

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Mohamed Boudiaf- MSILA

Faculty of Mathematics and informatics
Department of Computer Science
N° :.....



DOMAIN: Mathematics and Computer Science
SPECIALITY: Business Intelligence and Optimization
Presented For Obtaining The Grade of MASTER

By:
Babi Chaima

ENTITLED

**Fake News Detection Using Machine Learning and
Deep Learning Techniques**

Presented in front of the jury composed of:

Lounes Bilal	Msila University	President
Gadri Said	Msila University	Supervisor
Chalabi NourElhouda	Msila University	Reviewer

Academic Year: 2022 / 2023

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Mohamed Boudiaf- MSILA

Faculty of Mathematics and informatics

Department of Computer Science

N° :.....



DOMAIN: Mathematics and Computer Science

SPECIALITY: Business Intelligence and Optimization

Presented For Obtaining The Grade of MASTER

By:

Babi Chaima

ENTITLED

**Fake News Detection Using Machine Learning and
Deep Learning Techniques**

Presented in front of the jury composed of:

Lounes Bilal	Msila University	President
Gadri Said	Msila University	Supervisor
Chalabi NourElhouda	Msila University	Reviewer

Academic Year: 2022 / 2023

Dedication

I dedicate this thesis to To my parents, my greatest inspirations and unwavering supporters. Mom, your love and encouragement pulled me through my darkest times, when I almost quit. Dad, your memory fuels my determination to succeed. Your sacrifices and belief in me will forever be cherished. I strive to make you both proud. Thank you for everything.

Acknowledgement

I would like to express my heartfelt gratitude to Almighty Allah for His guidance and blessings throughout the completion of this work.

I am deeply thankful to my supervisor, **Dr.Kadri Said**, for his invaluable guidance and unwavering support.

I express my gratitude to each and every member of the esteemed jury panel for their diligent evaluation and meticulous assessment.

I extend my appreciation to the faculty members of **University of M'sila** for their knowledge and dedication.

Contents

List of Figures	vii
List of Tables	viii
List of Acronyms	ix
General Introduction	10
1 Fake News Detection (FND)	11
1 Introduction	12
2 Related Work	12
2.1 Machine Learning and Deep Learning Techniques	12
3 Fake News: Definition and Characteristics	14
3.1 Fake News Definition	14
3.2 Types and Manifestations of Fake News	14
3.3 Key Characteristics of Fake News	15
4 The Impacts of Fake News	17
5 Fake News Detection	18
5.1 Definition	18
5.2 Importance of Fake News Detection	18
6 The Challenges in Detecting Fake News	19
6.1 Information Overload	19
6.2 Rapid Creation and Adaptation	19
6.3 Confirmation Bias	19
6.4 Misinformation Ecosystems	19
7 Approaches to Fake News Detection	20
7.1 Traditional Approaches	20
7.2 Machine Learning Approaches	20
7.3 Deep learning Approaches	21

8	Conclusion	22
2	Machine learning and Deep learning	23
1	Introduction	24
2	Machine Learning	24
2.1	Definition	24
2.2	Machine Learning types	24
3	Deep Learning	31
3.1	Definition	31
3.2	Artificial Neural Networks	31
4	Artificial Neural Networks types	33
4.1	Deep Feed-Forward Neural Network(FNN)	33
4.2	Convolutional Neural Network	33
4.3	Deep Recurrent Neural Network	34
5	ML vs DL	38
6	Conclusion	38
3	Environment Tools	39
1	Introduction	40
2	Tools Presentation	40
2.1	Hardware	40
2.2	System Requirements	40
3	Conclusion	45
4	Realized work and Obtained Results	46
1	Introduction	47
2	Dataset	47
2.1	Dataset Description	47
2.2	Dataset Preparation	47
2.3	Dataset Preprocessing	48
3	The Proposed BiLSTM Model	49
3.1	Build the Model	49
3.2	The Obtained Results	50
3.3	The Evaluation of the Model	52
4	Comparing Obtained Results of ML Models with BiLSTM model	55
5	Application Interface	56
5.1	Home Page	56
5.2	Tools Page	58

