zafarani, 2020).

As major web and social media websites like Google, Facebook, Twitter, and Sina Weibo start to realize the importance of identifying fake news, it is believed that more crowd-sourced platforms will arise (Zhou & Zafarani, 2020). For instance, Twitter has introduced BirdWatch in January 2021[xix], a community-based approach that allows people to identify deceptive tweets and annotate them, with pilot participants ability to assess the accuracy of the annotations made by other contributors.

1.7.1.2 Automatic Fact-Checking

The process of manually fact-checking information, particularly on social media, becomes difficult to scale with the increasing volume of newly generated data. In response to this challenge, automatic fact-checking methods have emerged. These techniques make extensive use of various tools such as IR, NLP, and ML techniques, as well as network/graph theory, as reported by (Cohen et al., 2011), to verify the accuracy of statements made in news articles.

[9] https://fiskkit.com/

To enhance comprehension, we provide the following definitions of the principal concepts that frequently appear in automatic fact-checking literature:

- **knowledge** is a collection of (Subject, Predicate, Object) (SPO) triples that effectively represent the provided information (Nickel et al., 2015). As an example, it is possible to represent the information contained in the sentence "Donald Trump is the president of the U.S." as a triplet of (DonaldTrump, Profession, President) (Zhou & Zafarani, 2020).
- A **fact** is a knowledge (SPO triple) confirmed as true.
- A **knowledge base (KB)** a collection of facts.
- **knowledge graph (KG)** is a visual representation of the SPO (subject, predicate, object) triples present in a KB, where, entities are represented as nodes, and relationships are represented as edges connecting these nodes (Ciampaglia et al., 2015; Dong et al., 2014; Shi & Weninger, 2015; Zhou & Zafarani, 2020).

Automatic news fact-checking process can be divided into two stages: fact extraction (a.k.a. KB construction) and fact-checking (a.k.a. knowledge comparison). Figure 1.2 show the automatic fact-checking process.

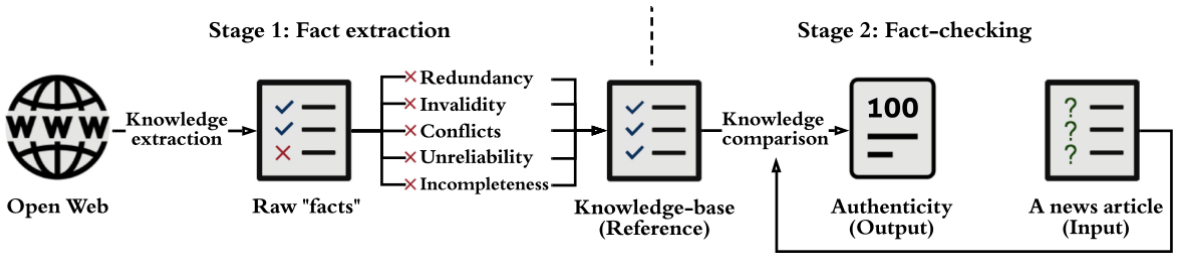


Figure 1.2. Automatic fact-checking process. (Zhou & Zafarani, 2020)

Fact extraction is the process of extracting raw "facts" from the internet and processing them to create a knowledge base or knowledge graph. Knowledge extraction can be done through a single reliable source or multiple potentially unreliable sources, which is known as open-source extraction (Zhou & Zafarani, 2020). Moreover, existing large-scale knowledge bases such as Freebase[10], NELL[11], PATTY[12], and Knowledge Vault (Dong et al., 2014) can be used to extract information and improve an existing knowledge base.

[10] https://doi.org/10.1145/1376616.1376746
[11] https://doi.org/10.1609/AAAI.V24I1.7519
[12] https://aclanthology.org/D12-1104

Fact-checking involves comparing the claims made in a news article or statement with the knowledge base (KB) constructed in the fact extraction stage. Non-existent SPO triples in a KB or KG need to be evaluated based on assumptions to determine their accuracy (Zhou & Zafarani, 2020).

There are various unresolved issues and potential areas for further research in this field. One such issue is the origin of the facts used to build knowledge bases (KBs) or knowledge graphs (KGs). While conventional sources like Wikipedia are often used, less attention has been given to fact-checking websites that offer expert analysis and justifications. Another crucial area for research is the development of dynamic KBs (KGs) that can automatically eliminate invalid information and adding new facts, particularly in news fact-checking where time is of the essence. Moreover, fast news fact-checking is crucial to identify fake news as soon as possible, which requires identifying check-worthy content and simplifying the KB to retain only valuable information (Zhou & Zafarani, 2020).

1.7.2 Features-based approaches

Existing learnings for fake news detection has been divided into two major categories: news content-based learning and social context-based learning. News content-based approaches (H. Ahmed et al., 2017; Ghosh & Shah, 2019; X. Zhang et al., 2015; Zhou & Zafarani, 2020) focus on extracting features related to information and writing style from fake news articles, as fake news publishers often use specific writing styles to appeal to and persuade a wide audience. Style-based techniques (Fazil & Abulaish, 2018; Ruchansky et al., 2017; Zhou & Zafarani, 2020) that employ linguistic features are useful in capturing the writing style of manipulators. Content-based methods have shown superior performance when compared to context-based methods since information about the propagation structure and user data is only accessible after the news has reached a large audience (Hangloo & Arora, 2022). However, relying solely on news content-based features can make it challenging to detect fake news accurately (Aswani et al., 2018; Ghosh & Shah, 2019; Vosoughi et al., 2017).

Therefore, it is also necessary to investigate the engagement of fake news articles with users. Social context-based learning (Ghosh & Shah, 2019; Gupta et al., 2012; Shu, Cui, et al., 2019; X. Zhang et al., 2015) involves examining the latent information between the user and news article. The semantic relationship between news articles and users can be used as a significant feature for fake news detection. Instance-based techniques focus on how users interact with individual social media posts to influence the credibility of specific news articles, while Propagation-based approaches consider the connections between important social media posts to guide the learning of credibility scores by transmitting reliability values among users, posts, and news items (X. Zhang et al., 2015).

The different approaches related to fake news detection are illustrated in Figure 1.3**.**

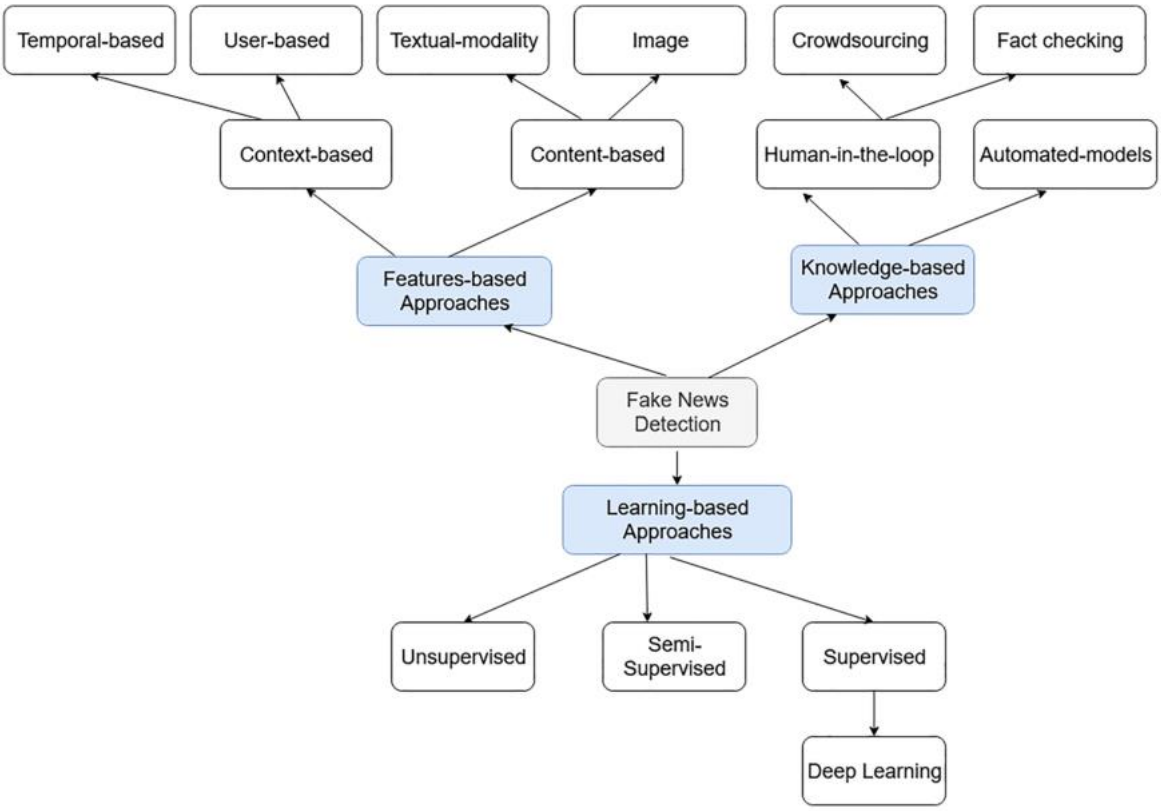


Figure 1.3. Approaches for fake news detection. (Kaliyar et al., 2021)

Many of the effective techniques currently used (Ghosh & Shah, 2019; Shu, Cui, et al., 2019; X. Zhang et al., 2015) involve utilizing unidirectional pre-trained word embedding models (such as GloVe, TF-IDF, and word2Vec) to analyze news content and context-level features. However, there is significant potential to incorporate bidirectional pre-trained word embedding models, which possess strong feature extraction capabilities.

1.7.3 Learning-based approaches

The study of fake news using learning-based techniques has led to the development of more effective methods for detecting and preventing the spread of false information (Meneses Silva et al., 2020). Various characteristics can be extracted and used in classification models such as naive Bayes, decision trees, logistic regression, and SVM, with the appropriate classifiers recommended from these frameworks (Azad et al., 2021; Orabi et al., 2020). However, further research is needed to develop more complex models and make better use of the extracted characteristics. Some examples of this include creating aggregation, probabilistic, ensemble, and projection methods (Collins et al., 2020; Oshikawa et al., 2018). For instance, aggregation algorithms can be developed to combine multiple feature representations into a weighted form, probabilistic models can be used to predict the likelihood of fake news, and ensemble approaches can be used to combine weak classifiers into a strong classifier (A. A. A. Ahmed et al., 2021) as Identifying false news based on a single criterion, such as the credibility of the source or the content of the news itself, is a challenging task (Paka et al., 2021). Projection techniques can also be used to map between original feature spaces and unique physical and behavioral spaces that may be more helpful for classification (Pilkevych et al., 2021).

However, existing solutions mainly rely on supervised learning, which requires a pre-annotated dataset of fake news for training the model. It is important to explore scenarios with limited or no labeled fake news articles and consider using semisupervised or unsupervised models for detecting false news. Unsupervised models are more practical as they can utilize unlabeled datasets, although supervised models may be more accurate when using a well-curated ground truth dataset (Preston et al., 2021).

1.7.4 Measures of social media platforms

The prevalence of false information and misleading news has increased on social media platforms, leading to potential harm such as the spread of inaccurate health advice and interference in political processes. To address this issue, social media platforms have implemented various measures. For example, Facebook has collaborated with third-party organizations in certain countries to help identify, review, and rate the accuracy of information[xx]. These organizations are certified through the non-partisan International Fact-Checking Network (IFCN). Meanwhile, Twitter has taken steps to combat misinformation around COVID-19 by introducing new labels and warning messages to provide users with additional context and information about tweets. This measure has made it easier for users to locate factual information and make informed decisions about tweets[xxi]. Snapchat has partnered with publishers to provide trustworthy news to its young users. The Discover section, which is supervised by an editorial team, is responsible for displaying reliable news and entertainment material. The sources for the content are carefully selected by the team to guarantee its authenticity. TikTok has launched a three-step strategy to combat misinformation, which includes improving misinformation regulations, extending reporting options and fact-checking, and opposing foreign intervention[xxii].

1.7.5 Media and News Literacy

To combat fake news, systematic media literacy education is necessary for the public (Schuetz et al., 2021; Şişman & Yurttaş, 2015). This education should cover awareness of social media, how to detect and challenge false information (Schuetz et al., 2021; Wasike, 2013), and various literacy skills, such as information literacy, news literacy, and digital literacy. Collaboration among library and information professionals and other stakeholders is important (Copenhaver, 2018; Fernandez, 2017; Wasike, 2013), and involvement from counselors and psychologists can help reduce negative effects (Hunt et al., 2018). Literacy in new media can help control the spread of fake news, and without media and news literacy education, disinformation cannot be stopped. Overall, media and news literacy are effective means of controlling the spread of false information on social media.

1.8 Challenges in fake news detection

1.8.1 Need for Accurate Dataset

Social media platforms place limitations on the collection of public data, resulting in a limited training dataset for identifying fake news. These datasets often lack crucial information necessary for detecting false news (Shao, Hui, et al., 2018). A comprehensive dataset is crucial in verifying the accuracy of news (Kumar Singh Manish et al., 2022).

1.8.2 Diversity

Fake news encompasses a wide range of false information that can take various forms, including rumors, satire news, disinformation, fake reviews, propaganda, hoaxes, and click-bait. This false information can have a significant impact on various aspects of people's lives (X. Zhang & Ghorbani, 2020).

1.8.3 Accuracy & malignant purposes

Fake news is deliberately crafted to resemble conventional news outlets and create confusion among readers, making it difficult to distinguish between genuine and fabricated information (Shao, Ciampaglia, et al., 2018; Shu et al., 2017; X. Zhang & Ghorbani, 2020). The structure of fake news is specifically tailored to take advantage of cognitive abilities, emotions, and ideological prejudices, posing difficulties for both humans and machines to identify it (Sharma et al., 2019).

1.8.4 Ever-evolving features of Fake News

The creation of new disinformation content is constantly evolving in response to advancements in automated fake news detection methods. Natural Language Processing (NLP) is frequently used to distinguish between false and legitimate news by analyzing writing style, since verifying facts from pre-defined knowledge bases remains a challenge. Consequently, current techniques for identifying fake news through content analysis involve extracting information from the news content itself and cross-checking the accuracy of the facts against knowledge bases (De Oliveira et al., 2020).

1.8.5 Velocity of diffusion

Detecting and mitigating the spread of fake news is a complex task as it can rapidly impact a vast audience, even when the information's authenticity is questionable (Friggeri et al., 2014). The transmission of such information is rapid and effortless, posing a challenge to verify its veracity. It makes it crucial to promptly verify its truthfulness, while also considering how it spreads throughout the network (Meel & Vishwakarma, 2020).

1.8.6 Volume

The production rate and quantity of fake news have made it impractical to fully fact-check everything. Human experts can no longer be relied upon for verifying news articles due to this volume (Shu et al., 2017). Furthermore, there is a growing need to create tools that can detect fake news at an early stage to prevent it from circulating on the internet (Liu & Wu, 2018).

1.9 Conclusion

In this chapter, we explored the problem of fake news detection, which is a pressing issue in the era of social media. We started by defining what fake news is, and we explored its different categories and concepts. We also discussed the broad and narrow definitions of fake news and the adverse effects it has on health, society, finance, and politics. Furthermore, we surveyed the previous solutions proposed to combat fake news and highlighted the challenges that prevent the development of effective fake news detection techniques.

As a result, fake news detection is a challenging task that requires the use of sophisticated machine learning and deep learning algorithms. Therefore, the next chapter will focus on exploring the state-of-the-art techniques for fake news detection.

Chapter 2 : Deep Learning-Based Models for Fake News Detection

2.1 Introduction

The problem of fake news has become increasingly prevalent in recent years, with the rise of social media and the ease with which information can be shared and disseminated online. As a result, there has been a growing body of research aimed at detecting and combating fake news. In this chapter, we provide a comprehensive overview of the state-of-the-art conducted in this area and explore related datasets. We focus on the methods and models used, such as deep learning, transfer learning, and machine learning.

2.2 Bad content

The widespread problem of fake news, the online world is filled with different types of harmful content that present significant challenges to users and the accuracy of information sharing. These forms include misleading information, propaganda, hate speech, and clickbait. Misleading information involves content that distorts or misrepresents facts, often with the intention to deceive or manipulate audiences. Propaganda refers to the deliberate spreading of biased or misleading information to shape public opinion. Hate speech includes expressions that promote discrimination, prejudice, or violence against individuals or groups based on attributes such as race, religion, or gender.

Lastly, clickbait, a common tactic used in digital media, employs sensationalized or misleading headlines and thumbnails to attract users to click on content that often fails to fulfill its promises. These various types of harmful content collectively undermine the credibility of online platforms, compromise the quality of information available to users, and can have far-reaching social and political consequences. Addressing and combating these issues is crucial for preserving the authenticity, reliability, and ethical standards of online content. Various researchers and experts have made significant contributions in addressing these issues.

One approach proposed by (Mazari et al., 2013; Mazari & Djeffal, 2022) is a novel concept-based approach for query expansion specifically designed for Arabic information retrieval. Their method enhances the retrieval process by expanding user queries, thereby improving the retrieval of relevant and reliable information.

To evaluate the effectiveness of combating bad content in Arabic contexts, (Mazari & Djeffal, 2021) constructed a comprehensive test information retrieval collection consisting of an Arabic corpus and a list of queries related to relevant documents. This collection provides a valuable resource for assessing and refining methods used to tackle harmful content in the Arabic language.

In the realm of sentiment analysis, (Mazari & Djeffal, 2021, 2022) utilized both traditional Machine Learning algorithms and Deep Learning models to analyze sentiment in Arabic Algerian social media corpus. Their work contributes to understanding the sentiment patterns within online content, which is crucial for identifying and combating the dissemination of harmful information.

To address the issue of toxic text and hate speech, (Mazari & Kheddar, 2023) created a new dataset for Algerian toxic text detection. Extracted from social media networks, this dataset was carefully labeled to distinguish hate speech, cyberbullying, and offensive language. Their efforts provide a foundation for developing effective models and techniques to identify and mitigate the presence of toxic content.

Furthermore, (Mazari et al., 2023) conducted a study on multi-aspect hate speech detection. They focused on classifying text into multiple labels, including 'identity hate', 'threat', 'insult', 'obscene', 'toxic', and 'severe toxic'. Their proposed approach leveraged the BERT model combined with Deep Learning algorithms to construct various ensemble learning architectures. This work enhances our understanding of the complexities associated with hate speech and contributes to the development of more robust detection systems.

2.3 Applied NLP

NLP is a subfield of artificial intelligence that focuses on the interaction between computers and humans through natural languages (Hirschberg & Manning, 2015). NLP plays a crucial role in AI research as it enables computers to understand, interpret, and generate human languages in a meaningful and useful manner. Recent years have witnessed significant advancements in NLP, primarily driven by deep learning techniques.

One notable advancement in NLP is the development of large-scale pre-trained language models (PLMs) that leverage deep learning techniques. These PLMs, such as BERT and GPT, have achieved remarkable results on various NLP tasks (Min et al., 2021). BERT, a transformer-based model, utilizes bidirectional self-attention to capture the context of words from both left and right sides (Devlin et al., 2018). It has outperformed other models on several NLP benchmarks, including GLUE, SQuAD, and NER.

GPT, another transformer-based model, employs unidirectional self-attention to generate natural language text from a given prompt (Radford et al., 2019). GPT-2 and GPT-3, its successors, have demonstrated the impressive capability to generate coherent and diverse texts across different domains and genres.

Additionally, transformer models, as introduced by (Vaswani et al., 2017) have revolutionized NLP tasks by utilizing the attention mechanism, which allows the model to weigh the importance of different words in a sentence when making predictions. Transformers have shown great success in machine translation and text summarization tasks.

Reinforcement learning, a type of machine learning, has also been applied to various NLP tasks, including dialogue systems and text summarization. In reinforcement learning, an agent learns to make decisions by interacting with an environment and optimizing a reward function. This approach has shown promise in improving the quality of responses and summaries generated by NLP models (Thomas et al., 2022).