Contents

5.3	About Page	59
6	Discussion	60
7	Conclusion	61
General Conclusion		62
Bibliography		63

List of Figures

2.1	Machine learning types [40]	24
2.2	Logistic Regression Example [42]	26
2.3	Decision Tree [44]	27
2.4	K-Nearest Neighbors (KNN) Model [47]	28
2.5	Reinforcement in ML [53]	30
2.6	Activation Function Types [56]	32
2.7	Structure of a hidden layer feed-forward neural network	33
2.8	CNN Model Architecture [61]	34
2.9	Deep Recurrent Neural Network schema	35
2.10	LSTM unit with input and output connections [64]	36
2.11	BiLSTM model with enabling feature. [67]	37
3.1	conda install pandas	43
3.2	install pip	44
3.3	Anaconda Navigator	45
4.1	Dataset Capture	47
4.2	Model Architecture	50
4.3	Model Training	50
4.4	Training and validation accuracy	51
4.5	Training and validation loss	52
4.6	Capture of the Menu of the application interface	56
4.7	Capture of the Home Page interface	57
4.8	Capture of "Enter"and"Upload" functions	58
4.9	Capture classifying the article as fake	58
4.10	Capture of the Tools Page interface	59
4.11	Screenshot of the About Page interface	60

List of Tables

1.1	Key Characteristics of Fake News	16
1.2	Approaches to Fake News Detection	22
2.1	Comparison between machine learning (ML) and deep learning (DL)	38
4.1	Description of the proposed LSTM Model	50
4.2	Accuracy Comparison of Machine Learning Models	55

List of Acronyms

FND Fake News Detection

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

RNN Recurrent Neural Network

CNN Convolutional Neural Network

LR Logistic regression

DT Decision Trees

NB Naive Bayes

KNN K-Nearest Neighbors

PCA Principal Component Analysis

GAN Generative Adversarial Networks

ANN Artificial Neural Network

ReLU Rectified Linear Unit

tanh Hyperbolic tangent function

FNN Feedforward Neural Networks

BPTT Backpropagation Through Time

LSTM Long Short-Term Memory

BiLSTM Bidirectional Long Short-Term Memory

General Introduction

The advent of artificial intelligence (AI), and specifically artificial neural networks (ANNs), has played a significant role in addressing the challenge of detecting fake news. AI refers to the development of computer systems capable of performing tasks that typically require human intelligence, such as learning, reasoning, and decision-making. ANN, a subset of AI, emulates the structure and function of the human brain to process information and make predictions.

By incorporating AI and ANN techniques into the field of fake news detection, researchers and practitioners have been able to improve the accuracy and effectiveness of computational approaches. These technologies allow algorithms to analyze large amounts of data, identify patterns, and learn from examples, enabling them to distinguish between real and false information. The use of machine learning and deep learning algorithms has shown promise in identifying the intricate characteristics and subtle nuances associated with fake news. Therefore, this thesis aims to harness the power of AI and ANN in the context of fake news detection. By leveraging these advanced computational techniques, robust and reliable mechanisms to combat the spread of fake news and preserve the integrity of information are to be developed.

This document is divided into the following chapters:

Chapter 1: Fake News Detection.

Chapter 2: Machine learning and Deep learning.

Chapter 3: Environment Tools.

Chapter 4: The realized work and the obtained results.

Chapter 1

Fake News Detection (FND)

1 Introduction

The rise of fake news in the digital era has posed significant challenges to society, necessitating effective fake news detection mechanisms. This chapter provides a comprehensive overview of the field of fake news detection, exploring its definition, characteristics, impacts, challenges, and various approaches.

2 Related Work

2.1 Machine Learning and Deep Learning Techniques

1. "A Comparative Study" by Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta (2020) explores sentiment analysis techniques using deep learning [1]. The study compares different approaches and models for analyzing sentiment in textual data, such as social media posts or customer reviews. The findings provide insights into the strengths and weaknesses of deep learning models for sentiment analysis tasks, aiding researchers and practitioners in selecting appropriate models for their specific needs.
2. "An Efficient System to Predict Customers' Satisfaction on Touristic Services Using ML and DL Approaches" [2]. This study by Said et al. presents an efficient system for predicting customers satisfaction in the tourism industry. The authors employ machine learning and deep learning approaches to achieve accurate predictions. The work was presented at the 22nd Arabic International Conference on Information Technology (ACIT 2021) and can be found in the IEEE Conference proceedings. It provides insights into the application of machine learning and deep learning techniques for customer satisfaction prediction in the tourism domain.
3. In the study conducted by L. Borges, B. Martins, and P. Calado [3]. The authors investigate the problem of stance detection in the context of checking fake news. Their research focuses on combining similarity features and deep representation learning techniques to accurately determine the stance (supportive, opposing, or neutral) of a given text towards a specific claim or topic. The proposed approach demonstrates promising results in improving fake news detection and verification methods, providing valuable insights for addressing the challenges associated with identifying the stance of textual content.
4. "Efficient Traffic Signs Recognition Based on CNN Model for Self-Driving Cars" [4]. Gadri and Adouane propose an efficient traffic sign recognition system based on

- a Convolutional Neural Network (CNN) model. The study focuses on enabling reliable recognition of traffic signs for self-driving cars. The work was presented at the Intelligent Computing & Optimization (ICO 2021) conference and can be found in the Lecture Notes in Networks and Systems by Springer. It contributes to the development of accurate traffic sign recognition techniques for autonomous driving.
5. The spread of true and false news online is a seminal article published in Science in 2018 by S. Vosoughi, D. Roy, and S. Aral [5]. The study investigates the dissemination patterns and dynamics of true and false news on online platforms. It provides valuable insights into the mechanisms influencing information spread, contributing to our understanding of the challenges posed by misinformation. This article is highly relevant to my thesis on information verification and credibility assessment.
 6. Sentiment Analysis: Developing an Efficient Model Based on Machine Learning and Deep Learning Approaches [6]. Gadri et al. present a study on sentiment analysis, aiming to develop an efficient model using machine learning and deep learning approaches. The authors explore techniques to accurately analyze and classify sentiment in textual data. This work was also presented at the ICO 2021 conference and can be found in the Lecture Notes in Networks and Systems by Springer. It offers insights into the development of efficient sentiment analysis models using deep learning techniques.
 7. Detecting fake news for reducing misinformation risks using analytics approaches, by Zhang et al [7]. The researchers train a combination of machine learning algorithms using ensemble methods such as bagging, boosting, or stacking. Real-world datasets are used for training and evaluating the performance of the ensemble learners.
 8. Convolutional neural networks for stance detection on fake news, by Monti et al [8]. The authors apply convolutional neural networks (CNNs) to detect the stance (e.g., agreement, disagreement) of news articles towards a given claim, aiding in fake news detection.
 9. Said G. Efficient Arabic handwritten character recognition based on machine learning and deep learning approaches [9]. This article presents a study on Arabic handwritten character recognition using machine learning and deep learning techniques. The author explores various approaches to achieve efficient recognition and discusses the results obtained. The research is published in the Journal of Advanced Research in Dynamical & Control Systems, an indexed journal.

10. Said G. Developing an efficient Predictive model based on ML and DL approaches to detect diabetes [10]. This article focuses on the development of a predictive model using machine learning and deep learning approaches to detect diabetes. The author presents various techniques and methodologies employed in the model and discusses its efficiency. The research is published in The International Journal of Computing and Informatics Informatica.

Social Network Analysis

1. The spreading of misinformation online, by Vicario et al [11]. This study investigates the dynamics of misinformation spread in online social networks and explores the factors contributing to its propagation.
2. Political polarization on Twitter, by Conover et al [12]. The authors analyze political polarization on Twitter, including the dissemination of false or misleading information within polarized communities.
3. Rumor cascades, by Friggeri et al [13]. This study examines the characteristics and dynamics of rumor propagation in online social networks, providing insights into the spread of false information.

3 Fake News: Definition and Characteristics

3.1 Fake News Definition

Fake news can be defined as intentionally false information presented as news to deceive or mislead readers or viewers. It often mimics the format and style of traditional news media, making it difficult for consumers to distinguish between legitimate news and falsehoods. Fake news can be created and spread through various mediums, including social media platforms, websites, blogs, and even traditional news outlets. [14]

3.2 Types and Manifestations of Fake News

Fake news can take different forms, each serving a specific purpose or agenda. Some common types and manifestations include:

Fabricated News

Fabricated news refers to entirely fictional stories created to deceive readers or viewers. These stories are invented and lack any factual basis. They often aim to attract attention, generate clicks, or promote specific viewpoints or ideologies. [15]

Manipulated Content

Manipulated content involves the alteration or distortion of genuine information to mislead the audience. This can include selectively edited videos or images, misquoted statements, or out-of-context information. Manipulated content often aims to create a false narrative or reinforce preconceived notions. [16]

Satire or Parody:

Satirical or parody news articles and websites present fictional or exaggerated stories in a humorous or satirical manner. While these pieces are not intended to deceive, they can sometimes be misconstrued as real news, especially when taken out of context or shared without proper attribution. [17]

Clickbait News:

Clickbait refers to sensational or misleading headlines designed to grab attention and generate web traffic. These headlines often lure readers into clicking on the article, only to find that the content does not match the exaggerated or misleading claims made in the headline. Clickbait can be a gateway to spreading fake news. [18]

3.3 Key Characteristics of Fake News

Identifying fake news requires a critical evaluation of certain key characteristics that often accompany such misinformation :

Inaccurate Information

Fake news typically contains factual inaccuracies or distortions. This can include fabricated quotes, misrepresented statistics, or false narratives designed to mislead or manipulate the audience. [5]

Biased or Partisan Language

Fake news often employs biased or partisan language to appeal to specific ideological or political affiliations. This language aims to manipulate readers' emotions and reinforce existing beliefs, further polarizing public opinion. [19]