These recent advances in NLP, encompassing PLMs, transformer models, and reinforcement learning, have significantly enhanced the capabilities of AI systems in understanding and generating human language (Guo & Yu, 2022). They have opened up new possibilities for applications in various domains, leading to improved performance in tasks such as question answering, sentiment analysis, named entity recognition, text completion, translation, and summarization. Furthermore, NLP research has also focused on addressing the challenge of fake news by developing techniques to detect and combat the spread of misinformation through natural language processing algorithms.

2.4 Word embeddings

Word embeddings are a type of representation that assigns each word in a text corpus a word vector of real numbers (weights). This method, also known as word vector representations, is widely used in both machine learning and deep learning models, with the goal of capturing the word's context and relationship with other words in the corpus. The embedding dimension of each word vector depends on the model complexity and the task at hand. When using word embeddings, there exist two approaches: generating embeddings by learning from a large text corpus or utilizing pre-existing embeddings and either retaining them or refining them with new data.

2.4.1 Collobert and Weston (C&W) model

The C&W model (Collobert et al., 2011) uses a multi-layered neural network architecture that incorporates both syntactic and semantic meaning. It is trained on a large dataset and is designed without any task-specific feature engineering. This makes it a versatile word representation model that can be applied to a variety of NLP tasks (Pathak et al., 2020).

2.4.2 Word2Vec

Word2Vec, a popular unsupervised word embedding technique proposed by (Mikolov et al., 2013), has gained significant attention in the field of natural language processing. The methodology utilizes a two-layer neural network to capture both semantic and syntactic relationships between words. By constructing a vocabulary and training word vectors in a defined dimensional space, Word2Vec can generate vector representations for each word in the input texts, enabling accurate modeling of vast corpora comprising hundreds of millions of words across 50-300 dimensions. Additionally, this approach offers two distinct models for construction: Continuous Bag of Words and Skip-Gram. CBOW predicts a target word given its context, whereas Skip-Gram employs a word to forecast its context. Notably, Skip-Gram excels in scenarios with limited training data and uncommon words, while CBOW demonstrates superior speed and accuracy when dealing with frequently encountered words (Chatzakou et al., 2017).

2.4.3 GloVe

Global Vectors for Word Representation or GloVe (Pennington et al., 2014) is an unsupervised model that creates word embeddings by using global matrix factorization and local context window to represent words. GloVe creates word embeddings by using a corpus to aggregate global word-word co-occurrence matrices, with each element representing how often a word appears in the context of another word. The model scans the corpus by looking for context terms within a defined window size and gives less weight to more distant words. This method allows GloVe to capture both syntactic and semantic information of words (Hu et al., 2022).

2.4.4 FastText

FastText, created by the Facebook AI research team (Bojanowski et al., 2017), is a library that uses character-level information to learn word representation effectively. It performs better than Word2Vec and GloVe on smaller datasets because those models view each word as a whole and disregard its morphological structure. FastText is particularly useful for social media content since it can handle unusual or unfamiliar words by dividing them into character n-grams. The word vector is obtained by summing up the character n-grams (Raza, 2021). Notably, the character n-gram approach employed by FastText enables capturing fine-grained linguistic patterns and enhancing the representation of morphologically rich languages.

2.4.5 Contextual String Embedding (Flair)

Flair is a type of character-level word embedding that considers words as sequences of characters within their context. This results in distinct embeddings for a word depending on the context it is used in. The model uses an LSTM to forecast the next character in a sequence, generating a hidden state for each character. It is a forward-backward LSTM with two layers, one going from left-to-right and the other going from right-to-left. A word embedding is created by combining the hidden state output after the final character in the forward layer with the hidden state output before the initial character in the backward layer. An example by (Akbik et al., 2018) showed that this method produced embeddings where the nearest neighbors, calculated using cosine distance, shared the same context as the word "Washington" in different scenarios, validating the effectiveness of Flair's contextual word representations.

2.4.6 WordPiece Embedding

WordPiece model is a data-driven method for generating a deterministic segmentation for any sequence of characters, which has been widely adopted in various natural language processing tasks to enhance the performance of models (Devlin et al., 2019). Initially, it solved Japanese/Korean segmentation in Google speech recognition (Schuster & Nakajima, 2012). It splits words into sub-word units to balance character flexibility and word efficiency, enabling models like BERT to effectively handle out-of-vocabulary words (Wu et al., 2016).

2.4.7 Graphs from LOw-level unit Modeling (GLoMo)

GLoMo, short for Graph-based Latent Model, is an innovative and unsupervised approach for enhancing the performance of various Natural Language Processing (NLP) tasks such as sentiment analysis and question answering. This transformative framework incorporates transfer learning techniques as discussed in (Pathak et al., 2020; Yang et al., 2019).

2.4.8 Universal Language Model Fine-tuning (ULMFiT)

ULMFiT, developed by (Howard & Ruder, 2018), is a transfer learning model that can be used for a range of natural language processing tasks. One such task is sentiment analysis, for which pre-trained models are utilized. The model uses a bi-LSTM language model, pre-trained on general domain and fine-tuned on target domain. It is a universal model that works regardless of document size, number, or label, and can be applied to different tasks with the same architecture. The model does not require domain-specific documents or labels (Pathak et al., 2020).

Below, Figure 2.1 illustrates hierarchy of Word Embeddings.

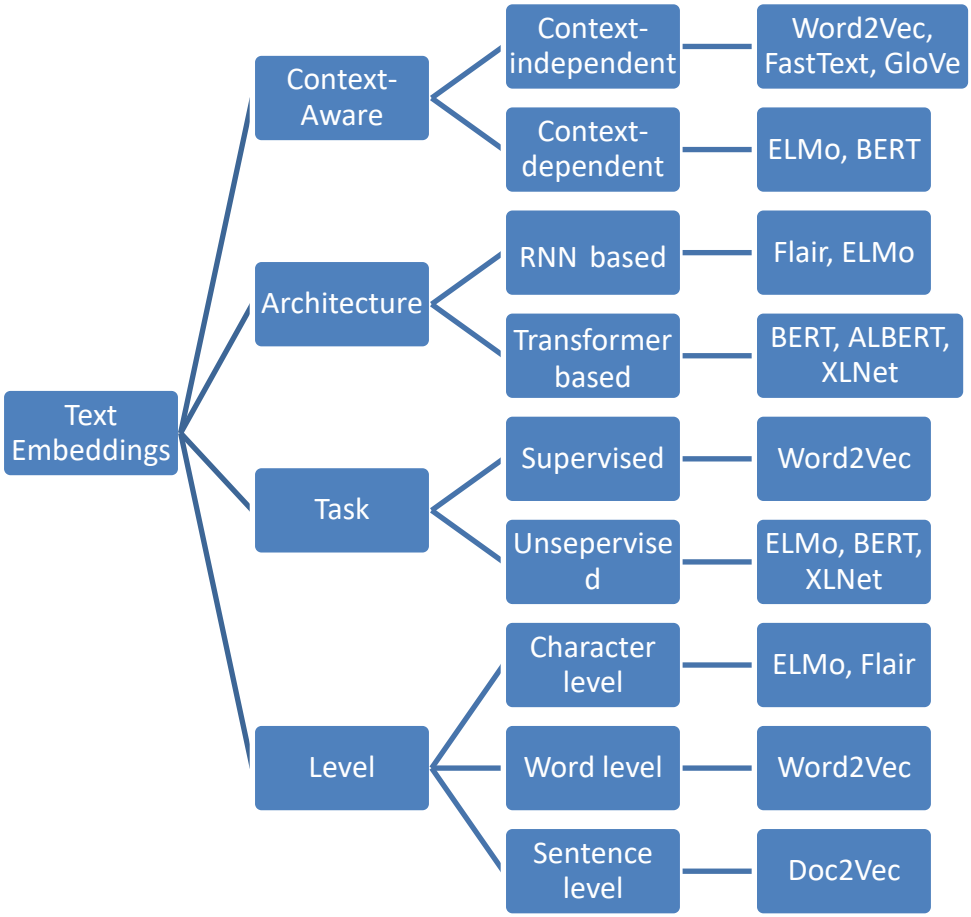


Figure 2.1. Illustration that shows hierarchy of Word Embeddings.

2.5 Related work

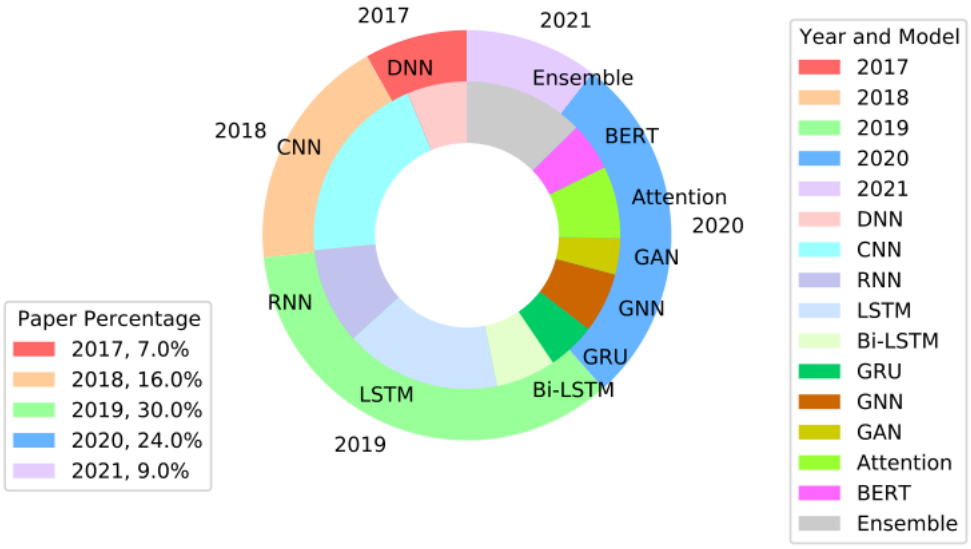


Figure 2.2. A nested pie chart depicting the proportion of both published articles and popular models for each year. (Mridha et al., 2021)

Table 2.1. A brief overview of the published papers focused on fake news detection.

Reference	Dataset	Contributions	Results	Models
(SATHVIK et al., 2023)	CoAID, GossipCop, and PolitiFact	Application and fine-tuning of BERT model for fake news prediction. The accuracy of BERT is compared with leading deep learning methods like LSTM, BiLSTM, CNN-BiLSTM, and ten well-known machine learning techniques.	BERT method achieves better Accuracy and F1-score of 91.94% and 92.06% for PolitiFact, 99.09% and 98.38% for CoAID, 85.59% and 82.40% for GossipCop.	CNN-BiLSTM, BiLSTM, LSTM, BERT, MNB, RF, DT, KNN, NB, SVM, LR, PAC, GB, Linear SVM
(Nadeem et al., 2023)	ISOT, FA-KES	hybrid model that uses DistilBERT for word embeddings, convolution neural network for spatial feature extraction, BiGRU for contextual	The model has an Accuracy of 61.1% and 99.3% for the FA-KES and ISOT datasets, respectively	DistilBERT, CNN, BiGRU, CapsNet

		feature extraction, and CapsNet with self-attention for hierarchical understanding of the features.		
(Sastrawan et al., 2022)	BuzzFeedNews, ISOT, LIAR, and FakeNewsNet	used deep learning methods with various architectures such as CNN, Bidirectional LSTM, and ResNet, combined with pre-trained word embeddings.	Accuracy of CNN= 99.88%, Bi-LSTM=99.95%, ResNet=99.90% on ISOT dataset	CNN, Bidirectional LSTM, and ResNet
(Seddari et al., 2022)	BuzzFeedNews	hybrid system that combines both linguistic and knowledge-based approaches, utilizing eight features, including title, number of words, reading ease, lexical diversity, sentiment, Fact check, Coverage, and Reputation.	94% Accuracy	LR, RF, Additional Trees Discriminant, and XGBoost
(Jiang et al., 2021)	ISOT, KDnugget	proposed a stacking model approach by combining the outputs of several models and running them via another machine learning model and used TF-IDF as embedding technique.	99.94% Accuracy on the ISOT dataset and 96.05% on the KDnugget dataset.	CNNs, LSTM and GRU, DT, LR, KNN, SVM, RF
(Kaliyar et al., 2021)	Fake-News dataset extracted from Kaggle	Model that combines BERT and three parallel blocks of 1d-CNN having different kernel-sized convolutional layers with different filters, and usage of GloVe and BERT word embeddings	Accuracy = 98.90%	BERT, CNN
(Shifath et al., 2021)	dataset of 10,700 COVID-19 related posts	developed an ensemble classifier to identify fake news related to COVID-19 using eight pre-trained transformer models and a Multi-Layer Perceptron (MLP)	Accuracy 0.984 on the validation dataset and 0.979 on the test dataset	BERT, BART, RoBERTa, ALBERT, DistilRoBERTa, GPT-2, XLNet, DeBERTa, and M

(Blackledge & Atapour-Abarghouei, 2021)	ISOT, Combined Corpus Dataset	compared a combination of machine learning and deep learning models as baseline models with three pre-trained transformer models	Best F1-score value in DistilBERT= 0.750 when trained on the CC dataset and tested on the ISOT dataset and BERT= 0.702 when inverted	BERT, DistilBERT, DeBERTa, LR, NB, RF and LSTM
(Hakak et al., 2021)	ISOT and LIAR	26 extracted features classified using an ensemble model comprising of three machine learning models.	Accuracy = 100% on ISOT and 44.15% on Liar	Decision Tree, Random Forest, and Extra Tree
(Ozbay & Alatas, 2020)	BuzzFeed Political News, Random Political News, and ISOT	Implemented twenty-three supervised artificial intelligence algorithms in three datasets	Best score of Recall = 1 in ZeroR, WIHW, and CVPS. DT produced the highest mean Accuracy = 0.745, Precision = .0741, and F1-measure = 0.760 values	BayesNet, JRip, OneR, Decision Stump, ZeroR, SGD, CVPS, RFC, LMT, LWL, CvC, WIHW, Ridor, MLP, OLM, ASC, J48, SMO, Bagging, DT, IBk, and KLR
(Umer et al., 2020)	dataset from the Fake News Challenges (FNC) website	Proposed a model that combines CNN and LSTM with PCA and Chi-Square for Fake News Stance Detection	Accuracy = 97.8%	CNN, LSTM
(Kaliyar et al., 2020)	Kaggle fake news dataset	developed FNDNet, a deep neural network using convolutional neural network (CNN) architecture. It was analyzed using GloVe as a pre-trained word embedding.	Accuracy = 98.36%	CNN
(Kumar et al., 2020)	Dataset of 1356 news instances from various users via Twitter and media sources, FakeNewsNet dataset	Proposed CNN +BiLSTM ensembled model with attention mechanism.	Accuracy = 88.78%.	CNN, BiLSTM
(Jadhav &	LIAR dataset	The proposed	99% training Accuracy	LSTM

Thepade, 2019)		approach, DSSM-LSTM, aims to identify fake news through the classification of news based on semantic features.		
(Jwa et al., 2019)	FNC-1 dataset	Propose an automatic fake news detection model that improves upon BERT by mitigating the data imbalance problem.	F1-score = 0.746	BERT
(Monti et al., 2019)	Buzzfeed, Snopes and PolitiFact	Geometric deep learning model with two convolutionary layers and two-dimensional features	Accuracy = 92.7%	CNN,
(Kaliyar et al., 2019)	Kaggle fake news dataset	Evaluated different machine learning models for fake news detection.	Gradient boosting with highest Accuracy of 86%	RF, Multinomial Naive Bayes, GB, DT, LR, Linear-SVM
(Jain et al., 2019)	-	Include machine learning and deep learning models in one system.	Accuracy = 93.6%,	NB, SVM
(Reis et al., 2019)	2282 news articles related to the US election	Addition of new features for training classifiers	Accuracy = 85%	KNN, NB, RF, SVM, XGB
(Chen et al., 2018)	dataset used contains a variety of original Weibos, consisting of both 2,325 non-rumors and 1,257 rumors	method for detecting rumors as anomalies in credible micro-blogs using a combination of recurrent neural networks and auto-encoders for unsupervised learning	Accuracy = 92.49%, F1-score = 89.16%.	RNN, AE
(Y. Yang et al., 2018)	dataset of 20,015 news articles from the American presidential election	model called TI-CNN that combines text and image information to detect fake news based on explicit and latent features	F1-measure = 92%	CNN
(Ajao et al., 2018)	5,800 tweets centered on five rumor stories	Used a hybrid of CNN and RNN	Accuracy = 82%	CNN, LSTM, RNN
(Del	Italian Facebook	Proposed a classifier	Accuracy = 91%	Linear

Vicario et al., 2018)	dataset with 300K official media news, 50K incorrect information	with different features, such as semantic features, user-based features, structural features, sentiment-based features, and predicted features.		Regression, Logistic Regression, SVM, KNN, Multi-layer Perceptron L-BFGS algorithm, DT.
(Ahmed et al., 2017)	12 600 fake news articles from Kaggle and 12 600 truthful political articles.	Compared different feature extraction techniques on six different machine learning models.	Accuracy in SVM = 85%, LSTM = 92% and in KNN = 77%	LR, SGD, m, KNN, LSVM, SVM
(Ruchansky et al., 2017)	Twitter, Weibo datasets	proposed a hybrid model named CSI that comprises three modules: Capture, Score, and Integrate.	89.2% and 95.3% Accuracies on Twitter and Weibo datasets, respectively	LSTM, RNN, CSI
(Bajaj, 2017)	dataset consisting of 63,000 articles (50,000 authentic articles and 13,000 fake articles)	Evaluated different machine learning and neural network models to develop a classifier that could accurately identify fake news using only its content. GloVe embeddings were used.	The RNN with GRUs demonstrated the best F1-score (84%) and Recall (79%). Enhancing the CNN with attention resulted in the greatest Precision (97%)	RNN, GRU, CNN

2.5.1 Methods and models

2.5.1.1 Deep Learning models

Several scholars are developing remarkable neural networks to identify fake news.

The study (SATHVIK et al., 2023) explores four deep learning techniques, including BERT, LSTM, BiLSTM, and CNN-BiLSTM, to detect fake news. The models were trained on three large datasets, namely CoAID, GossipCop, and PolitiFact; and compared with ten machine learning classifiers. BERT method achieved better accuracy and F1-score and outperformed traditional machine learning models in terms of Accuracy.

(Nadeem et al., 2023) proposed hybrid model that uses DistilBERT for word embeddings, convolution neural network for spatial feature extraction, BiGRU for contextual feature extraction, and CapsNet with self-attention for hierarchical understanding of the features. The model is evaluated on two fake news datasets (ISOT and FA-KES). The model extends the concept of symmetry into deep learning approaches for natural language processing.

(Sastrawan et al., 2022) use deep learning methods with various architectures such as CNN, Bidirectional LSTM, and ResNet, combined with pre-trained word embeddings, trained using four different datasets. The authors perform data augmentation using the back-translation method to reduce data imbalances between classes. The results show that the Bidirectional LSTM architecture outperforms CNN and ResNet on all tested datasets.

(Jiang et al., 2021) suggest a technique of stacking, that involves combining the outputs of multiple models and feeding them into another machine learning model. Specifically, they integrate five traditional machine learning models (DT, LR, KNN, SVM, and RF) and three deep learning models (CNNs, LSTM, and GRU). The study employs two datasets: ISOT and KDnugget. Various preprocessing techniques were performed and then embedding techniques were used to tokenize the cleaned data. Their results were reported as the most accurate among previous research, with 99.94% Accuracy on the ISOT dataset and 96.05% Accuracy on the KDnugget dataset.

(Umer et al., 2020) suggested a method for examining the consistency between news article headlines and their main text using a neural network architecture comprising convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. They also integrated two dimensionality reduction techniques, principal component analysis (PCA) and Chi-square, into the model. The authors found that their approach achieved an Accuracy of 97.8% with less computation time.

(Kaliyar et al., 2020) developed a deep neural network called FNDNet using convolutional neural network (CNN) architecture. Their model achieved remarkable performance, surpassing previous results, and achieving an Accuracy rate of 98.36% on the fake news dataset available on Kaggle.

(Kumar et al., 2020) utilized a combination of a Convolutional Neural Network (CNN) and a Bidirectional Long Short-Term Memory (BiLSTM) model, enhanced with an attention mechanism. They analyzed their own datasets as well as the FakeNewsNet dataset and attained an Accuracy rate of 88.78%.

The detection of false news was addressed by the authors in (Monti et al., 2019) through the proposal of a geometric deep learning model. To achieve this, two convolutionary layers and two-dimensional features were utilized. The experiment involved three distinct datasets, namely Buzzfeed, Snopes, and PolitiFact. The outcomes revealed that the proposed model achieved an Accuracy rate of 92.7%. Further research could build upon this work by expanding to classify news based on topics and predicting fake news at an earlier stage.

In their paper (Jadhav & Thepade, 2019), the authors introduced a novel approach called Deep Structured Semantic Model (DSSM) with Recurrent Neural Networks (RNNs) to identify fake news by leveraging semantic features. The study utilized the LIAR dataset and employed several classifiers to classify the news. The findings indicated that the proposed DSSM-LSTM model achieved 99% Accuracy during training. Nevertheless, the authors did not demonstrate the use of semantic features in feature extraction, which raises concerns about the model's effectiveness in fake news detection.

(Ajao et al., 2018) employed a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to categorize fabricated news tweets. Their study involved the evaluation of three models, namely plain Long Short-Term Memory (LSTM), LSTM with dropout regularization, and LSTM-CNN hybrid model, on a dataset comprising around 5,800 tweets revolving around five rumor stories. The researchers found that the plain LSTM model demonstrated superior performance, while the LSTM model with dropout regularization suffered from underfitting and the LSTM-CNN hybrid model had limited data.

(Y. Yang et al., 2018) proposed a model called TI-CNN that combines text and image information to detect fake news based on explicit and latent features. They used a dataset of 20,015 news articles from the American presidential election, with almost 12,000 fake news articles. The model has two branches, one for text and another for image, which are merged into a single feature space to recognize fake news. They achieved a 92% F1-measure and show that image information is effective in detecting fake news.

(Bajaj, 2017) article deals with the issue of fake news from the perspective of natural language processing, with a focus solely on the content while disregarding any other cues. Various machine learning and neural network models were examined, using a dataset consisting of 63,000 articles (50,000 authentic articles and 13,000 fake articles). The author discovered that enhancing the CNN with attention resulted in the greatest Precision (97%), although this did not contribute significantly to the findings since only ten epochs of training could be carried out due to resource and time constraints. The RNN with GRUs demonstrated the best F1-score (84%) and Recall (79%).

(Ruchansky et al., 2017) proposed a hybrid model named CSI that comprises three modules: Capture, Score, and Integrate. The LSTM-based Capture module captures the temporal pattern of user activity in each article. The Score module capture source features using a fully connected network to extract vector representations. The Integrate component merges the Capture and Score outcomes for identification. The authors evaluated the proposed model using two datasets, one from Twitter and the other from Weibo, achieving Accuracies of 89.2% and 95.3%, respectively.

Figure 2.2 displays the proportion of research papers on detecting fake news using deep learning techniques, categorized by the classifiers utilized in the past few years.

2.5.1.2 Transfer Learning models

NLP research has recently shifted towards an increased emphasis on pre-trained models. BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) are two highly advanced language models that have been pre-trained on unlabelled text to acquire as much information from data as possible through unsupervised learning. In the second phase, these models are fine-tuned for specific tasks using a small set of labeled data, making it a semi-supervised sequence learning task. These models are also employed in studies on fake news. BERT employs bidirectional representations to gain knowledge and is typically more appropriate for NLP activities such as text categorization and interpretation. Conversely, GPT-2 utilizes unidirectional representation for anticipating the future through left-to-right context and is more effective for autoregressive tasks that require timely performance.

(Kaliyar et al., 2021) introduce the FakeBERT model to identify fake news, using a combination of BERT Transformers and CNN. They used datasets gathered during the 2016 U.S. Presidential Election and performed multiple machine and deep learning experiments on them with both GloVe and BERT word embeddings. The model outperformed other machine and deep learning models with an Accuracy of 98.90%.

To identify fake news related to COVID-19, (Shifath et al., 2021) developed an ensemble classifier by using eight pre-trained transformer models and a Multi-Layer Perceptron (MLP). They created this solution for the CONSTRAINT workshop and tested it on a dataset[1] of 10,700 COVID-19 related posts. The researchers selected eight transformer-based models, namely BERT, BART, RoBERTa, ALBERT, DistilRoBERTa, GPT-2, XLNet, and DeBERTa, to create an ensemble classifier. They attached an MLP to each model and merged the output of all the models. This merged output was then used as input for the final output generation through a learner. The Accuracy of their ensemble model was 0.984 on the validation dataset and 0.979 on the test dataset of the competition.

(Blackledge & Atapour-Abarghouei, 2021) focuses on out-of-distribution generalization, using two datasets: the ISOT Dataset and the Combined Corpus Dataset[2]. The authors compared a combination of machine learning and deep learning models as baseline models with three pre-trained transformer models (BERT, DistilBERT, DeBERTa). The study found that transformer models performed better than baseline models, with DistilBERT achieving the highest F1-score of 0.750 when trained on the CC dataset and tested on the ISOT dataset, while BERT achieved the highest F1-score of 0.702 when trained on the ISOT dataset and tested on the CC dataset.

In related work, (Zellers et al., 2019) suggest a Grover model for identifying false information that employs a language model similar to the structure of GPT-2, trained on millions of news articles.

[1] https://doi.org/10.1007/978-3-030-73696-5_3
[2] https://doi.org/10.1016/J.MLWA.2021.100032

(Jwa et al., 2019) conducted a study on automatic fake news detection. They utilized the BERT model to analyze the connection between the news story's headline and body text to detect fake news. Their findings show a significant improvement of 0.14 F1-score over the current state-of-the-art models.

(Chen et al., 2018) suggest a method for detecting rumors as anomalies in credible micro-blogs using a combination of recurrent neural networks and auto-encoders for unsupervised learning. The dataset used contains a variety of original Weibos, consisting of both 2,325 non-rumors and 1,257 rumors. They tested their approach and found a high level of Accuracy, with a score of 92.49%, and an F1-score of 89.16%.

2.5.1.3 Machine Learning models

A novel hybrid system has been proposed by (Seddari et al., 2022) to enhance the effectiveness fake news detection systems. It combines both linguistic and knowledge-based approaches, and utilizes eight features, including title, number of words, reading ease, and lexical diversity. The proposed system employs fewer features than most state-of-the-art approaches. According to the evaluation results on BuzzFeedNews dataset, this combination of features has achieved more than 94% Accuracy in detecting fake news, which is an improvement of over 7% compared to using only linguistic-based features.

(Hakak et al., 2021) introduce an ensemble classification approach aimed at identifying fake news. This method involves utilizing three classifiers, namely DT, RF, and ET, and extracting various features from the text to reduce dimensionality and eliminate irrelevant information. The features include the number of sentences, words, and characters, average sentence length, date, time, among others. They used two datasets, Liar and ISOT. Results from the experiment demonstrated that the performance of the machine learning algorithms improved significantly through feature engineering.

(Jain et al., 2019) introduced a combination of natural language processing techniques, Naïve Bayes classifier, and SVM to analyze a fake news dataset. Their model achieved an Accuracy rate of 93.6%, surpassing the baseline results by 6.85%.

The impact of various feature extraction techniques on outcomes was assessed by a group of researchers.

In their study, (Reis et al., 2019), the authors introduced a supervised learning approach to identify fake news, which included additional features in the training of classifiers to enhance the accuracy of detection. Their experiment focused on analyzing 2282 news articles concerning the US election and found that the RF classifier achieved an impressive Accuracy rate of 85%.

(Ahmed et al., 2017) conducted a comparison between two distinct methods of feature extraction, specifically TF and TF-IDF. Additionally, they tested the performance of six different machine learning classification models, namely SGD, SVM, LSVM, LR, KNN, and DT on two datasets. Their findings revealed that as the n-gram size increased, the accuracy of the models decreased.

Some researchers evaluated a lot of machine learning models on different datasets to choose the best individual model.

(Ozbay & Alatas, 2020) developed a model that integrates text analysis with supervised Machine Learning techniques to identify fake news. They employed twenty-three supervised algorithms, including DT, LWL, and KLR, and tested them on three distinct real-world datasets, namely BuzzFeed Political News, Random Political News, and ISOT Fake News. The outcomes of the study revealed that the decision tree algorithm produced the highest mean Accuracy, Precision, and F-measure values, while the algorithms ZeroR, WIHW, and CVPS achieved the best mean Recall value.

In their study, (Kaliyar et al., 2019) explored various machine learning algorithms such as Random Forest, Multinomial Naïve Bayes, Gradient Boosting, Decision Tree, Logistic Regression, and Linear-SVM to detect fake news. Their research indicated that gradient boosting outperformed the other algorithms, with an impressive Accuracy of 86% on the Fake News Challenge dataset.

Current advancements in the field of fake news detection prioritize the development of techniques for identifying fake news, rather than for detecting it at an earlier stage.

As per the findings presented in (Shu, Wang, et al., 2019), detecting fake news after at least 12 hours of its propagation which might not be timely enough. On the other hand, research in (Vosoughi et al., 2018) indicates that fake news can spread rapidly within a matter of minutes once it has been introduced. This was illustrated by the incident involving Elon Musk's Tesla team, where millions of dollars were lost within the first few minutes of the fake news about inviting people to give bitcoins in exchange for double the amount.

In a significant study detailed in (Del Vicario et al., 2018), researchers have introduced a novel approach to predicting the occurrence of fake news on social media in advance. They have proposed a new classifier that utilizes a range of features, including semantic, user-based, structural, sentiment-based, and predicted features, and they applied it to an Italian Facebook dataset comprising 300 K legitimate media news items and 50 K posts from various blogs and websites containing false or inaccurate information. The results indicate that the proposed technique for early detection of fake news achieved a training Accuracy of 77% and a testing Accuracy of 91%.

2.5.2 Related datasets

The FA-KES dataset[3] includes news events related to the Syrian war, with 804 articles, of which 376 were fake. The dataset was semi-supervised using a fact-checking labelling approach employing the Syrian Violation Documentation Center (VDC) database. The authors created the FakeNewsNet[4] dataset by combining claims from PolitiFact and GossipCop with social context and spaciotemporal information for each content. Thorne generated the FEVER[5] dataset for fact extraction and verification. They created claims from Wikipedia articles and classified them into three categories (Supported, Refuted, or NotEnoughInfo). Wang created the LIAR[6] dataset, which contained 12.8K manually labeled statements from PolitiFact. Various classifiers were evaluated for their accuracy in classifying news into six truthfulness levels, and incorporating additional features such as subject, speaker, and history improved the results. SemEval-2017 and SemEval-2019 workshops released two Twitter-based datasets[7] for rumor evaluation, aiming to analyze tree-structured conversations formed of tweets replying to an original rumorous tweet. However, these datasets are limited to social media rumors. (Gao et al., 2016) collected 5 million posts from Twitter and Sina Weibo microblogs containing 778 reported events, of which 64% were rumors. They tested several baseline algorithms and RNNs and LSTM models to classify events. Ferreira & Vlachos released the Emergent[8] dataset for stance classification of rumored claims. It includes 300 rumored claims and 2,595 associated news articles, collected, and labeled by journalists, with an estimation of their veracity (true, false, or unverified), from a variety of sources such as rumor sites (example: Snopes) and Twitter accounts (example: @Hoaxalizer). Mitra developed the CREDBANK[9] dataset, which consisted of 37 million event-related tweets manually labeled for their accuracy level. (Ferreira & Vlachos, 2016) conducted one of the earliest studies on the automated detection of fake news. They collected data from two fact-checking websites and used k-Nearest Neighbors classifiers to differentiate between true and false news. The PHEME project[xxiii] released multiple rumor-related datasets challenging the ability to assess news veracity.

A pie chart of used benchmark datasets is given in Figure 2.3**.**

[3] https://doi.org/10.1609/ICWSM.V13I01.3254
[4] https://doi.org/10.1089/big.2020.0062
[5] https://doi.org/10.18653/v1/n18-1074
[6] https://doi.org/10.18653/v1/P17-2067
[7] https://arxiv.org/abs/1704.05972v1
[8] https://doi.org/10.18653/V1/N16-1138
[9] https://ojs.aaai.org/index.php/ICWSM/article/view/14625

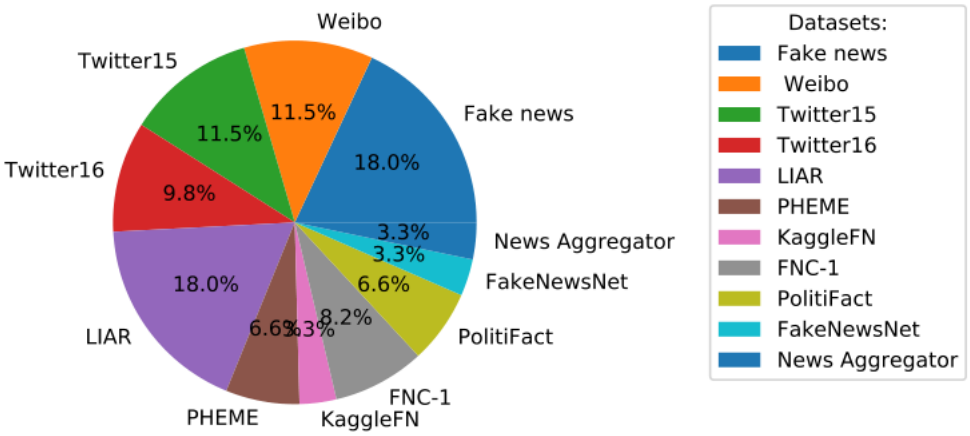


Figure 2.3. A pie chart of the benchmark datasets utilized in research on detecting fake news. (Kumar Singh Manish et al., 2022)

2.6 Conclusion

This chapter explored the use of deep learning-based models for fake news detection. First, we reviewed related work in the field, including methods and models such as deep learning models, transfer learning models, and machine learning models. In addition, we discussed related datasets that have been used in this domain. Overall, the use of deep learning and transfer learning models have shown promising results in fake news detection.

Chapter 3 : Proposed Approach

3.1 Introduction

In recent years, the rise of fake news has posed challenges to society, affecting public opinion, politics, and social stability. Detecting and countering fake news has thus become a crucial research area. In this chapter, we present our approach for fake news detection, which involves deep learning models and transfer learning using the BERT pre-trained language model. We discuss the choices made in designing the architectures, the preprocessing steps undertaken, and the method employed for fine-tuning the models to enhance its performance. Additionally, we outline various evaluation metrics, such as Accuracy, F1-score, Recall, and Precision, which enable a comprehensive assessment of the effectiveness of our approach.

3.2 Methodology

The aim of our contribution, as shown in Figure 3.1 below, is to design transfer learning and deep learning-based architectures that, after undergoing training, produces models with the ability to effectively identify and classify fake news content, thereby contributing to the mitigation of misinformation.

The process of text news classification involves multiple steps, which include dataset import, text preprocessing, dataset splitting, word embedding, model creation, training, hyperparameter adjustment, and evaluation. Initially, the dataset is imported in CSV format and goes through preprocessing. During this step, the dataset is cleaned, tokenized, and lemmatized to ensure its quality and consistency. Once the preprocessing is complete, the dataset is divided into train, test, and validation sets. The sequences of data are then padded/truncated to have equal lengths. Afterward, the word embedding step is performed, where each word in the dataset is transformed into a numerical representation that captures its semantic meaning. These representations are used as input for the models.

Following the word embedding step, the model creation phase takes place, where DL and Bert models are constructed. Next, hyperparameter adjustment is conducted to fine-tune the model's hyperparameters. This step involves random search technique to explore different combinations of hyperparameter values and identify the optimal configuration. Once the hyperparameters are determined, the training process begins using the training set. The model is trained on the input data, and an iterative optimization process occurs to update the model's parameters based on the training data's patterns and features.

After training, the model is evaluated using the validation set to assess its performance and make any necessary adjustments. Evaluation metrics such as Accuracy, Precision, Recall, and F1-score are commonly used to measure the model's effectiveness in classifying the text news. Finally, with the selected hyperparameters and well-trained models, the classification task is performed on the test set to assess the models’ generalization and its ability to classify new, unseen data accurately.

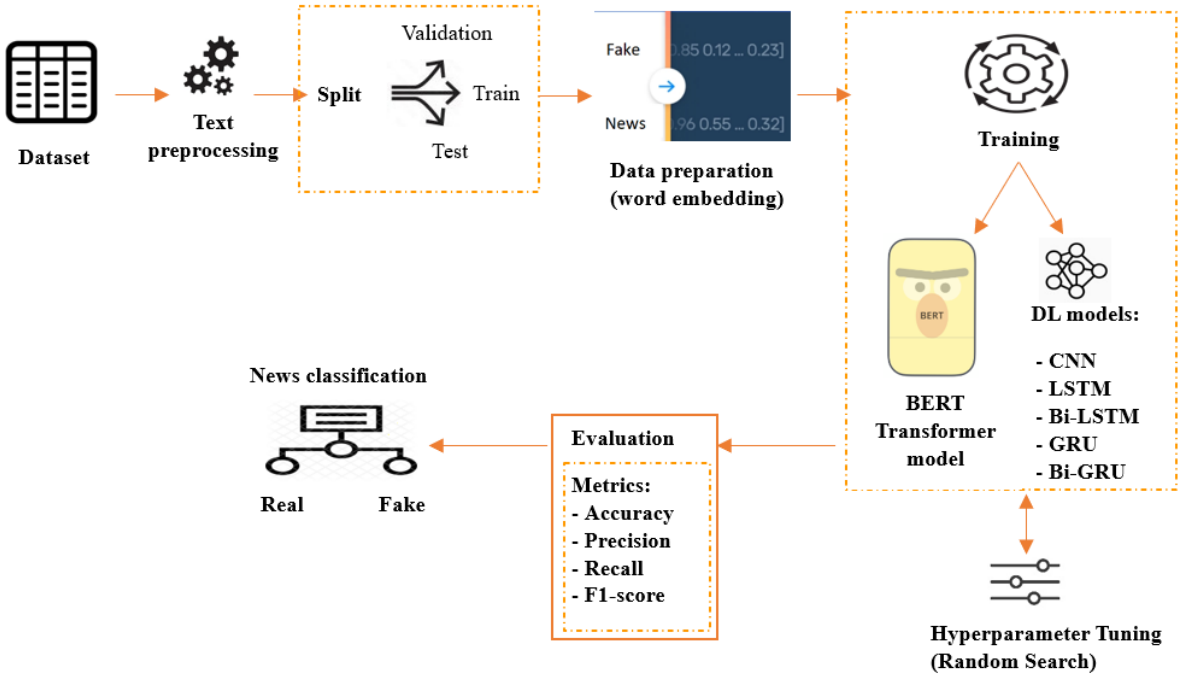


Figure 3.1. Proposed Approach Overview flowchart.

3.3 Proposed Approach Steps

Our method suggests multiple deep learning structures, and usage of BERT model, as a solution for dealing with the issue of identifying fake news. By utilizing the BERT model, we can effectively leverage both transfer learning and the attention mechanism. The subsequent sections provide a detailed explanation of the steps involved in our approach.

3.3.1 Preprocessing and Sentence Encoding

3.3.1.1 Data Preprocessing

In this report, we focus on text-based attributes for news categorization. we employ a series of text preprocessing steps to clean the dataset and prepare it for analysis, while preserving its textual characteristics, including spelling and grammatical mistakes. The objective is to remove irrelevant elements that could diminish the performance of models. The preprocessing steps are inspired by research articles, including (Blackledge & Atapour-Abarghouei, 2021; Hakak et al., 2021; Jiang et al., 2021).