Lack of Credible Sources

Authentic news articles rely on verifiable sources and journalistic standards. In contrast, fake news often lacks credible sources, referencing unverified or anonymous individuals, or even presenting nonexistent experts or organizations. [14]

Virality and Speed of Spread

One notable characteristic of fake news is its rapid spread across social media platforms and online communities. Due to the viral nature of social sharing, fake news can quickly reach a wide audience, making it challenging to contain or refute. [20]

Characteristic	Description
Inaccurate Information	Fake news contains factual inaccuracies or distortions, including fabricated quotes, misrepresented statistics, or false narratives.
Biased or Partisan Language	Fake news uses biased or partisan language to appeal to specific ideologies or political affiliations, manipulating readers' emotions and reinforcing existing beliefs.
Lack of Credible Sources	Fake news lacks credible sources and often references unverified or anonymous individuals, or presents nonexistent experts or organizations.
Virality and Speed of Spread	Fake news spreads rapidly across social media platforms and online communities, reaching a wide audience due to the viral nature of social sharing.

Table 1.1: Key Characteristics of Fake News

4 The Impacts of Fake News

Fake news has profound impacts for society, politics, and the economy and even individuals:

Erosion of Trust

The dissemination of fake news undermines public trust in news sources and institutions, leading to a widespread skepticism towards information. This erosion of trust can have detrimental effects on democratic processes, public discourse, and social cohesion. [15]

Polarization and Divisiveness

Fake news often employs biased or partisan language to appeal to specific ideological or political affiliations, further polarizing public opinion. This can deepen societal divisions and hinder constructive dialogue among individuals with differing viewpoints. [15]

Manipulation of Public Opinion

The deliberate spread of misinformation can manipulate public opinion and influence electoral outcomes, policy debates, and public sentiment. False narratives and distorted information can shape people's perceptions, potentially leading to misguided decisions and actions. [21]

Economic Impact

The deliberate spread of misinformation can manipulate public opinion and influence electoral outcomes, policy debates, and public sentiment. False narratives and distorted information can shape people's perceptions, potentially leading to misguided decisions and actions. [22]

Social Media Amplification

Social media platforms play a pivotal role in the rapid dissemination of fake news. Their algorithms and design features often prioritize engagement and viral content, leading to the rapid spread of false information within online communities. [23]

Echo Chambers and Filter Bubbles

Fake news tends to thrive in echo and filter chambers, where people are mainly presented with content that corresponds to their existing views and opinions. [24]

Viral Nature and Emotional Appeal

Fake news often exploits emotional triggers to capture attention and elicit strong reactions. Content designed to evoke anger, fear, or excitement is more likely to be shared widely, contributing to the viral spread of misinformation. [25]

5 Fake News Detection

5.1 Definition

Fake News Detection (FND) is the process of identifying and categorizing news articles that are intentionally produced or altered to mislead audiences. It involves analyzing text, sources, and distribution patterns to determine authenticity. Mathematically, FND can be defined as a binary classification problem where the goal is to predict whether a given news story is fake or not based on a set of input features. These features may include the text of the article, the source of the article, the publication date, and other relevant metadata. [26]

$$F(a) = \begin{cases} 1 & \text{if } a \text{ is a piece of fake news} \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

where F is the prediction function we want to learn.

5.2 Importance of Fake News Detection

The detection of fake news has become increasingly important in today's society due to its potential consequences. Fake news can influence public opinion, spread misinformation, and even incite violence. Therefore, identifying fabricated news can hinder the dissemination of false information and support the accuracy of news reporting.

In addition, fake news detection can help maintain trust in news sources and the media, which is essential for a healthy democracy. By identifying and eliminating false information, individuals can have greater confidence in the data they collect and the judgments they make based on that data.

Numerous studies and research papers have been conducted on fake news detection techniques, highlighting its importance in maintaining the integrity of information in our society. Therefore, it is crucial to continue investing in and developing effective fake news detection methods to combat the spread of false information. [27]

6 The Challenges in Detecting Fake News

Detecting and combating fake news present numerous challenges that complicate the task of ensuring accurate information. Key challenges include:

6.1 Information Overload

The vast amount of information available online makes it challenging for individuals to differentiate between accurate and false news. The abundance of content, combined with limited time and cognitive resources, increases the risk of consuming and spreading misinformation. [28]

6.2 Rapid Creation and Adaptation

The rapid creation and adaptation of online information further compounds the challenge for individuals to distinguish between accurate and false news. The sheer volume of content, coupled with limited time and cognitive resources, heightens the risk of inadvertently consuming and propagating misinformation. [29]