The following section outlines the specific steps we follow:

- Removal of Metadata, Source Information and Attribution
- Removal of Time and Date Stamps
- Removal of Links
- Removal of Twitter Mentions
- Removal of Named Entities
- Lowercasing and Removal of Non-Alphanumeric Characters

In contrast to the approach taken by (Blackledge & Atapour-Abarghouei, 2021; Hakak et al., 2021; Jiang et al., 2021), we choose to not remove stop words to maintain the contextual meaning of sentences and to preserve valuable information that may be linked to these commonly used words.

3.3.1.2 Tokenization and Sentence Encoding

In the field of NLP, tokenization and sentence encoding play crucial roles in understanding and analyzing textual data. This process involves breaking down sentences into individual tokens and converting them into numerical representations that can be processed by DL and TL models.

In the following, the steps of tokenization and sentence encoding are detailed for DL and BERT models:

- **BERT tokenization and sentences encoding**

The process of tokenization and sentence encoding in BERT involves several sequential steps:

1. Load a pre-trained BERT tokenizer, which has been trained on a large corpus of text data. This tokenizer is essential for breaking down sentences into individual tokens, allowing BERT to understand and analyze the text.
2. To ensure proper sentence separation, a special token [CLS] is added at the beginning of each sentence, indicating the start of the input, and the special token [SEP] is appended at the end of each sentence, marking its completion.
3. Tokenize each sentence into sub-words using the word-piece tokenizer concept. If a word is not in the tokenizer vocabulary, break it into smaller sub-words until all sub-words are in the vocabulary. Prefix any sub-word that is not a whole word with two hashes ##.
4. After tokenization, each token is converted into a numerical ID based on the tokenizer's vocabulary, enabling BERT to process the text numerically.
5. Lastly, to ensure consistent input lengths, sentences are either padded with special tokens or truncated to a fixed length, facilitating uniform processing across all inputs.

- **Deep learning models tokenization and sentences encoding**

Tokenization is a fundamental pre-processing step for most natural language processing (NLP) applications, including fake news detection. The procedure consists of multiple steps as described subsequently:

1. Split the text into word segments.
2. Assign an index to each unique token in the text corpus to create a vocabulary.
3. Pad or truncate the token sequences to a fixed length that matches the input size of the DL model.
4. Convert each token into a numerical vector using an embedding layer that maps each index to a high-dimensional vector.

3.4 Proposed models

In our report, we conducted a comprehensive examination of existing literature on NLP in the context of fake news detection, and discovered that various approaches, such as ML, DL, and transformers, have demonstrated their capability to effectively tackle this task. To further investigate these approaches, we propose the following models:

3.4.1 CNN

Introduced by (LeCun et al., 1989), CNN is a powerful deep learning model widely used in computer vision tasks due to its ability to extract relevant features from images. CNNs are composed of convolutional layers that apply filters to input data, followed by pooling layers to reduce spatial dimensions. In recent years, CNNs have also been successfully applied to various NLP tasks, leveraging their ability to capture local patterns and hierarchical representations. Therefore, we explored the effectiveness of CNNs in fake news detection.

3.4.2 LSTM

LSTM (Hochreiter & Schmidhuber, 1997) is an advanced version of RNNs (Rumelhart et al., 1986) that facilitates the storage of past data in memory. LSTM is well-suited for processing sequential data, such as NLP problems. Therefore, we utilized LSTM for the task of fake news detection.

3.4.3 Bidirectional LSTM

The BiLSTM is comprised of two LSTM units, one processing input in the forward direction and the other in the backward direction. This arrangement enhances the network's access to information, thereby improving the algorithm's contextual understanding (Graves & Schmidhuber, 2005).

3.4.4 GRU

GRU is a simplified variant of the LSTM cell, introduced by (Cho et al., 2014) as a solution to the vanishing gradient problem encountered in traditional RNN architectures. GRU exhibits a more efficient training process and lower parameter count compared to LSTM, primarily due to the absence of an output gate (Jiang et al., 2021). It incorporates two fundamental gates, namely the reset gate and the update gate.

3.4.5 Bidirectional GRU

Originally introduced by (Cho et al., 2014), BiGRUs are a type of sequence processing model that consists of two GRUs, one processing the input in a forward direction and the other in a backward direction. This allows BiGRU to capture more context and information from the input sequence, which can lead to better performance.

3.4.6 BERT

Transformers have become increasingly dominant in the field of Natural Language Processing. BERT (Devlin et al., 2018), a transformer-based machine learning technique, has revolutionized the NLP field with its state-of-the-art performance. It incorporates two key aspects: its deep architecture enables efficient processing of lengthy sentences through the utilization of the 'attention' mechanism, and its bidirectional nature allows it to generate outputs based on the complete input sentence.

The process of developing a model for natural language processing can be expensive and time-consuming due to the extensive set of parameters involved. However, we can take advantage of pre-existing BERT models, allowing for efficient transfer learning. Among various models available, we choose to use the “bert-base-uncased” model, which contains a standard set of parameters and does not differentiate between upper-case and lower-case letters.

3.5 Metrics Used for Models Performance Evaluation

We utilize the metrics that are commonly used in previous studies and contributions regarding fake news (Blackledge & Atapour-Abarghouei, 2021; Jiang et al., 2021; Rajalaxmi et al., 2021) to assess the effectiveness of the models.

These metrics include the following four key measures:

- **True Positives (TP)** refer to correctly predicted positive values, where both the true class and predicted class are affirmative.
- **True Negatives (TN)** represent accurately identified negative values, where both the actual class and predicted class are negative.
- **False Positives (FP)** occur when the actual class differs from the predicted class, resulting in an incorrect positive prediction.
- **False Negatives (FN)** arise when the actual class is positive, but the predicted class is negative, indicating an incorrect negative prediction.

3.5.1 Accuracy

To assess the effectiveness of a machine learning classification algorithm in correctly categorizing data points, one common approach is to measure its accuracy. Accuracy refers to the proportion of accurately predicted samples out of the total number of samples.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})}$$

3.5.2 Precision

Precision is a metric that measures the correctness of positive predictions, acting as a simple indicator for the accuracy of less-represented categories. To calculate Precision, one divides the count of correctly predicted positive instances by the total count of predicted positive instances.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

3.5.3 Recall

The Recall metric quantifies the proportion of correct positive predictions among all potential positive predictions. Unlike accuracy, which focuses solely on correctly predicted positives, Recall considers positive predictions that were missed or overlooked.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

3.5.4 F1-score

The F1-Score, which is obtained by calculating the weighted average of Precision and Recall, incorporates both false positives and false negatives.

$$\mathrm{F1-Score} = \frac{(2 * Recall * Precision)}{(\mathrm{Recall} + \mathrm{Precision})}$$

3.6 Conclusion

In this chapter, we presented a comprehensive approach for detecting fake news using deep learning and transfer learning models. The proposed methodology encompassed various steps, including data preprocessing, tokenization, sentence encoding, and the utilization of CNN, LSTM, Bi-LSTM, GRU, Bi-GRU, and BERT models. To evaluate the effectiveness of the models, we suggested metrics such as Accuracy, Precision, Recall, F1-score. In the upcoming chapter, we will implement the suggested models to enable a comparative analysis and determine the most effective one.

Chapter 4 : Experimentation and Evaluation

4.1 Introduction

In this chapter, we explore into the execution and evaluation of the deep learning and transfer learning models proposed in Chapter 3 for fake news detection. We discuss the development environment, coding framework, dataset selection, preprocessing steps, and the training, validation, and testing of our architecture. We present the results obtained by applying the models on the ISOT dataset and provide a comprehensive analysis and discussion of these findings.

4.2 Environment and Development Tools

4.2.1 Python Modules Used

Python[1] is a high-level programming language that is interpreted and object-oriented. It has dynamic semantics and offers built-in data structures, dynamic typing, and dynamic binding. These features make it highly desirable for Rapid Application Development and as a scripting or glue language for connecting existing components. Python's syntax is simple and easy to learn, prioritizing readability and reducing program maintenance costs. It supports modules and packages, promoting code modularity and reuse. The Python interpreter and extensive standard library are freely available in source or binary form for major platforms and can be distributed without charge.

In the Google Colab execution environment, Python version 3.10.11 is used, along with specific modules mentioned here:

- **For Data Manipulation**

Pandas[2] is an open-source library that provides high-performance, user-friendly data structures and data analysis tools. It offers easy handling of data sheets, such as CSV and Excel files, including reading and manipulating data frames.

We utilized Pandas to read datasets from CSV files and manipulate their corresponding data frames.

[1] https://www.python.org/
[2] https://pandas.pydata.org/

Numpy[3] is a Python package that is openly accessible and has its source code available to the public. It provides an extensive collection of mathematical functions, random number generators, linear algebra operations, Fourier transformations, and other useful features. Additionally, Numpy has a user-friendly syntax and can be used on various hardware and computing platforms. It is also compatible with distributed systems, GPU processing, and libraries that handle sparse arrays. In our work, we rely on Numpy for various tasks, including defining variable types, modifying array shapes, and storing model outputs.

- **For Text Processing**

The **Re**[4] module, added in Python 1.5, is a vital component of computer programming that facilitates pattern matching and manipulation of strings. It provides Perl-style regular expression patterns, making it indispensable for tasks like data validation and text parsing.

In our work, the re module is used for preprocessing text using regular expressions. It helps remove unwanted elements such as metadata, timestamps, links, and mentions, ensuring a clean dataset for further analysis.

NLTK[5] stands for Natural Language Toolkit. It is a popular open-source Python library that provides various tools and resources for working with human language data. It offers a wide range of functionalities for tasks such as tokenization, stemming, tagging, parsing, semantic reasoning, and more. It also includes a collection of corpora, lexical resources, and pre-trained models for different NLP tasks.

We utilized NLTK library for various text processing tasks, such as tokenizing text into words, lemmatizing words to their base form, and downloading additional resources like WordNet.

SpaCy[6] is an efficient and user-freely open-source library for natural language processing. It offers a wide range of NLP functionalities such as tokenization, part-of-speech tagging, named entity recognition, and text classification. With its pre-trained models and the ability to train custom models, SpaCy is a versatile tool that provides state-of-the-art performance on various NLP tasks.

Spacy library is used for text preprocessing and named entity recognition tasks. Specifically, we use the en_core_web_sm model to remove named entities of type "PERSON" from our dataset. This step helps in reducing the influence of specific individuals' names on the classification model, ensuring a more generalized and robust model for detecting fake news.

[3] https://numpy.org/
[4] https://docs.python.org/3/library/re.html
[5] https://www.nltk.org/
[6] https://spacy.io/

- **For Deep Learning Implementation**

TensorFlow[7] is a freely available software library that enables dataflow and differential programming for a wide range of applications. It is particularly utilized in machine learning and deep learning. Originally created by Google in 2011 as DistBelief, TensorFlow was officially launched in 2017. This powerful library can operate on various computing units, such as CPUs and GPUs, and is compatible with multiple platforms, including mobile devices. we used the Version 2.12.0.

Keras[8] is one of the leading high-level neural network APIs written in Python. It has gained popularity for its simplicity and versatility, as it supports multiple back-end neural network computation engines. In fact, TensorFlow has adopted Keras as the high-level API for TensorFlow 2.0, further solidifying its prominence in the deep learning community.

Keras is primarily used for building and training a deep learning models for fake news detection.

Gensim[9] is an open-source Python library widely used for text analysis tasks. It offers efficient algorithms and data structures for tasks such as document similarity comparison, topic modeling, and word vector representations.

In our case, Gensim is used to load pre-trained word embeddings using the KeyedVectors class. This is done to use these embeddings for initializing the embedding layer of the deep learning model.

Scikit-learn[10], or sklearn, is a widely used open-source machine learning library for Python. It offers a comprehensive range of tools and algorithms for various machine learning tasks. With its user-friendly interface and integration with other Python libraries, sklearn provides researchers and practitioners with an efficient framework to implement, experiment with, and evaluate machine learning models.

We used it for splitting the dataset into training, validation, and test sets, and conducting hyperparameter optimization using RandomizedSearchCV, as well as calculating metrics such as Accuracy, Recall, Precision, and F1-scores.

[7] https://www.tensorflow.org/
[8] https://keras.io/
[9] https://pypi.org/project/gensim/
[10] https://scikit-learn.org/stable/

- **For transfer learning implementation**

Transformers Hugging Face[11] is a popular open-source library that provides pre-trained language models based on the Transformer architecture. It offers a wide range of pre-trained models, namely BERT, GPT-3, RoBERTa, XLM, DistilBER, for tasks such as natural language understanding, natural language generation, sentiment analysis, text classification, and more.

- **For data Visualization**

Matplotlib[12] is a widely used plotting library in Python that allows developers to create various types of visualizations, such as line plots, bar plots, scatter plots, and more. These visualizations can help in understanding data distributions, spotting trends or anomalies, and effectively communicating results to others. We used it to visualize the distribution of text lengths in the dataset.

4.2.2 Google Co-laboratory

Colab, also known as Colaboratory[13], is a cloud-based development platform offered by Google that features a Jupyter notebook interface. It allows users to write and execute Python code directly in a web browser, eliminating the need for local setup. One of its key advantages is the accessibility of powerful hardware accelerators such as GPUs and TPUs, which are ideal for computationally intensive tasks and machine learning applications.

In terms of session duration, Colab provides 12 hours of continuous execution. The available Random Access Memory (RAM) space is fixed at 12 GB, without the option to increase it freely. Additionally, while Google Drive offers storage space, Colab also permits using over 30 GB of storage within a session. It is important to note that any Python modules or files stored in this space will be deleted once the session ends.

Regarding execution options, Colab supports the utilization of CPUs, GPUs, and TPUs. In our specific case, we chose to leverage GPUs for model training due to their ease of use and ability to accelerate the execution of complex models such as BERT.

[11] https://huggingface.co/docs/hub/models-libraries
[12] https://matplotlib.org/
[13] https://colab.research.google.com/

4.3 Data Used for Training and Test

- **ISOT Fake News Dataset**

We tested our approach on the ISOT Fake news dataset [14], created by Ahmed (Ahmed et al., 2017, 2018) in the University of Victoria, it was entirely collected from real-world sources. The dataset includes 44898 data in total, consisting of 21417 real news (labeled as 1) and 23481 fake news (labeled as 0). The real news was collected by crawling articles from Reuters.com, while the fake news were collected from unreliable websites that were flagged by Politifact and Wikipedia. These articles were mostly released from 2016 to 2017. The dataset features include title, text (news body), subject, and date. The news subjects encompass various categories such as 'politicsNews,' 'worldnews,' 'News,' 'politics,' 'Government News,' 'left-news,' 'US_News,' and 'Middle east.' To train our models, we selected text as a feature in the ISOT dataset, which contains full-length articles. This dataset has been widely used in fake news detection studies (Blackledge & Atapour-Abarghouei, 2021; Hakak et al., 2021; Jiang et al., 2021; Ozbay & Alatas, 2020). Sample text of ISOT dataset have been presented in Table 4.1.

Table 4.1. Description of ISOT dataset.

Title	Text	Subject	Date
UK transport police leading in...	LONDON (Reuters) - British counter-terrorism.	worldnews	September 15, 2017
Pacific nations crack down on...	WELLINGTON (Reuters) - South Pacific island nation.	worldnews	September 15, 2017
Three suspected al Qaeda...	ADEN, Yemen (Reuters) - Three suspected al Qaeda...	worldnews	September 15, 2017
Chinese academics prod Beijing...	BEIJING (Reuters) - Chinese academics are publicly.	worldnews	September 15, 2017
Classic! Kid Rock Hits Back At...	Not much to say after this classic response from.	politics	Sep 2, 2017
'My Pillow' CEO Mike Lindell...	Who hasn t seen his commercials over and over and...	politics	Sep 1, 2017
Bitter John McCain Calls Trump...	What the heck! Senator John McCain just admitted..	politics	Sep 1, 2017
Muslim Activist Caught Sending...	This woman has no shame1 Muslim activist Linda...	politics	Sep 1, 2017

To overview the dataset, we carried out the visualization of dataset in the form word-cloud. Figure 4.1 represents the word-cloud of real news for ISOT dataset whereas Figure 4.2 represents the word-cloud of fake news for ISOT dataset.

The word cloud representing real news shows frequent occurrences of words such as 'Trump', 'say', 'Russia', 'House', 'North', and 'Korea'. Conversely, the word cloud for fake news prominently features words like 'VIDEO', 'Trump', 'Obama', 'WATCH', and 'Hillary'. The word 'say' is frequently seen in real news but not in fake news, while 'VIDEO' and 'WATCH' are commonly found in fake news but not in real news. These word clouds provide valuable insights to distinguish between the two categories of data.

[14] https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/

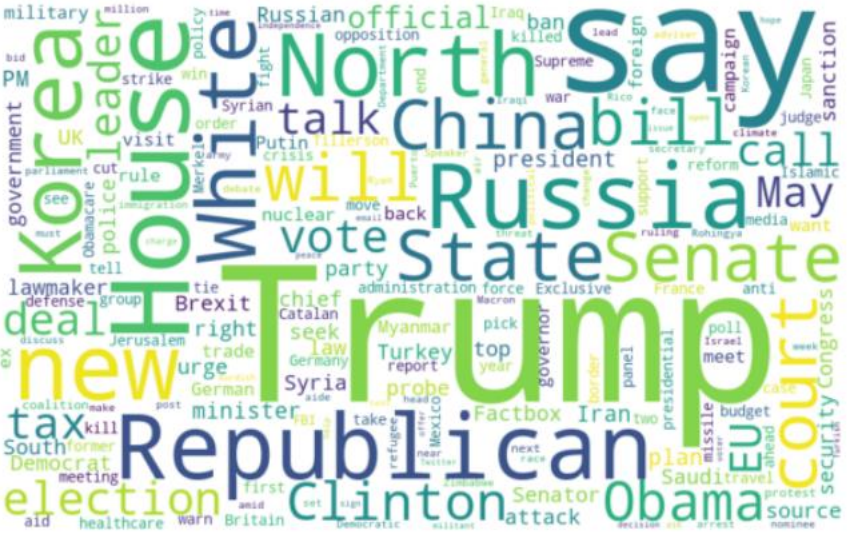


Figure 4.1. ISOT Dataset real news word-cloud.

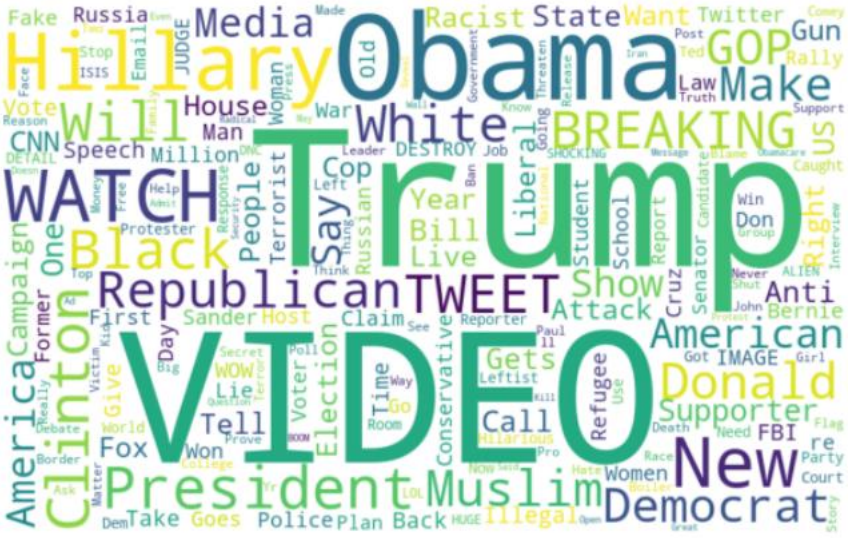


Figure 4.2. ISOT dataset fake news word-cloud.