6.3 Confirmation Bias

Confirmation bias, the tendency to seek and interpret information that aligns with preexisting beliefs, makes it challenging to convince individuals to critically evaluate information and accept corrections. People may be more inclined to accept and share fake news that confirms their existing viewpoints. [30]

6.4 Misinformation Ecosystems

Fake news exists within a complex ecosystem involving creators, distributors, and consumers. Addressing the issue requires collaborative efforts from social media platforms, fact-checking organizations, educators, policymakers, and individuals to combat misinformation effectively. [31]

7 Approaches to Fake News Detection

7.1 Traditional Approaches

Traditional approaches to fake news detection rely on established techniques that do not involve sophisticated machine learning algorithms. These approaches include:

Rule-based Methods

Rule-based methods utilize a predefined set of rules or criteria to identify and classify fake news articles. These rules are typically based on specific characteristics, patterns, or indicators associated with fake news, and they serve as decision-making guidelines to determine the authenticity and credibility of news content. Rule-based methods rely on explicit rules or patterns to make determinations rather than relying on statistical modeling or machine learning algorithms. [32]

Content-based Methods

Content-based methods analyze the textual content of news articles to detect fake news. They often focus on linguistic features, such as the use of emotional language, exaggerated claims, or the presence of misinformation indicators like false statistics or fabricated quotes. Content-based methods may also examine metadata, such as publishing sources or timestamps, to determine the credibility of the information. [33]

Source-based Methods

Source-based methods evaluate the credibility and reputation of the sources from which the news originates. This approach considers factors such as the track record of the source, its expertise, and whether it adheres to professional journalistic standards. Source-based methods may utilize external databases or fact-checking organizations to verify the reliability of the sources. [34]

7.2 Machine Learning Approaches

Machine learning approaches can effectively handle large volumes of data and adapt to evolving fake news techniques. Key machine learning techniques for fake news detection include:

Feature Engineering for Fake News Detection

Feature engineering involves extracting relevant features from the news data to represent important characteristics associated with fake news. These features can include lexical and syntactic features, sentiment analysis, readability measures, or social network features. The selection and engineering of appropriate features significantly impact the performance of machine learning models. [35]

Supervised Learning Algorithms

Supervised learning algorithms employ techniques to identify patterns and create predictive models that can classify unseen news articles. [36]

Unsupervised Learning Algorithms

Unsupervised learning algorithms aim to identify patterns and clusters within the news data without prior knowledge of fake news labels. [37]

7.3 Deep learning Approaches

Deep learning approaches, employ neural networks with multiple hidden layers to automatically learn intricate patterns and representations from the news data. [38]

Approach	Description
Traditional Approaches	
Rule-based Methods	Analyze textual content and linguistic features to detect fake news.
Content-based Methods	Analyze textual content and metadata to determine credibility.
Source-based Methods	Evaluate credibility and reputation of news sources.
Machine Learning Approaches	
Feature Engineering	Extract relevant features to represent characteristics associated with fake news.
Supervised Learning Algorithms	Create predictive models to classify news articles.
Unsupervised Learning Algorithms	Identify patterns and clusters without prior knowledge of fake news labels.
Deep Learning Approaches	
Deep Learning Techniques	Utilize neural networks with multiple hidden layers to automatically learn patterns and representations.

Table 1.2: Approaches to Fake News Detection

8 Conclusion

In conclusion, the chapter provides a comprehensive introduction to the field of fake news detection. It underscores the significance of addressing the challenges posed by fake news and highlights the potential impacts on society. By presenting a range of detection approaches, the chapter sets the stage for further exploration and development of effective mechanisms to combat the spread of fake news.

Chapter 2

Machine learning and Deep learning

1 Introduction

The field of machine learning (ML) and deep learning (DL) has experienced tremendous growth in recent years, with applications in various domains. This chapter aims to introduce the basic concepts of ML and DL, such as supervised, unsupervised, and reinforcement learning, data preprocessing, feature engineering, and model selection. It also covers DL architectures such as Recurrent Neural Network(RNN) and Convolutional Neural Network (CNN), and provides an overview of current trends and challenges in the field.

2 Machine Learning

2.1 Definition

Machine learning (ML) is a specialized branch of artificial intelligence that is dedicated to constructing algorithms and statistical models that enable computer systems to learn from empirical data and improve performance iteratively, without the need for explicit programming. Machine learning algorithms are engineered to learn and advance automatically from data by recognizing intricate patterns, connections, and valuable insights within complex and extensive datasets, resulting in improved performance overtime. [39]

2.2 Machine Learning types

Machine learning can be divided into two, three, or more categories depending on the context and goals of the discussion. ML can be categorized into supervised, unsupervised, semi-supervised, and reinforcement learning.

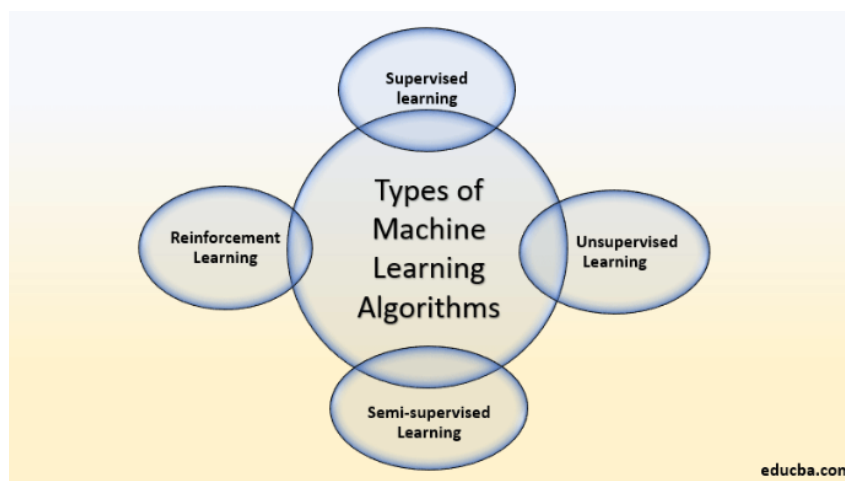


Figure 2.1: Machine learning types [40]

Supervised Learning

Supervised learning is a type of machine learning where the algorithm is trained on a labeled dataset, which means that the output variable is already known. The algorithm learns to map the input to the output based on the labeled data. This type of learning is used in various applications, such as image classification, speech recognition, and natural language processing. [36]

Supervised Learning Algorithms

1. Logistic regression (LR)

Is a regression analysis method that is commonly used for binary classification tasks to model the probability of a binary target variable (such as true/false, yes/no, or 0/1) based on one or more input variables (predictors or features). Logistic regression operates by fitting a hypothesis function, often represented by a sigmoid function(also called a logistic function) to the input data, which maps each real-valued input to a value between 0 and 1, indicating the probability of the positive class. Logistic regression estimates the probability of the output variable with the input variables by applying the logistic function :

$$P(y = 1|x) = \frac{1}{1 + e^{-z}} \quad (2.1)$$

where y is the binary target variable, x is a vector of input variables, and z is a linear combination of the input variables and their corresponding weights:

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_px_p$$

In this equation, w_0 is the intercept term, while $w_1, w_2, \dots, w_p$ are the weights assigned to the input variables x . [41]

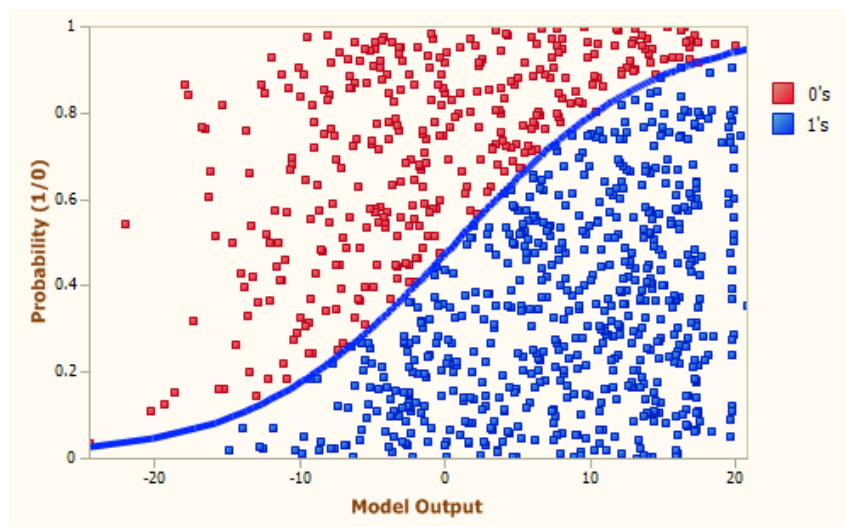


Figure 2.2: Logistic Regression Example [42]

2. Decision Trees (DT)

Is a nonparametrically based technique for classification and regression problems that represents a structure similar to a flowchart with internal nodes representing features or attributes, branches representing the decision rules based on those features, and leaf nodes representing the result or prediction. It is employed for regression and classification tasks on the basis of entropy function and information gain.