4.4 Experimentation Steps

4.4.1 Data Preprocessing

The pre-processing approach discussed in chapter 3 is employed to preprocess the text of I dataset. This approach is based on existing studies conducted on fake news detection (Blackledge & Atapour-Abarghouei, 2021; Hakak et al., 2021; Jiang et al., 2021),). The subsequent section provides detailed steps we implement:

- **Removal of Metadata, Source Information and Attribution**: We remove metadata, source information, unnecessary editorial statements from the text, attribution statements that appeared at the beginning of the text, such as "MEXICO CITY (Reuters)". This step involves eliminating sentences or paragraphs that provided information about the source or accuracy of the data.
- **Removal of Time and Date Stamps**: We eliminate time stamps (example: "[1713 EST]") and date stamps (example: "December 31, 2017") present in the text, as they are not relevant to our analysis.
- **Removal of Links:** Given that our datasets comprise real-world news articles, we encounter numerous meaningless URLs that do not provide any relevant information. To address this issue, all kinds of links are removed. Here is an example of shortened URLs: "45.wh.gov/XpFsZu" and "bit.ly/2jBh4LU," that are eliminated.
- **Removal of Twitter Mentions**: We eliminate Twitter mentions (example: "@realDonaldTrump") from the text.
- **Removal of Named Entities**: We use the spaCy library to identify and remove named entities from the text. Specifically, we filter out tokens classified as "PERSON" entities to ensure the privacy and anonymity of individuals mentioned in the data.
- **Lowercasing and Removal of Non-Alphanumeric Characters**: We convert the remaining text to lowercase and removed non-alphanumeric characters, ensuring that the text was consistent and easier to process.

Figure 4.3 below illustrates the steps of preprocessing procedure:

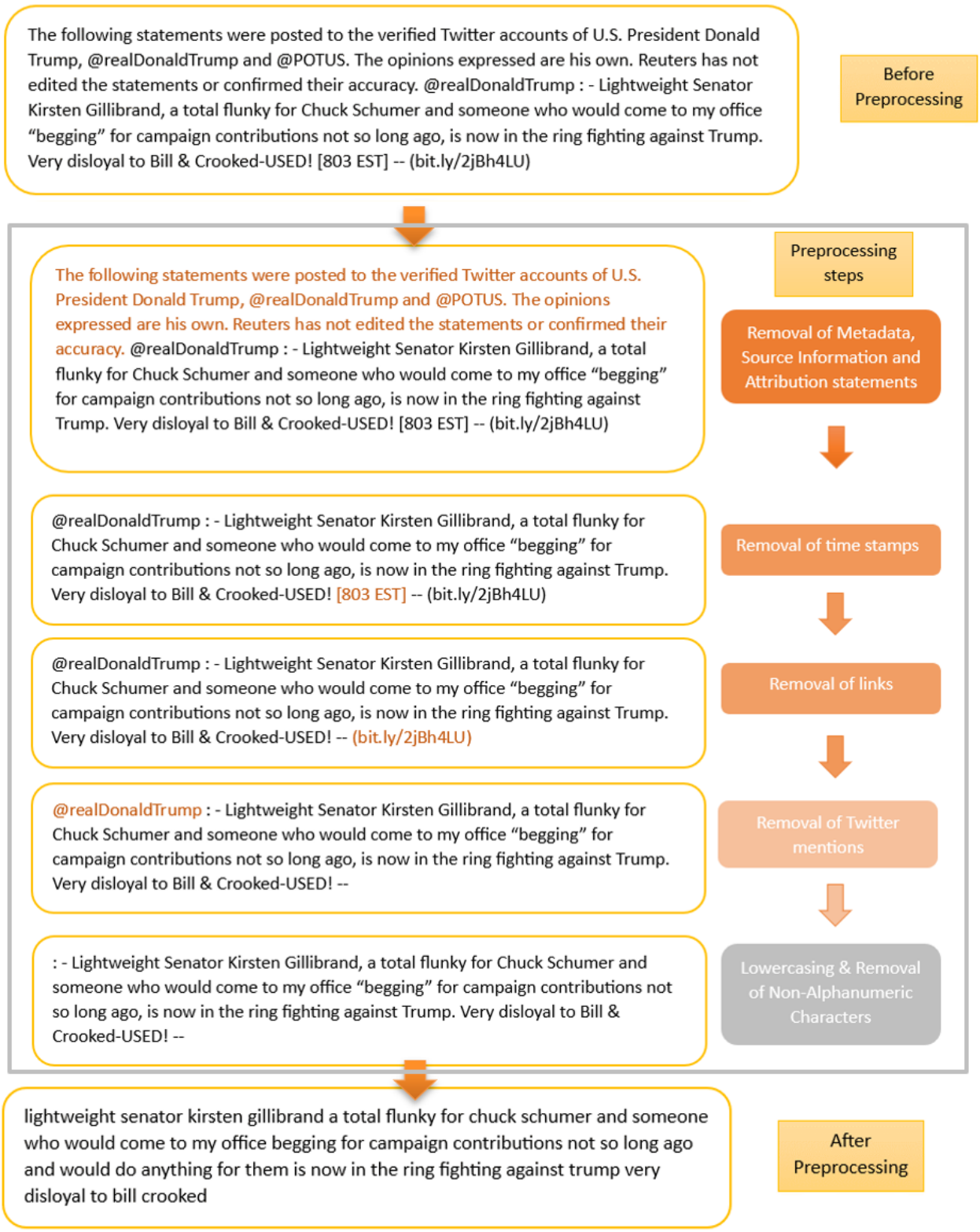


Figure 4.3. Preprocessing steps illustration.

4.4.2 Tokenization and Sentence Encoding

- **BERT tokenization and sentences encoding**

BERT tokenization and sentences encoding process is a crucial step in using a pre-trained BERT model. The following steps involved in this process are:

1. Initialization: The process begins by setting up the BERT tokenizer, derived from a pre-trained model named 'bert-base-uncased' in the transformers package developed by Hugging Face.
2. Tokenization Function: A function named 'tokenize_and_pad' is defined. It accepts a text and a maximum length as inputs. Within this function:
   - Encoding: The text is transformed using the BERT tokenizer's 'encode' method. This includes adding special tokens like [CLS] at the beginning and [SEP] at the end of each sentence.
   - Length Adjustment: If the encoded sequence exceeds the maximum length, it is shortened (truncated). On the other hand, if the sequence is shorter than the maximum length, it is extended by adding zeros (padding).
3. Application: The 'tokenize_and_pad' function is applied to the cleaned dataset. This application generates an array of input_ids for each sentence in the dataset.

The maximum length will be set for sentences to 512 tokens. This aligns with the maximum length limit of the 'bert-base-uncased' pre-trained model.

The Figure 4.4 provided illustrates the various steps involved in the Bert tokenization and sentence encoding process for the given pre-processed news article:

“despite thousands of hours wasted and many millions of dollars spent the democrats have been unable to show any collusion with russia so now they are moving on to the false accusations and fabricated stories of women who i don t know and or have never met fake news lightweight senator kirsten gillibrand a total flunky for chuck schumer and someone who would come to my office begging for campaign contributions not so long ago and would do anything for them is now in the ring fighting against trump very disloyal to bill crooked used wishing all of those celebrating hanukkah around the world a happy and healthy eight nights in the”

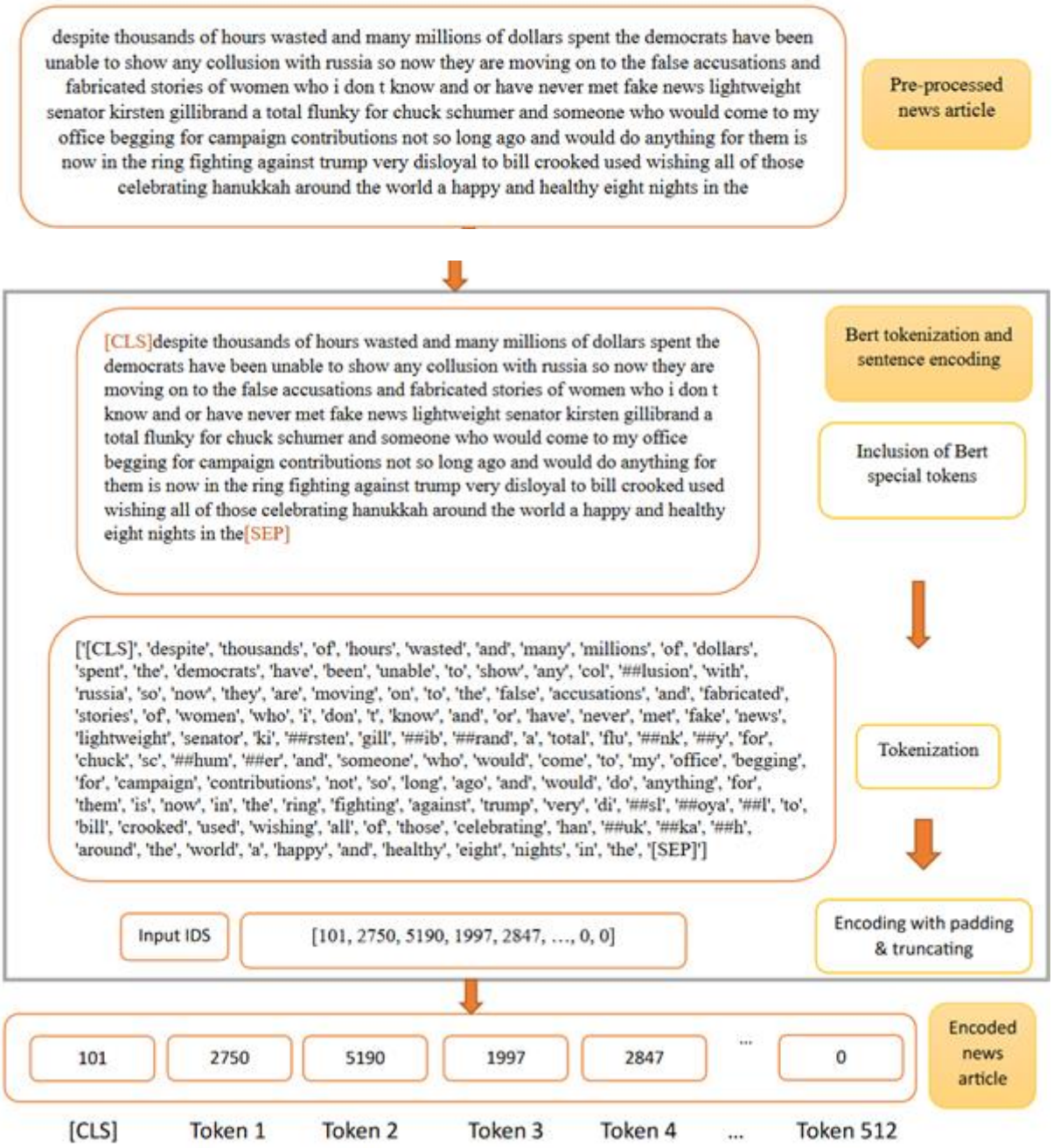


Figure 4.4. BERT tokenization and sentences encoding illustration.

- **Deep learning models tokenization and sentences encoding**

In the context of deep learning models, tokenization and sentence encoding play a vital role. The process involves several steps as outlined below:

1. Initialization: The first step is to set up the tokenization process. In this case, the Tokenizer class from the Keras library is utilized. An instance of this class is created to perform the tokenization and encoding.
2. Tokenization: The text data is tokenized using the fitted tokenizer. This involves breaking down the text document into individual words or tokens. The Tokenizer's fit_on_texts method is applied to the cleaned text data in the training set to build the vocabulary of the tokenizer. Each unique word is assigned an integer index.
3. Encoding: After fitting the tokenizer, the cleaned text data in the training, validation, and test sets are converted into sequences of integers. This is achieved using the texts_to_sequences method provided by the tokenizer. Each word in the text is replaced with its corresponding integer index from the tokenizer's vocabulary.
4. Padding: To ensure that all sequences have the same length, padding is performed. We set the max length of sentences to 1000 tokens (experimentally). The pad_sequences function from Keras is utilized to pad the sequences in the training, validation, and test sets to the specified maximum length.

The following figure show the code of tokenization and sentences encoding steps for DL models:

```
# Tokenize and pad sequences
tokenizer = Tokenizer() # create an instance of the Tokenizer class.
tokenizer.fit_on_texts(train_data['clean_text']) # fit a tokenizer on the cleaned text in the training set

# convert the cleaned text in the training set, validation set and test set to sequences of integers using the fitted tokenizer
X_train = tokenizer.texts_to_sequences(train_data['clean_text'])
X_val = tokenizer.texts_to_sequences(val_data['clean_text'])
X_test = tokenizer.texts_to_sequences(test_data['clean_text'])

# pad the sequences in the training set, validation set and test set to have a uniform length
max_length = 1000 # set the maximum length of a text sequence to 1000
X_train = pad_sequences(X_train, maxlen=max_length)
X_val = pad_sequences(X_val, maxlen=max_length)
X_test = pad_sequences(X_test, maxlen=max_length)

# Extract the labels
y_train = train_data['label'].values
y_val = val_data['label'].values
y_test = test_data['label'].values
```

Figure 4.5. Python code for tokenizing and encoding sentences in DL models.

4.4.3 Train, Validation and Test Sets Partitioning

The dataset initially consists of two CSV files containing fake and real news respectively. We merged the two datasets and then divided them into training, validation, and test sets. The splitting was done with a ratio of 70% for training, 10% for validation, and 20% for testing. The combined dataset in its original form consisted of 44,898 data entries. Table 4.2 displays the distribution of data across the training, validation, and test sets.

Table 4.2. Distribution of Data.

Training	Validation	Test
70%	10%	20%
31,428	4,489	8,981

4.4.4 Data preparation

After the tokenization and encoding process, the next step involves creating an embedding matrix. In our deep learning experiments, we used pretrained word embedding technique to obtain text representation. To reduce trainable parameters and increase time efficiency, we used a file of FastText word embeddings called ‘crawl-300d-2M.vec’ and loaded it in our model.

4.4.5 Models Building

- **CNN model**

Our CNN architecture as shown in Figure 4.6, is similar to the model proposed by (Jiang et al. 2021), incorporates various layers such as an embedding layer, a dropout layer, a convolutional layer, a global max pooling layer, and two dense layers. To mitigate over-fitting, we follow the approach suggested by Tan et al. (2019) and typically utilize one or two densely connected layers in our networks. The input is a 1000-token sequence represented by 300-dimensional vectors. A dropout layer with a 0.2 rate is used to prevent overfitting. The convolutional layer has 128 filters with a kernel size of 7. A global max pooling layer extracts important features, followed by another dropout layer. The output passes through two dense layers, with the final layer performing binary classification. The model has a total of 28,303,589 parameters, with 273,089 trainable and 28,030,500 non-trainable parameters.

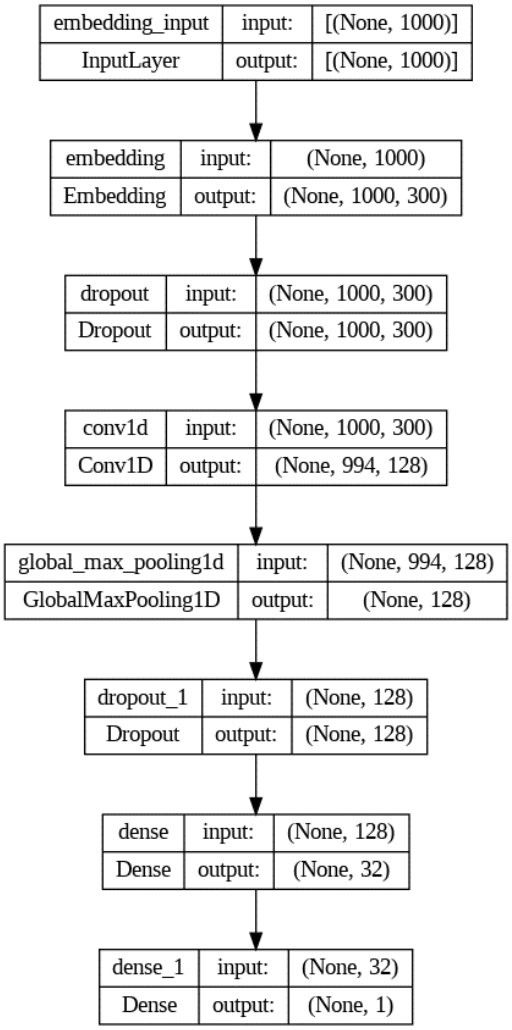


Figure 4.6. CNN architecture.

The following Figure 4.7 shows an example of python code for CNN model implementation:

```
# Define the CNN model
def create_model(learning_rate, dropout_rate, num_filters, kernel_size):
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], input_length=max_length, trainable=False))
    model.add(Dropout(dropout_rate))
    model.add(Conv1D(filters=num_filters, kernel_size=kernel_size, activation='relu'))
    model.add(GlobalMaxPooling1D())
    model.add(Dropout(dropout_rate))
    model.add(Dense(units=32, activation='relu'))
    model.add(Dense(units=1, activation='sigmoid'))
    optimizer = Adam(learning_rate=learning_rate)

    # Compile the model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Figure 4.7. Python code for CNN model building.

- **GRU model**

We employed the same layer configuration as (Jiang et al., 2021), consisting of an embedding layer, a GRU layer, a dropout layer, and a dense layer. Our implementation processed a 1000-token sequence represented by 300-dimensional vectors. The GRU layer output a 128-dimensional hidden state. To prevent overfitting, a dropout layer with a 0.3 rate was incorporated. The output passed through a dense layer with a single classification unit. Our model had a total of 28,195,749 parameters, with 165,249 trainable and 28,030,500 non-trainable parameters (Figure 4.8).

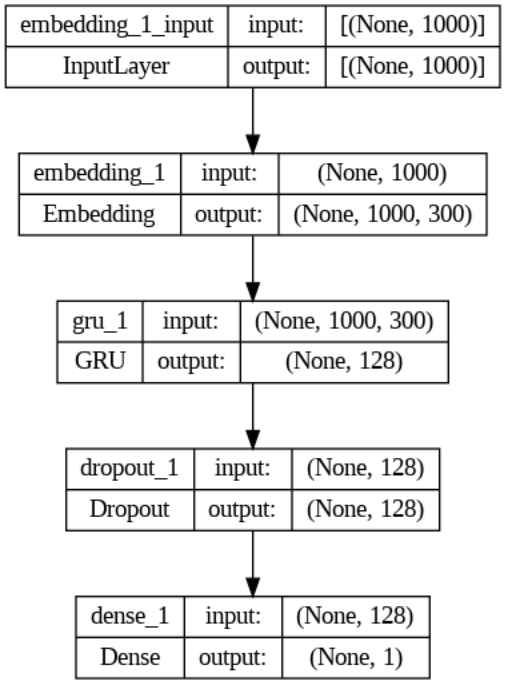


Figure 4.8. GRU architecture.

The python code for implementing the GRU model is illustrated in Figure 4.9.

```
# Define the GRU model
def create_model(learning_rate, dropout_rate, gru_units):
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], input_length=max_length, trainable=False))
    model.add(GRU(units=gru_units, dropout=dropout_rate))
    model.add(Dropout(dropout_rate))
    model.add(Dense(units=1, activation='sigmoid'))
    optimizer = Adam(learning_rate=learning_rate)

    # Compile the model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Figure 4.9. Python code for GRU model building.

- **Bidirectional GRU model**

We used the same layer configuration as GRU model, except using two GRU layers as showed in Figure 4.10. The Bidirectional Gru model has a total of 28,360,997 parameters, with 330,497 trainable and 28,030,500 non-trainable parameters.

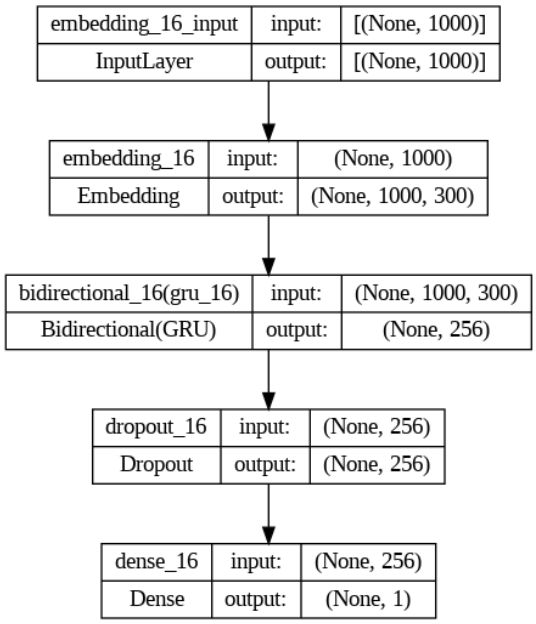


Figure 4.10. Bi-GRU architecture.

The python code for implementing the Bi-GRU model is illustrated in Figure 4.11:

```
# Define the Bi-GRU model
def create_model(learning_rate, dropout_rate, gru_units):
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], input_length=max_length, trainable=False))
    model.add(Bidirectional(GRU(units=gru_units, dropout=dropout_rate)))
    model.add(Dropout(dropout_rate))
    model.add(Dense(units=1, activation='sigmoid'))
    optimizer = Adam(learning_rate=learning_rate)

    # Compile the model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Figure 4.11. Python code for Bi-GRU model building.

- **LSTM model**

The LSTM (Long Short-Term Memory) model architecture implemented as shown in Figure 4.12 consists of multiple layers that are added sequentially. It starts with an embedding layer that maps input words to dense vector representations. This is followed by an LSTM layer to process sequential data and capture long-term dependencies. A dropout layer is added to prevent overfitting, followed by a global max pooling layer to extract the most important features. Finally, a dense layer with sigmoid activation performs binary classification. The LSTM model has a total of 28,250,277 parameters, with 219,777 trainable and 28,030,500 non-trainable parameters.

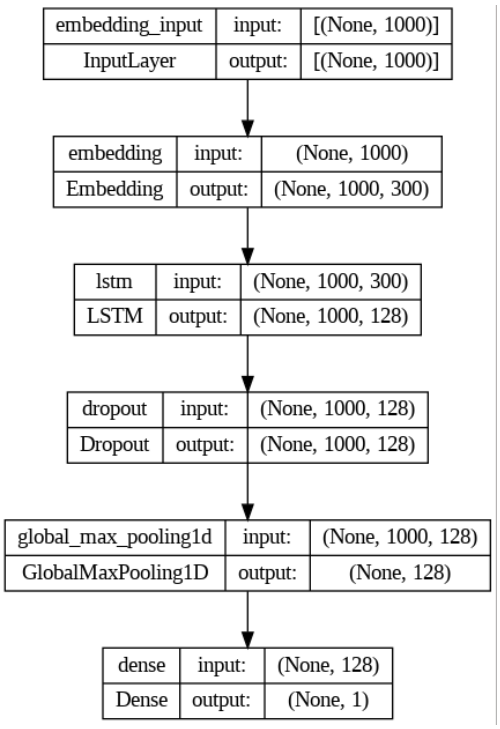


Figure 4.12. LSTM architecture.

The provided Figure 4.13 displays a sample of Python code used to implement LSTM model:

```
# Define the LSTM model
def create_model(learning_rate, dropout_rate, lstm_units):
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], input_length=max_length, trainable=False))
    model.add(LSTM(units=lstm_units, return_sequences=True))
    model.add(Dropout(dropout_rate))
    model.add(GlobalMaxPooling1D())
    model.add(Dense(units=1, activation='sigmoid'))
    optimizer = Adam(learning_rate=learning_rate)

    # Compile the model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Figure 4.13. Python code for LSTM model building.

- **Bidirectional LSTM model**

We employed an identical layer configuration as the LSTM model, with the exception of using two Bi-LSTM layers as shown in Figure 4.14.The model has a total of 28,872,293 parameters, with 841,793 trainable and 28,030,500 non-trainable parameters.

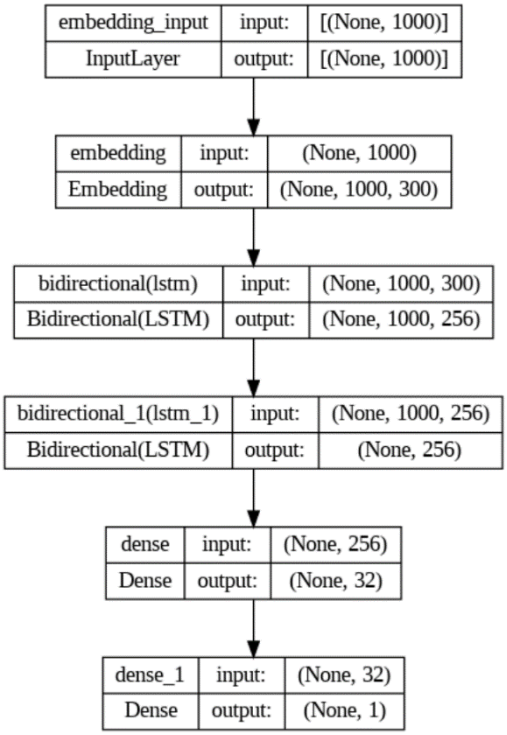


Figure 4.14. Bi-LSTM architecture.

The following Figure illustrates a sample of Python code utilized to implement a Bi-LSTM model:

```
# Define the Bi-LSTM model
def create_model(learning_rate, dropout_rate, lstm_units):
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, weights=[embedding_matrix], input_length=max_length, trainable=False))
    model.add(Bidirectional(LSTM(units=lstm_units, return_sequences=True)))
    model.add(Bidirectional(LSTM(units=lstm_units, return_sequences=False)))
    model.add(Dense(units=32, activation='relu'))
    model.add(Dense(units=1, activation='sigmoid'))
    optimizer = Adam(learning_rate=learning_rate)

    # Compile the model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Figure 4.15. Python code for Bi-LSTM model building.

- **BERT model**

BERT is a pre-trained language model that utilizes a transformer architecture. The model is built using the BertForSequenceClassification class from the transformers library. The architecture of the BERT model is based on the 'bert-base-uncased' variant, which consists of multiple transformer layers.

The following Figure 4.16 shows an example of python code for PyTorch implementation of the BERT model using the transformers Hugging Face library.

```
# Install the transformers library
!pip install transformers

import torch
from transformers import BertForSequenceClassification

# Initialize the BERT model for sequence classification
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=2)

# Set the device to GPU if available, otherwise use CPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = model.to(device)
```

Figure 4.16. Python code for BERT model building.

This piece of code sets up a BERT model to classify sequences, using the pre-trained 'bert-base-uncased' model. The num_labels variable is assigned a value of 2, indicating that it will be used for binary classification tasks such as distinguishing between fake news and real news. It also checks if a GPU is available and moves the model to the GPU if one is present; otherwise, it uses the CPU for processing. Before executing this code, it is necessary to have the transformers library installed.

4.4.6 Hyperparameter Tuning

To select the best hyper-parameters, the random search technique is employed. Hyper-parameters are predetermined values of machine learning models that directly influence their output during training. Although each model comes with default parameter values, these may not necessarily lead to optimal performance. Additionally, it is difficult to determine the best hyper-parameter values before training. Hence, multiple combinations are tested to find the most suitable values. Two popular techniques for this purpose are grid search and random search.

- **Random Search**

Random Search is a technique that selects hyperparameter combinations randomly instead of exhaustively exploring all possible permutations. It works well in both discrete and continuous spaces and can outperform Grid Search, especially when a small number of hyperparameters significantly impact the final output of the learning algorithm.

The hyperparameters examined for this work are presented in Table 4.3. We considered learning rate, dropout rate, number of units, number of filters and kernel size when optimizing hyperparameters.

Table 4.3. Hyperparameters values chosen for optimization.

Hyperparameter	Values
learning_rate	0.001, 0.01, 0.1
Number of units	128, 256, 512
dropout_rate	0.2, 0.3, 0.4
kernel_size	3, 5, 7
num_filters	32, 64, 128

- **Learning rate** is a crucial factor in optimization algorithms. It determines the size of the steps taken towards minimizing the loss function during each iteration. In simpler terms, it determines the extent to which the model should be adjusted based on the estimated error when updating the model weights. If the learning rate is too small, the training process takes a long time, whereas excessively large values can lead to suboptimal weights, rapid but unstable training (Saito & Rehmsmeier, 2015). In our work, we used binary_crossentropy as a loss function.

- **Dropout rate**: During the training phase, the dropout rate selects neurons to ignore. By excluding certain neurons and their associated weights, dropout aims to modify the weights of the neurons in a neural network. This approach encourages neighboring neurons to become more specialized, but it also introduces a risk of overfitting in the model.

- An **epoch**: represents a single iteration of training the model on the complete dataset. Therefore, the number of epochs indicates how many times the dataset is fed to the model during training. We conducted 3 training sessions for the BERT model using the training set. For deep learning classifiers we fixed an epoch value for 20.

- B**atch size:** indicates the quantity of input samples that are given to the model during training. The decision regarding the batch size depends on two factors: memory and time limitations, as well as the model's performance. A smaller batch size will result in longer training time, require less memory for training, and potentially produce more accurate predictions from the model. In our case, batch size is fixed to 128 for deep learning classifiers and 16 size for Bert model.

Table 4.4 presents the values of hyperparameters mentioned above:

Table 4.4. Hyperparameters for proposed models

Hyperparameters	DL models	TL model (BERT)
Max-length	1000	512
Loss Function	Binary-crossentropy	-
epoch	20	3
Batch-size	128	16
Optimizer	Adam	AdamW

The following Figure 4.17 shows Python code used for hyperparameter tuning and model optimization using random search method:

```
# Create a KerasClassifier with the model function and fixed training parameters
model = KerasClassifier(build_fn=create_model, epochs=20, batch_size=128, verbose=0)

# Set the number of iterations for random search
n_iter_search = 5

# Perform random search cross-validation
random_search = RandomizedSearchCV(estimator=model, param_distributions=param_dist, n_iter=n_iter_search, cv=3, n_jobs=1, verbose=2)