Entropy is used to quantify the uncertainties or randomnesses associated with the class labels of the data instances. It is used to decide the best characteristic for dividing the data at every inner node. Mathematically, entropy is calculated by the following formula:

$\text{Entropy}(S) = -\sum (p(i) \cdot \log_2(p(i)))$ where S is the data set, $p(i)$ is the proportion of instances belonging to class i , and the summation is over all the unique .The entropy value ranges from 0 to 1, where 0 represents a completely pure dataset(all instances belong to the same class) Information gain measures the effectiveness of a particular feature in classifying the instances in a data set. It quantifies the entropy reduction obtained by splitting the data set based on that feature. At every internal node of the decision tree, the feature with the highest information gain is selected as the split criterion.

Mathematically, the information gain is calculated according to the following formula

$$\text{Information Gain}(S, A) = \text{Entropy}(S) - \sum \left(\frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \right)$$

Where S is the data set, A is a particular feature, S_v represents the subset of instances for which feature A has value v , and the summation is performed over all possible values of feature A . Information gain is calculated by subtracting the weighted average of entropies of the resulting subsets from the entropy of the original dataset. [43]

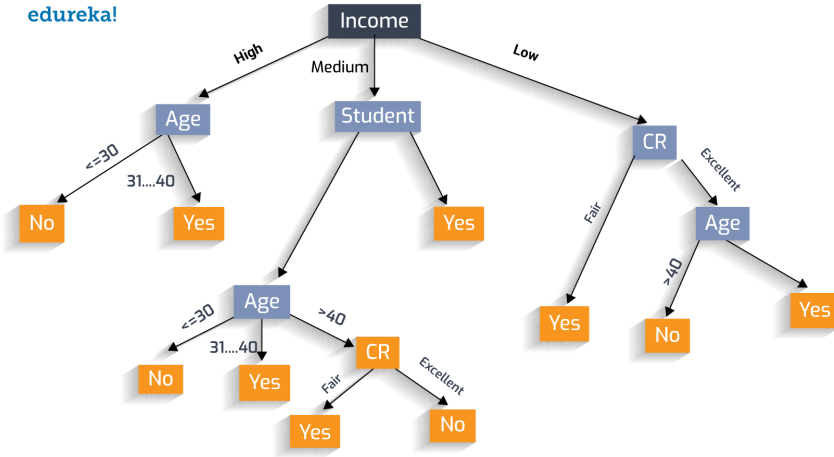


Figure 2.3: Decision Tree [44]

3. Naive Bayes (NB)

Is a probabilistic classifier that is based on the Bayes theorem. According to this theorem, the probability of a hypothesis (such as a class label) given some observed evidence (a set of features) is proportional to the prior probability of the hypothesis and the probability of the evidence given the hypothesis. Naive Bayes computes the posterior probability of each potential class label based on the observed features, and then assigns the input to the class with the highest probability.

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(y) \times P(x_1|y) \times P(x_2|y) \times \dots \times P(x_n|y)}{P(x_1, x_2, \dots, x_n)} \quad (2.2)$$

where $P(y|x_1, x_2, \dots, x_n)$ is the probability of class y given the feature values $x_1, x_2, \dots, x_n$; $P(y)$ is the prior probability of class y ; $P(x_i|y)$ is the conditional probability of feature x_i given class y ; and $P(x_1, x_2, \dots, x_n)$ is the probability of observing the feature values $x_1, x_2, \dots, x_n$. [45]

4. K-Nearest Neighbors (KNN)

Is a non-parametric classifier that can be used for either classification or regression tasks. It forecasts the result of a new data point by considering the k most similar data points in the training set, where k can be adjusted. The class or value of

the new data point is determined by the majority vote or average of the k nearest neighbors. Once a new data point is provided, the predicted class is specified as:

$$y = \underset{y' \in k\text{-closest-neighbors}(x)}{\operatorname{argmax}} \left(\sum \operatorname{count}(y') \right) \quad (2.3)$$

where x is the new data point, y is the predicted class, and k is the number of neighbors considered. [46]

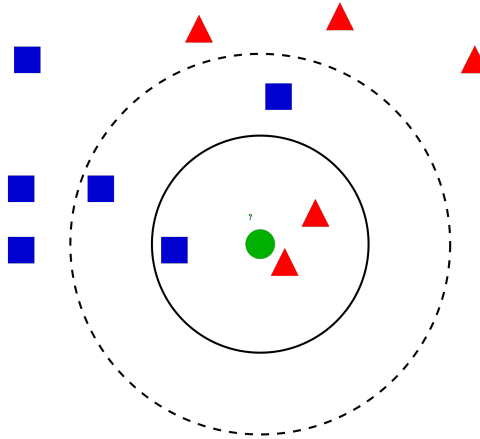


Figure 2.4: K-Nearest Neighbors (KNN) Model [47]

Supervised Learning Applications

- Image classification: classifying images into different groups or categories using their distinguishing features.
- Speech transcription: recognizing spoken words and converting them into written text.
- Emotion analysis: determining the emotions conveyed in a piece of text, whether they are positive, negative, or neutral.
- Fraud detection: identifying fraudulent activities or transactions using historical data and patterns.
- Credit risk prediction - forecasting the probability of default or credit risk for an individual or organization based on their credit history and other relevant factors.
- Language processing: analyzing and generating natural language text, as seen in language translation programs or chatbots.