# Fit the random search on the training data and validate on the validation data
random_search.fit(X_train,
                  y_train,
                  validation_data=(X_val, y_val),
                  callbacks=[EarlyStopping(patience=3, restore_best_weights=True), ReduceLROnPlateau(patience=1)]
                  )
```

Figure 4.17. Python code for hyperparameter tuning using random search method.

The presented Python code snippet involved hyperparameter tuning using random search cross-validation, employing the Keras library and scikit-learn's RandomizedSearchCV class. KerasClassifier is instantiated with the create_model function, which defines the architecture of the deep learning model. The model is trained for 20 epochs with a batch size of 128. To explore a range of hyperparameter combinations, a random search with 5 iterations was performed, utilizing parallel processing for efficiency (n_jobs=-1). This randomized search cross-validation involved evaluating the model on 3-fold cross-validation and selecting the best hyperparameters based on performance. To prevent overfitting and enhance generalization, early stopping with a patience of 3 epochs and restoration of the best weights were implemented, along with a learning rate reduction using the ReduceLROnPlateau callback with a patience of 1. By conducting this optimization process, we aim to improve the model's performance and robustness for detection.

An example output of the above code snippet on GRU model is shown in Figure 4.18. During the model fitting process, cross-validation is performed with different combinations of hyperparameters. Each fit is timed depending on the specific hyperparameter combination being evaluated.

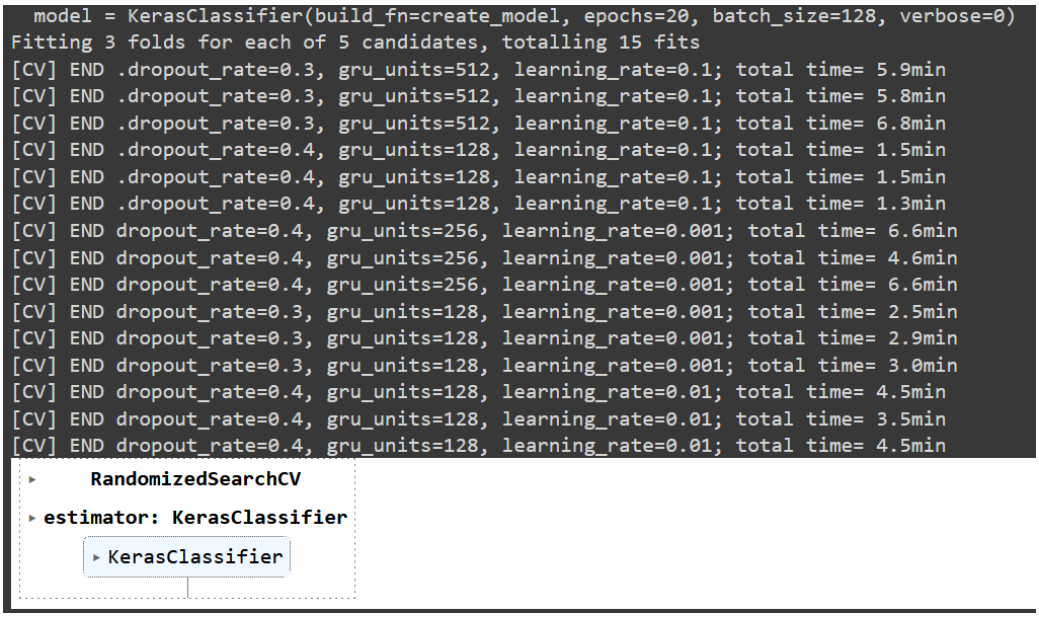


Figure 4.18. Results of GRU Model's Hyperparameter Tuning Process.

After applying the random search technique for the deep learning models covered in chapter 3, we obtain the best hyperparameters provided in the **Table 4.5** below:

Table 4.5. Optimal Hyperparameters from Random Search for Deep Learning Models.

Model / Parameter	GRU	Bi-GRU	LSTM	Bi- LSTM
learning_rate	0.01	0.01	0.01	0.01
Number of units	128	128	128	128
dropout_rate	0.3	0.4	0.2	0.1
kernel_size	-	-	-	-
num_filters	-	-	-	-

4.5 Results and Discussion

In this section of the report, we present the outcomes of our suggested method, which were assessed by determining the scores of performance metrics achieved by the suggested models on the test sets of ISOT dataset.

4.5.1 Training and Validation Results

The illustrations below represent plots of Training and validation loss and accuracy values against number of epochs for all the suggested models. These curves show the changes in loss and accuracy as the training progresses, observed on both the training and validation sets. They allow us to assess the effectiveness of the learning process by determining whether the models have excessively adapted to the training set, indicating overfitting.

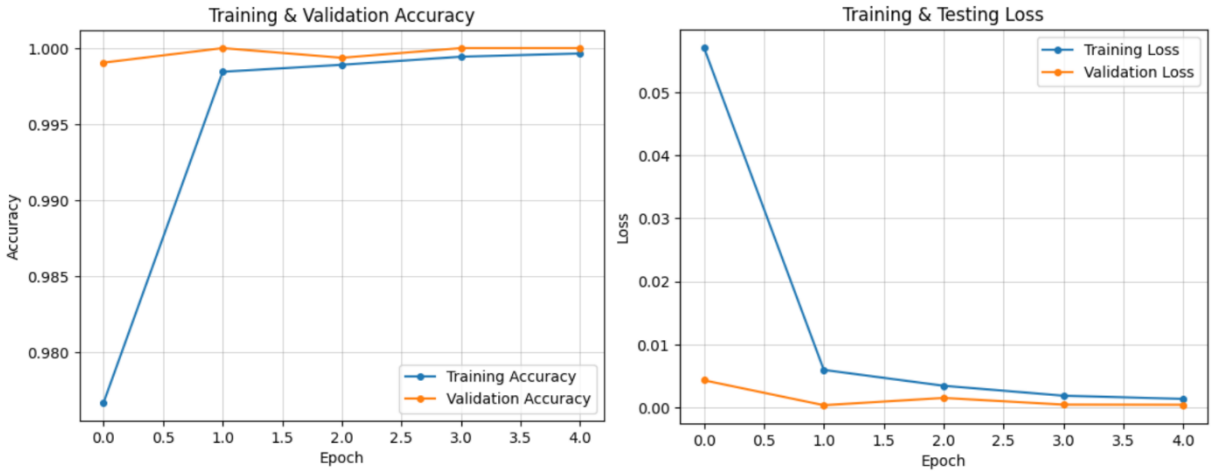


Figure 4.19. Training and validation accuracy and loss graphs of GRU model.

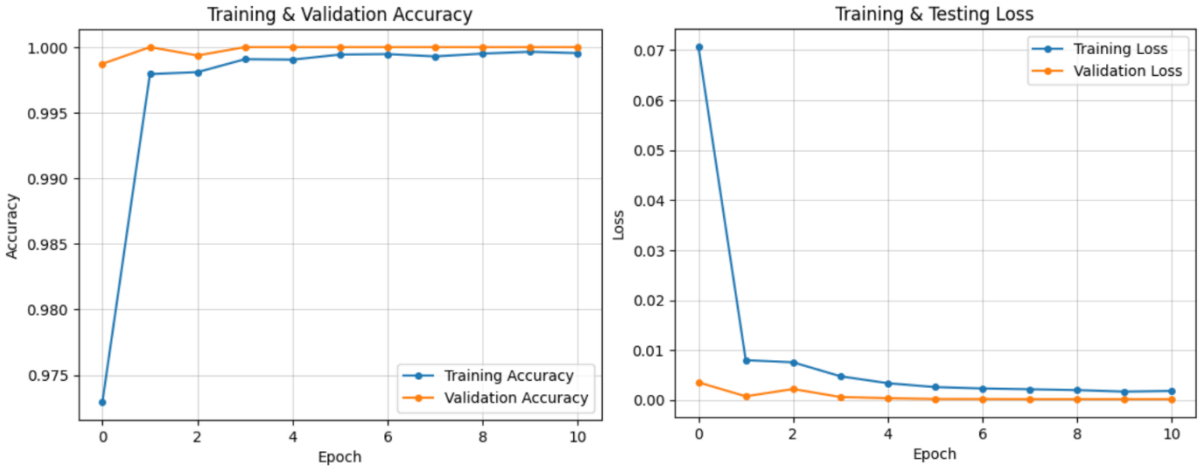


Figure 4.20. Training and validation accuracy and loss graphs of Bi-GRU model.

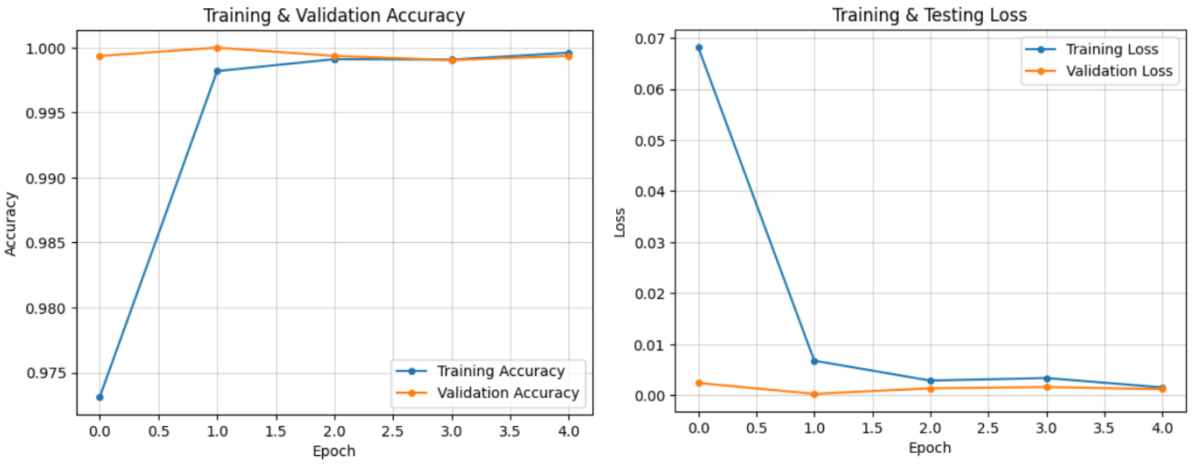


Figure 4.21. Training and validation accuracy and loss representations of CNN model.

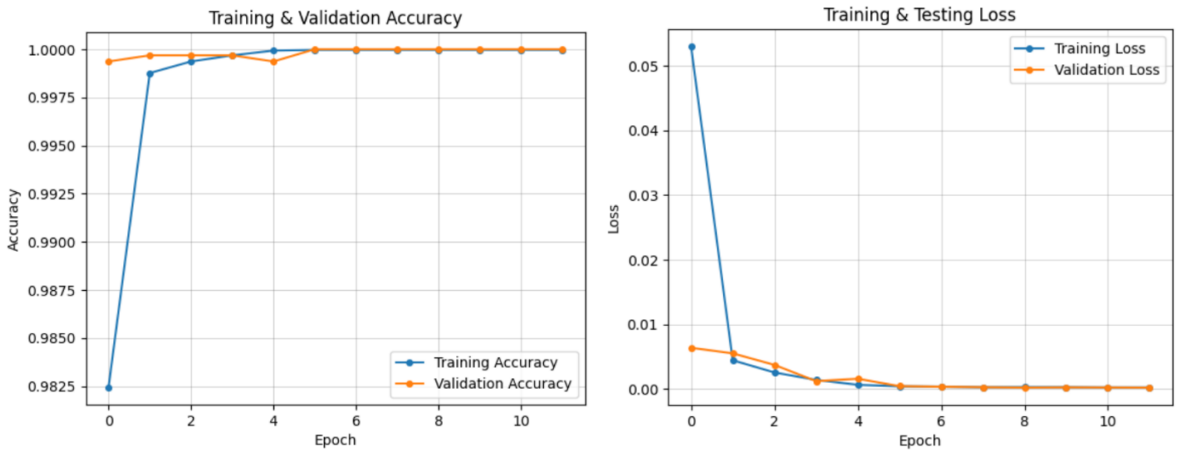


Figure 4.22. Training and validation accuracy and loss representations of LSTM model.

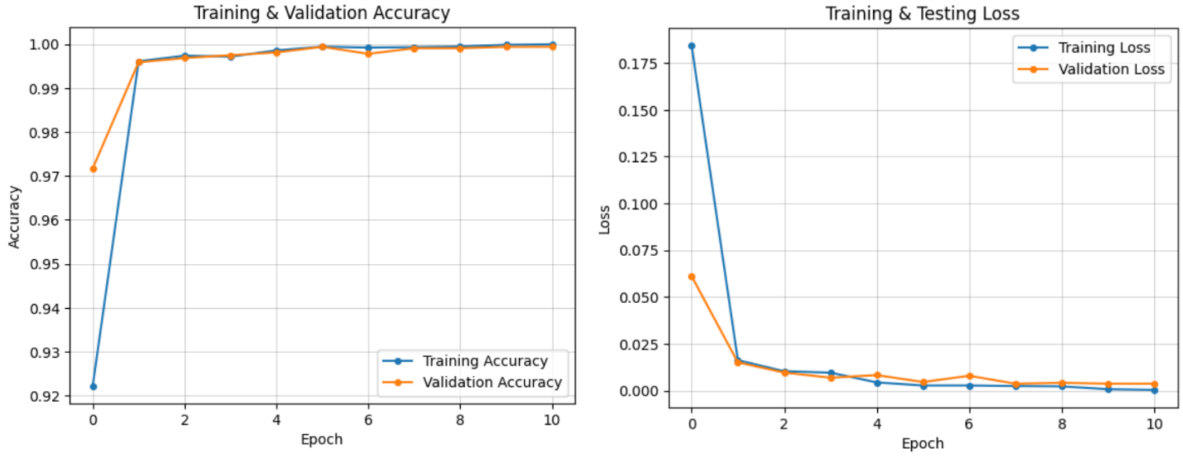


Figure 4.23. Training and validation accuracy and loss representations of Bi-LSTM model.

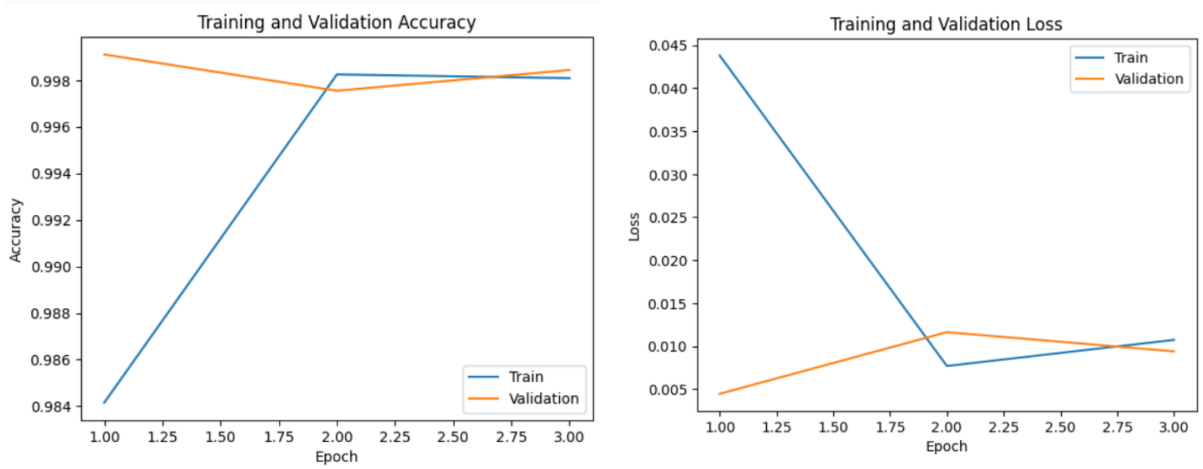


Figure 4.24. Training and validation accuracy and loss representations of BERT model.

4.5.2 Performance Evaluation

The Table 4.6 presents the results of suggested models applied to the ISOT Fake News dataset, showcasing the evaluation metrics.

Table 4.6. The performance of suggested models.

Model	Accuracy	Precision	Recall	F1-Score
CNN	99.91%	99.92%	99.89%	99.90%
LSTM	99.92%	99.96%	99.87%	99.92%
Bi-LSTM	99.88%	99.95%	99.79%	99.87%
GRU	99.88%	99.96%	99.78%	99.87%
Bi-GRU	99.91%	99.98%	99.82%	99.90%
BERT	99.86%	99.78%	99.92%	99.85%

Among the models, both LSTM and Bi-GRU achieve the highest accuracy of 99.92%, closely followed by CNN and Bi-LSTM at 99.91% and 99.88%, respectively. GRU and BERT achieve slightly lower accuracy scores at 99.88% and 99.86%, respectively.

In terms of precision, LSTM and Bi-GRU exhibit the highest values at 99.96% and 99.98%, respectively, implying a low rate of false positives. CNN, GRU, and BERT display similarly high precision scores above 99.78%. Bi-LSTM achieves a slightly lower precision at 99.95%, but still demonstrates excellent performance in this regard.

Regarding recall, LSTM obtains the highest score of 99.87%, followed closely by Bi-LSTM and Bi-GRU at 99.79% and 99.82%, respectively. CNN and GRU achieve recall values of 99.89% and 99.78%, while BERT exhibits the highest recall among the models at 99.92%.

These models consistently demonstrated high Precision, Recall, and F1-scores, highlighting their ability to effectively differentiate between fake news and authentic news articles.

4.5.3 Comparison of proposed approach with existing works

This part is focused on evaluating our models by comparing them to previous studies in the literature that utilized the ISOT dataset for both training and testing purposes. The approaches we compare against are as follows: (Nadeem et al., 2023) introduced a combination of BERT and hypercontext to extract features, (Sastrawan et al., 2022) proposed various architectures such as CNN, Bidirectional LSTM, and ResNet, and (Hakak et al., 2021) presented an ensemble machine learning with feature extraction. Table 4.7 presents the comparative analysis of the present work with existing works.

Table 4.7. Comparison of proposed approach with previous contributions' results.

Reference	Algorithm	Accuracy
(Hakak et al., 2021)	Decision tree	100%
	Random forest	100%
	Extra tree	100%
(Sastrawan et al., 2022)	CNN	99.88%
	Bi- LSTM	99.95%
	ResNet	99.90%
(Nadeem et al., 2023)	HyproBert hybrid model	99.30%
Proposed	CNN	99.91%
	LSTM	**99.92%**
	Bi-LSTM	99.88%
	GRU	99.88%
	Bi-GRU	99.91%
	BERT	99.86%

Discussion

The proposed models demonstrate superior accuracy compared to previous work referenced in the literature.

- Our suggested models are almost equal in accuracy compared to the contribution of (Hakak et al., 2021). However, our approach and theirs use different techniques. (Hakak et al., 2021) used three machine learning classifiers, namely DT, RF, and ET for extracting various features.
- The models proposed by (Sastrawan et al., 2022), such as CNN, Bi-LSTM, and ResNet achieved accuracy rates of 99.88%, 99.95%, and 99.90%, respectively, which were slightly lower than our proposed models' accuracies, except for Bi-LSTM model.
- While (Nadeem et al., 2023) reported an accuracy of 99.3%, our proposed models outperformed it with higher accuracies.

4.6 Conclusion

This chapter discusses the outcomes of our methodology used to address the issue of fake news. Additionally, we discovered that combining various high-performing classifiers, each trained separately, can improve the predictive capabilities of the transfer learning classifier that we developed. Ultimately, achieving better automated fake news detection relies on the availability of high-quality training data and pre-trained models with excellent performance.

General Conclusion

In this report, the focus was on utilizing deep learning and transfer learning models to identify fake news. The aim was to improve the automated detection capabilities and performance of these models to filter out false and misleading information. This would ultimately enhance the quality of content in online environments.

The proposed methodology employed a data-driven supervised learning approach. It utilized deep learning algorithms and the pre-trained BERT BASE model (Devlin et al., 2018) to create an effective system for detecting fake news. By exploring deep learning techniques, the model could extract complex patterns and features from text data, enabling accurate identification and classification of fake news.

The proposed models trained and evaluated on the ISOT dataset, including CNN, LSTM, Bi-LSTM, GRU, Bi-GRU, and BERT, demonstrated high accuracy in detecting fake news. The results achieved were as follows: Proposed CNN (99.91%), Proposed LSTM (99.92%), Proposed Bi-LSTM (99.88%), Proposed GRU (99.88%), Proposed Bi-GRU (99.91%), and Proposed BERT (99.86%). These impressive values indicate the effectiveness of deep learning and transfer learning models in fake news detection tasks.

In the future, the aim is to extend the scope of this work by incorporating additional datasets and exploring combinations of ensemble models, which may involve machine learning models. Additionally, the intention is to investigate the application of different word embeddings for the deep learning models. While the current work primarily focuses on binary classification, there is a desire to further examine the effectiveness of these models in addressing multi-class classification problems.

References

Abdali, S., Bastidas, G. G., Shah, N., Papalexakis, E. E., Abdali, S., Bastidas, G. G., Papalexakis, · E E, & Shah, N. (2020). *Tensor Embeddings for Content-Based Misinformation Detection with Limited Supervision*. 117–140. https://doi.org/10.1007/978-3-030-42699-6_7

Ahmed, A. A. A., Aljabouh, A., Donepudi, P. K., & Choi, M. S. (2021). *Detecting Fake News Using Machine Learning : A Systematic Literature Review*. https://arxiv.org/abs/2102.04458v1

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10618 LNCS*, 127–138. https://doi.org/10.1007/978-3-319-69155-8_9

Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*(1), e9. https://doi.org/10.1002/SPY2.9

Ahmed, S., Hinkelmann, K., & Corradini, F. (2022). Combining Machine Learning with Knowledge Engineering to detect Fake News in Social Networks-a survey. *CEUR Workshop Proceedings*, *2350*. https://arxiv.org/abs/2201.08032v1

Ajao, O., Bhowmik, D., & Zargari, S. (2018). Fake news identification on Twitter with hybrid CNN and RNN models. *ACM International Conference Proceeding Series*, 226–230. https://doi.org/10.1145/3217804.3217917

Akbik, A., Blythe, D., & Vollgraf, R. (2018). *Contextual String Embeddings for Sequence Labeling* (pp. 1638–1649). https://aclanthology.org/C18-1139

Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, *31*(2), 211–236. https://doi.org/10.1257/JEP.31.2.211

Aswani, R., Kar, A. K., & Vigneswara Ilavarasan, P. (2018). Detection of Spammers in Twitter marketing: A Hybrid Approach Using Social Media Analytics and Bio Inspired Computing. *Information Systems Frontiers*, *20*(3), 515–530. https://doi.org/10.1007/S10796-017-9805-8/METRICS

Azad, R., Mohammed, B., Mahmud, R., Zrar, L., & Sdiq, S. (2021). Fake News Detection in low-resourced languages “Kurdish language” using Machine learning algorithms. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(6), 4219–4225. https://turcomat.org/index.php/turkbilmat/article/view/8393

Bajaj, S. (2017). *“ The Pope Has a New Baby ! ” Fake News Detection Using Deep Learning*.

Blackledge, C., & Atapour-Abarghouei, A. (2021). Transforming Fake News: Robust Generalisable News Classification Using Transformers. *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, 3960–3968. https://doi.org/10.1109/BigData52589.2021.9671970

Bojanowski, P., Grave, É., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/TACL_A_00051

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). Mean birds: Detecting aggression and bullying on Twitter. *WebSci 2017 - Proceedings of the 2017 ACM Web Science Conference*, 13–22. https://doi.org/10.1145/3091478.3091487

Chen, W., Zhang, Y., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Unsupervised rumor detection based on users behaviors using neural networks. *Pattern Recognition Letters*, *105*, 226–233. https://doi.org/10.1016/J.PATREC.2017.10.014

Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). Misleading online content: Recognizing clickbait as "false news." *WMDD 2015 - Proceedings of the ACM Workshop on Multimodal Deception Detection, Co-Located with ICMI 2015*, 15–19. https://doi.org/10.1145/2823465.2823467

CHEQ. (2019). *The economic cost of bad actors on the internet.*

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Computational Fact Checking from Knowledge Networks. *PLOS ONE*, *10*(6), e0128193. https://doi.org/10.1371/JOURNAL.PONE.0128193

Cohen, S., Hamilton, J. T., & Turner, F. (2011). Computational journalism. *Communications of the ACM*, *54*(10), 66–71. https://doi.org/10.1145/2001269.2001288

Coleman, A. (2020). COVID-19-Related infodemic and its impact on public health: A global social media analysis. *American Journal of Tropical Medicine and Hygiene*, *103*(4), 1621–1629. https://doi.org/10.4269/AJTMH.20-0812

Collins, B., Hoang, T., Nguyen, N. T., & Hwang, D. (2020). *Journal of Information and Telecommunication ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20 Trends in combating fake news on social media-a survey*. https://doi.org/10.1080/24751839.2020.1847379

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, *12*, 2493–2537. https://arxiv.org/abs/1103.0398v1

Copenhaver, K. (2018). Fake news and digital literacy: The academic library's role in shaping digital citizenship. *Https://Doi.Org/10.1080/02763877.2018.1484412*, *59*(3), 107. https://doi.org/10.1080/02763877.2018.1484412

Dai, E., Sun, Y., & Wang, S. (2020). Ginger Cannot Cure Cancer: Battling Fake Health News with a Comprehensive Data Repository. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*, 853–862. https://doi.org/10.1609/ICWSM.V14I1.7350

De Oliveira, N. R., Medeiros, D. S. V., & Mattos, D. M. F. (2020). A Syntactic-Relationship Approach to Construct Well-Informative Knowledge Graphs Representation. *2020 4th Conference on Cloud and Internet of Things, CIoT 2020*, 75–82. https://doi.org/10.1109/CIOT50422.2020.9244288

de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., & Mattos, D. M. F. (2021). Identifying Fake News on Social Networks Based on Natural Language Processing: Trends and Challenges. *Information 2021, Vol. 12, Page 38*, *12*(1), 38. https://doi.org/10.3390/INFO12010038

Del Vicario, M., Quattrociocchi, W., Scala, A., & Zollo, F. (2018). Polarization and Fake News: Early Warning of Potential Misinformation Targets. *ACM Transactions on the Web*, *13*(2). https://doi.org/10.1145/3316809

Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G. W. S., & Zubiaga, A. (2017). SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 69–76. https://arxiv.org/abs/1704.05972v1

Dessí, D., Dragoni, M., Fenu, G., Marras, M., & Reforgiato Recupero, D. (2020). *Deep Learning Adaptation with Word Embeddings for Sentiment Analysis on Online Course Reviews*. 57–83. https://doi.org/10.1007/978-981-15-1216-2_3

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://arxiv.org/abs/1810.04805v2

Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., & Zhang, W. (2014). *Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion*. https://research.google/pubs/pub45634/

Fazil, M., & Abulaish, M. (2018). A Hybrid Approach for Detecting Automated Spammers in Twitter. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2707–2719. https://doi.org/10.1109/TIFS.2018.2825958

Fernandez, P. (2017). The technology behind fake news. *Library Hi Tech News*, *34*(7), 1–5. https://doi.org/10.1108/LHTN-07-2017-0054/FULL/PDF

Ferrara, E. (2017). Disinformation and Social Bot Operations in the Run Up to the 2017 French Presidential Election. *First Monday*, *22*(8). https://doi.org/10.5210/fm.v22i8.8005

Ferrara, E. (2020). What types of COVID-19 conspiracies are populated by Twitter bots? *First Monday*. https://doi.org/10.5210/FM.V25I6.10633

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, *59*(7), 96–104. https://doi.org/10.1145/2818717

Ferreira, W., & Vlachos, A. (2016). Emergent: a novel data-set for stance classification. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 1163–1168. https://doi.org/10.18653/V1/N16-1138

Fetzer, J. H. (2004). Disinformation: The Use of False Information. *Minds and Machines*, *14*(2), 231–240. https://doi.org/10.1023/B:MIND.0000021683.28604.5B/METRICS

Friggeri, A., Adamic, L. A., Eckles, D., & Cheng, J. (2014). Rumor Cascades. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 101–110. https://doi.org/10.1609/ICWSM.V8I1.14559

Gao, W., MITRA Sejeong KWON Bernard JANSEN, P. J., Author Jing, A. M., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., Ma, J., & Cha, M. (2016). *Detecting rumors from microblogs with recurrent neural networks*. 3818–3824. https://ink.library.smu.edu.sg/sis_research/4630/

Gelfert, A. (2018). Fake news: A definition. *Informal Logic*, *38*(1), 84–117. https://doi.org/10.22329/IL.V38I1.5068

Ghosh, S., & Shah, C. (2019). Toward Automatic Fake News Classification. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2019-January*, 2254–2263. https://doi.org/10.24251/HICSS.2019.273

Glenski, M., Volkova, S., & Kumar, S. (2020). *User Engagement with Digital Deception*. 39–61. https://doi.org/10.1007/978-3-030-42699-6_3

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5–6), 602–610. https://doi.org/10.1016/J.NEUNET.2005.06.042

Guo, X., & Yu, H. (2022). *On the Domain Adaptation and Generalization of Pretrained Language Models: A Survey*. https://arxiv.org/abs/2211.03154v1

Gupta, M., Zhao, P., & Han, J. (2012). Evaluating event credibility on twitter. *Proceedings*, 153–164. https://doi.org/10.1137/1.9781611972825.14

Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., & Khan, W. Z. (2021). An ensemble machine learning approach through effective feature extraction to classify fake news. *Future Generation Computer Systems*, *117*, 47–58. https://doi.org/10.1016/J.FUTURE.2020.11.022

Hangloo, S., & Arora, B. (2021). *Fake News Detection Tools and Methods -- A Review*. https://arxiv.org/abs/2112.11185v1

Hangloo, S., & Arora, B. (2022). Combating multimodal fake news on social media: methods, datasets, and future perspective. *Multimedia Systems 2022 28:6*, *28*(6), 2391–2422. https://doi.org/10.1007/S00530-022-00966-Y

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266. https://doi.org/10.1126/SCIENCE.AAA8685

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735

Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 328–339. https://doi.org/10.18653/v1/p18-1031

Howard, P. N., & Kollanyi, B. (2016). Bots, #Strongerin, and #Brexit: Computational Propaganda During the UK-EU Referendum. *SSRN Electronic Journal*. https://doi.org/10.2139/SSRN.2798311

Hunt, D., Robertson, D., & Pow, A. (2018). The Counselor's Role in the Age of Social Media and Fake News. *Journal of Creativity in Mental Health*, *13*(4), 405–417. https://doi.org/10.1080/15401383.2018.1462748

Jadhav, S. S., & Thepade, S. D. (2019). Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier. *Https://Doi.Org/10.1080/08839514.2019.1661579*, *33*(12), 1058–1068. https://doi.org/10.1080/08839514.2019.1661579

Jain, A., Shakya, A., Khatter, H., & Gupta, A. K. (2019). A smart System for Fake News Detection Using Machine Learning. *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. https://doi.org/10.1109/ICICT46931.2019.8977659

Jiang, T., Li, J. P., Haq, A. U., Saboor, A., & Ali, A. (2021). A Novel Stacking Approach for Accurate Detection of Fake News. *IEEE Access*, *9*, 22626–22639. https://doi.org/10.1109/ACCESS.2021.3056079

Jwa, H., Oh, D., Park, K., Kang, J. M., & Lim, H. (2019). exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT). *Applied Sciences 2019, Vol. 9, Page 4062*, *9*(19), 4062. https://doi.org/10.3390/APP9194062

Kaliyar, R. K., Goswami, A., & Narang, P. (2019). Multiclass Fake News Detection using Ensemble Machine Learning. *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 103–107. https://doi.org/10.1109/IACC48062.2019.8971579

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications*, *80*(8), 11765–11788. https://doi.org/10.1007/S11042-020-10183-2/TABLES/22

Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2020). FNDNet – A deep convolutional neural network for fake news detection. *Cognitive Systems Research*, *61*, 32–44. https://doi.org/10.1016/j.cogsys.2019.12.005

Knuutila, A., Neudert, L.-M., & Howard, P. N. (2022). Who is afraid of fake news? Modeling risk perceptions of misinformation in 142 countries. *Harvard Kennedy School Misinformation Review*. https://doi.org/10.37016/MR-2020-97

Kshetri, N., & Voas, J. (2017). *The Economics of "Fake News"* .

Kucharski, A. (2016). Study epidemiology of fake news. *Nature 2016 540:7634*, *540*(7634), 525–525. https://doi.org/10.1038/540525a

Kumar, S., Asthana, R., Upadhyay, S., Upreti, N., & Akbar, M. (2020). Fake news detection using deep learning models: A novel approach. *Transactions on Emerging Telecommunications Technologies*, *31*(2). https://doi.org/10.1002/ETT.3767

Kumar Singh Manish, M., Hamdard Jawed Ahmed Jawed Jamia Hamdard Mohammad Afshar Alam Alam Jamia Hamdard Kamlesh Kumar Raghuvanshi Kamlesh, J., Kumar Sachin, S., Kumar Singh, M., Ahmed, J., Afshar Alam, M., Kumar Raghuvanshi, K., & Kumar, S. (2022). *A Comparative Study of Computational Fake News Detection on Social Media*. https://doi.org/10.21203/RS.3.RS-1964791/V1

Larzillière, P., Al-Badayneh, Prof. D. M., Hasan, A., Ayasrah, A., Al-Madi, B., Keen, D., Yadamsuren, B., Erdelez, S., Wang, C., Huang, H., Guilbeault, D., Woolley, S., Becker, J., Hanakova, A., Potmesil, M., Stejskalova, K., Gelfert, A., Bastick, Z., Allcott, H., … Albright, J. (2018). Journalism, fake news & disinformation :: handbook for journalism education and training /: Cherilyn Ireton and Julie Posetti. *Undefined*, *31*(1), 211–236. https://digitallibrary.un.org/record/1641987

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). *Handwritten Digit Recognition with a Back-Propagation Network*.

Lee Howell. (2013). *Global Risks 2013*.

Liu, Y., & Wu, Y. F. B. (2018). Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 354–361. https://doi.org/10.1609/AAAI.V32I1.11268

Mavridis, G. (2018). *Fake news and Social Media: How Greek users identify and curb misinformation online*. http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-23196

Mazari, A. C., Aliane, H., & Alimazighi, Z. (2013). A conceptual indexing approach for Arabic texts. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*. https://doi.org/10.1109/AICCSA.2013.6616420

Mazari, A. C., Boudoukhani, N., & Djeffal, A. (2023). BERT-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*. https://doi.org/10.1007/S10586-022-03956-X

Mazari, A. C., & Djeffal, A. (2022). Hybrid Query Expansion Model Based on Pseudo Relevance Feedback and Semantic Tree for Arabic IR. *Int. J. Inf. Retr. Res.*, *12*(1), 1–16. https://doi.org/10.4018/IJIRR.289949

Mazari, A. C., & Djeffal, A. (2021). A combined method of naïve-bayes and pooling strategy for building test collection for Arabic/English information retrieval. *International Journal of Computing and Digital Systems*, *10*(1), 629–638. https://doi.org/10.12785/IJCDS/100161

Mazari, A. C., & Djeffal, A. (2022). Sentiment Analysis of Algerian Dialect Using Machine Learning and Deep Learning with Word2vec. *Informatica*, *46*(6), 67–78. https://doi.org/10.31449/INF.V46I6.3340

Mazari, A. C., & Kheddar, H. (2023). Deep Learning-based Analysis of Algerian Dialect Dataset Targeted Hate Speech, Offensive Language and Cyberbullying. *International Journal of Computing and Digital Systems*, *13*(1), 965–972. https://doi.org/10.12785/IJCDS/130177

Meel, P., & Vishwakarma, D. K. (2020). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, *153*, 112986. https://doi.org/10.1016/J.ESWA.2019.112986

Meneses Silva, C. V., Silva Fontes, R., & Colaço Júnior, M. (2020). Intelligent Fake News Detection: A Systematic Mapping. *Journal of Applied Security Research*, *16*(2), 168–189. https://doi.org/10.1080/19361610.2020.1761224

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. https://arxiv.org/abs/1301.3781v3

Min, B., Ross, H., Sulem, E., Veyseh, A. P. Ben, Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., & Roth, D. (2021). *Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey*. https://arxiv.org/abs/2111.01243v1

Modha, S., & Majumder, P. (2019). *An Empirical Evaluation of Text Representation Schemes on Multilingual Social Web to Filter the Textual Aggression*. https://arxiv.org/abs/1904.08770v1

Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M. M., Ai, F., & Lugano, U. (2019). *Fake News Detection on Social Media using Geometric Deep Learning*. https://arxiv.org/abs/1902.06673v1

Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., & Rahman, M. S. (2021). A Comprehensive Review on Fake News Detection with Deep Learning. *IEEE Access*, *9*, 156151–156170. https://doi.org/10.1109/ACCESS.2021.3129329

Nadeem, M. I., Mohsan, S. A. H., Ahmed, K., Li, D., Zheng, Z., Shafiq, M., Karim, F. K., & Mostafa, S. M. (2023). HyproBert: A Fake News Detection Model Based on Deep Hypercontext. *Symmetry 2023, Vol. 15, Page 296*, *15*(2), 296. https://doi.org/10.3390/SYM15020296

Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2015). A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, *104*(1), 11–33. https://doi.org/10.1109/JPROC.2015.2483592

Orabi, M., Mouheb, D., Al Aghbari, Z., & Kamel, I. (2020). Detection of Bots in Social Media: A Systematic Review. *Information Processing & Management*, *57*(4), 102250. https://doi.org/10.1016/J.IPM.2020.102250

Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A Survey on Natural Language Processing for Fake News Detection. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 6086–6093. https://arxiv.org/abs/1811.00770v2

Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and Its Applications*, *540*, 123174. https://doi.org/10.1016/J.PHYSA.2019.123174

Paka, W. S., Bansal, R., Kaushik, A., Sengupta, S., & Chakraborty, T. (2021). Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection. *Applied Soft Computing*, *107*, 107393. https://doi.org/10.1016/J.ASOC.2021.107393

Pathak, A. R., Agarwal, B., Pandey, M., & Rautaray, S. (2020). *Application of Deep Learning Approaches for Sentiment Analysis*. 1–31. https://doi.org/10.1007/978-981-15-1216-2_1

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/V1/D14-1162

Pierri, F., & Ceri, S. (2019). False News On Social Media. *ACM SIGMOD Record*, *48*(2), 18–32. https://doi.org/10.1145/3377330.3377334

Pilkevych, I., Fedorchuk, D., Naumchak, O., & Romanchuk, M. (2021). Fake News Detection in the Framework of Decision-Making System through Graph Neural Network. *2021 IEEE 4th International Conference on Advanced Information and Communication Technologies, AICT 2021 - Proceedings*, 153–157. https://doi.org/10.1109/AICT52120.2021.9628907

Preston, S., Anderson, A., Robertson, D. J., Shephard, M. P., & Huhe, N. (2021). Detecting fake news on Facebook: The role of emotional intelligence. *PLOS ONE*, *16*(3), e0246757. https://doi.org/10.1371/JOURNAL.PONE.0246757

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*.

Rajalaxmi, R. R., Narasimha Prasad, L. V., Janakiramaiah, B., Pavankumar, C. S., Neelima, N., & Sathishkumar, V. E. (2021). Optimizing Hyperparameters and Performance Analysis of LSTM Model in Detecting Fake News on Social media. *Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3511897

Rasool, T., Butt, W. H., Shaukat, A., & Akram, M. U. (2019). Multi-label fake news detection using multi-layered supervised learning. *ACM International Conference Proceeding Series*, 73–77. https://doi.org/10.1145/3313991.3314008

Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised Learning for Fake News Detection. *IEEE Intelligent Systems*, *34*(2), 76–81. https://doi.org/10.1109/MIS.2019.2899143

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A Hybrid Deep Model for Fake News Detection. *International Conference on Information and Knowledge Management, Proceedings, Part F131841*, 797–806. https://doi.org/10.1145/3132847.3132877

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature 1986 323:6088*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Saeed, M. (2022). *Employing transformers and humans for textual-claim verification | EURECOM*. https://www.eurecom.fr/en/publication/7052

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, *10*(3). https://doi.org/10.1371/JOURNAL.PONE.0118432

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2022). Detection of fake news using deep learning CNN–RNN based methods. *ICT Express*, *8*(3), 396–408. https://doi.org/10.1016/J.ICTE.2021.10.003

SATHVIK, M., Mishra, M. K., & Padhy, S. (2023). *Fake News Detection by Fine Tuning of Bidirectional Encoder Representations from Transformers*. https://doi.org/10.36227/TECHRXIV.22215403.V1

Schuetz, S. W., Sykes, T. A., & Venkatesh, V. (2021). Combating COVID-19 fake news on social media through fact checking: antecedents and consequences. *Https://Doi.Org/10.1080/0960085X.2021.1895682*, *30*(4), 376–388. https://doi.org/10.1080/0960085X.2021.1895682

Schuster, M., & Nakajima, K. (2012). Japanese and Korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5149–5152. https://doi.org/10.1109/ICASSP.2012.6289079

Seddari, N., Derhab, A., Belaoued, M., Halboob, W., Al-Muhtadi, J., & Bouras, A. (2022). A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media. *IEEE Access*, *10*, 62097–62109. https://doi.org/10.1109/ACCESS.2022.3181184

Shao, C., Ciampaglia, G. L., Varol, O., Yang, K. C., Flammini, A., & Menczer, F. (2018). The spread of low-credibility content by social bots. *Nature Communications 2018 9:1*, *9*(1), 1–9. https://doi.org/10.1038/s41467-018-06930-7

Shao, C., Hui, P. M., Wang, L., Jiang, X., Flammini, A., Menczer, F., & Ciampaglia, G. L. (2018). Anatomy of an online misinformation network. *PLoS ONE*, *13*(4). https://doi.org/10.1371/JOURNAL.PONE.0196087

Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating Fake News. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(3). https://doi.org/10.1145/3305260

Shi, B., & Weninger, T. (2015). Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. *Knowledge-Based Systems*, *104*, 123–133. https://doi.org/10.1016/j.knosys.2016.04.015

Shifath, S. M. S.-U.-R., Khan, M. F., & Islam, Md. S. (2021). *A transformer based approach for fighting COVID-19 fake news*. https://arxiv.org/abs/2101.12027v1

Shu, K., Bernard, H. R., & Liu, H. (2019). *Studying Fake News via Network Analysis: Detection and Mitigation*. 43–65. https://doi.org/10.1007/978-3-319-94105-9_3/FIGURES/6

Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019). Defend: Explainable fake news detection. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 395–405. https://doi.org/10.1145/3292500.3330935

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). *Fake News Detection on Social Media: A Data Mining Perspective*. https://arxiv.org/abs/1708.01967v3

Shu, K., Wang, S., Lee, D., & Liu, H. (2020). *Mining Disinformation and Fake News: Concepts, Methods, and Recent Advancements*. 1–19. https://doi.org/10.1007/978-3-030-42699-6_1

Shu, K., Wang, S., & Liu, H. (2019). Beyond News Contents: The Role of Social Context for Fake News Detection. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, *9*. https://doi.org/10.1145/3289600

Şişman, B., & Yurttaş, Ö. U. (2015). An Empirical Study on Media Literacy from the Viewpoint of Media. *Procedia - Social and Behavioral Sciences*, *174*, 798–804. https://doi.org/10.1016/J.SBSPRO.2015.01.662

Spohr, D. (2017). Fake news and ideological polarization: Filter bubbles and selective exposure on social media. *Http://Dx.Doi.Org/10.1177/0266382117722446*, *34*(3), 150–160. https://doi.org/10.1177/0266382117722446

Stanford History Education Group and Robert R. McCormick Tribune Foundation. (2016). *Evaluating information: The cornerstone of civic online reasoning*.

Tandoc, E. C., Lim, W., & Ling, R. (2017). *Digital Journalism Defining "Fake News."* https://doi.org/10.1080/21670811.2017.1360143

Thomas, J., Sreeraj, A., Sreeraj, A., Varghese, M. M., & Kuriakose, T. (2022). *Automatic Text Summarization Using Deep Learning and Reinforcement Learning*. 769–778. https://doi.org/10.1007/978-981-16-5157-1_60

Thota, A., Tilak, P., Ahluwalia, S., & Lohia, N. (2018). Fake News Detection: A Deep Learning Approach. *SMU Data Science Review*, *1*(3). https://scholar.smu.edu/datasciencereview/vol1/iss3/10

Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B. W. (2020). Fake news stance detection using deep learning architecture (CNN-LSTM). *IEEE Access*, *8*, 156695–156706. https://doi.org/10.1109/ACCESS.2020.3019735

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *2017-December*, 5999–6009. https://arxiv.org/abs/1706.03762v5

Volkova, S., Shaffer, K., Jang, J. Y., & Hodas, N. (2017). Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *2*, 647–653. https://doi.org/10.18653/V1/P17-2102

Vosoughi, S., Mohsenvand, M. 'Neo,' & Roy, D. (2017). Rumor Gauge. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *11*(4). https://doi.org/10.1145/3070644

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science (New York, N.Y.)*, *359*(6380), 1146–1151. https://doi.org/10.1126/SCIENCE.AAP9559

Wasike, J. (2013). Social media ethical issues: role of a librarian. *Library Hi Tech News*, *30*(1), 8–16. https://doi.org/10.1108/07419051311320922/FULL/XML

Waszak, P. M., Kasprzycka-Waszak, W., & Kubanek, A. (2018). The spread of medical fake news in social media – The pilot quantitative study. *Health Policy and Technology*, *7*(2), 115–118. https://doi.org/10.1016/J.HLPT.2018.03.002

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. https://arxiv.org/abs/1609.08144v2

Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). *TI-CNN: Convolutional Neural Networks for Fake News Detection*. https://arxiv.org/abs/1806.00749v3

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, *32*. https://github.com/zihangdai/xlnet

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending Against Neural Fake News. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1905.12616v3

Zhang, C., Gupta, A., Kauten, C., Deokar, A. V., & Qin, X. (2019). Detecting fake news for reducing misinformation risks using analytics approaches. *European Journal of Operational Research*, *279*(3), 1036–1052. https://doi.org/10.1016/J.EJOR.2019.06.022

Zhang, J., Cui, L., Fu, Y., & Gouza, F. B. (2018). Fake News Detection with Deep Diffusive Network Model. *ArXiv*. https://doi.org/arXiv:1805.08751v1

Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, *57*(2), 102025. https://doi.org/10.1016/J.IPM.2019.03.004

Zhang, X., Zhao, J., & Lecun, Y. (2015). Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*, *2015-January*, 649–657. https://arxiv.org/abs/1509.01626v3

Zhou, X., & Zafarani, R. (2020). A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys (CSUR)*, *53*(5). https://doi.org/10.1145/3395046

Web Links

[i] https://www.theverge.com/2020/10/18/21521685/facebook-clegg-voting-instagram-ads-fact-check-zuckerberg

[ii] https://www.martenscentre.eu/publication/weeding-out-fake-news-an-approach-to-social-media-regulation/

[iii] https://www.nytimes.com/interactive/2017/06/23/opinion/trumps-lies.html

[iv] https://www.who.int/publications/m/item/covid-19-research-and-innovation---powering-the-world-s-pandemic-response-now-and-in-the-future

[v] Retrieved March 21, 2023, from https://www.bbc.com/news/business-36189252

[vi] Retrieved March 21, 2023, from https://www.poynter.org/coronavirusfactsalliance/

[vii] Retrieved March 21, 2023, from https://www.prnewswire.com/news-releases/2020-the-year-of-fake-news-covid-related-scams-and-ransomware-301180568.html

[viii] Retrieved March 22, 2023, from https://www.pewresearch.org/fact-tank/2021/01/12/more-than-eight-in-ten-americans-get-news-from-digital-devices/

[ix] Retrieved March 22, 2023, from https://www.bbc.com/news/60589965

[x] https://www.indiaspend.com/manipulative-fake-news-on-the-rise-in-india-under-lockdown-study/

[xi] Retrieved March 22, 2023, from https://ijasre.net/index.php/ijasre/article/view/1007

[xii] Retrieved March 22, 2023, from https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[xiii] Retrieved March 22, 2023, from https://www.financemagnates.com/cryptocurrency/news/john-mcafee-twitter-handle-hack-results-pump-dump-multiple-coins/

[xiv] Retrieved March 22, 2023, from http://www.chinadaily.com.cn/2011-03/18/content_12189516.htm

[xv] https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/?sh=2c7b76af2fac

[xvi] https://www.statista.com/chart/6795/fake-news-is-a-real-problem/

[xvii] https://www.washingtonpost.com/news/the-intersect/wp/2016/11/17/facebook-fake-news-writer-i-think-donald-trump-is-in-the-white-house-because-of-me/

[xviii] https://www.independent.co.uk/news/world/middle-east/middle-east-fake-news-consequences-israel-saudi-arabia-syria-lebanon-hezbollah-mossad-qatar-uae-a7936621.html

[xix] https://blog.twitter.com/en_us/topics/product/2021/introducing-birdwatch-a-community-based-approach-to-misinformation

[xx] Retrieved April 4, 2023, from https://www.facebook.com/help/1952307158131536?_rdc=2&_rdr

[xxi] Retrieved April 4, 2023, from https://blog.twitter.com/en_us/topics/product/2020/updating-our-approach-to-misleading-information

[xxii] Retrieved April 10, 2023, from https://www.internetmatters.org/hub/news-blogs/stopping-the-spread-of-fake-news-on-popular-online-platforms/?fbclid=IwAR0xIh2yqkmfLLBf5yIgjyrlEC57Ug0XePQBi2Hs1UBklDy51t8BgYXWf88

[xxiii] Retrieved April 10, 2023, from http://www.derczynski.com/sheffield/papers/pheme-eswc-pn.pdf