Unsupervised Learning

Unsupervised learning is a type of machine learning where the algorithm is trained on an unlabeled dataset, which means that the output variable is unknown. The algorithm learns to discover patterns and relationships within the data based on the inherent structure of the data-set. This type of learning is used in various applications, such as anomaly detection, clustering, and dimensionality reduction. [37]

Unsupervised Learning Algorithms

1. The K-means clustering technique:
is a straightforward algorithmic approach that groups comparable data points into k clusters.
2. Principal Component Analysis (PCA):
is an approach used to decrease the complexity of a dataset by identifying the most significant components, ultimately reducing the number of features.
3. Generative Adversarial Networks (GANs):
A type of model that is capable of generating new data samples that bear resemblance to the data used for training.

Unsupervised Learning Applications

- Cluster analysis: assembling similar data points together (e.g. for client categorization or image categorization).
- recognizing uncommon data points in a dataset (e.g. for identifying fraud or detecting malfunctions in a system).
- Dimensionality reduction: decreasing the number of variables in a dataset while upholding the most significant information (e.g. for data presentation or characteristic selection).

Semi-Supervised Learning

Semi-supervised learning is a type of machine learning where the algorithm is trained on a combination of labeled and unlabeled data. This type of learning is used in various applications, such as sentiment analysis, where the sentiment of a sentence can be inferred from the context of the words, even if the sentence is not labeled as positive or negative. [48]

Semi-Supervised Learning Algorithms

- **Self-education:** the process of acquiring knowledge and skills through personal initiative and independent learning. [49]
- **Co-training:** A machine learning technique where multiple models are trained collaboratively using a shared dataset and iteratively improving each other's performance. [50]
- **Multi-perspective:** The consideration and integration of diverse viewpoints or angles to gain a more comprehensive understanding of a subject or situation. [51]

Semi-Supervised Learning Applications

- Image Categorization: uses a limited number of labeled images.
- Opinion Mining: uses a limited number of labeled text documents.
- Voice Recognition: uses a small number of labeled audio recordings.

Reinforcement Learning

Reinforcement learning is a type of machine learning where the algorithm learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The algorithm learns to take actions that maximize the cumulative reward over time. This type of learning is used in various applications, such as game playing, robotics, and autonomous driving. [52]

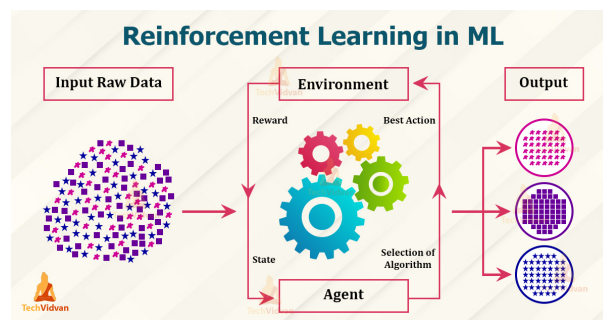


Figure 2.5: Reinforcement in ML [53]

Reinforcement Learning Algorithms

- Q-learning

- Policy gradients
- Actor-Critic

Reinforcement Learning Applications

- Gaming: Training an agent to play games such as chess, Go, or video games.
- Robotics: Training a robot to perform tasks in a physical environment.
- Autonomous vehicles: Training a vehicle to navigate and make decisions in traffic.

3 Deep Learning

3.1 Definition

Deep learning is a subset of machine learning that uses artificial neural networks to learn patterns and extract unique characteristics from large and complex data sets. It has been beneficial in pharmaceutical research, genetics, and astrophysics, and has resulted in progress in areas such as pharmaceutical research, genetics, and astrophysics. [38]

3.2 Artificial Neural Networks

An artificial neural network (ANN) is a simulation of the structure and behavior of a natural neuron. It consists of several layers of neurons, including an input layer and an output layer. Each neuron receives inputs from other neurons or external sources and applies a mathematical function to produce an output that is then sent to other neurons in the network. Weights are adjusted during training to reduce the difference between the network's actual output and the desired output. [54]

3.2.1 Activation function

An activation function is used to introduce nonlinearity into a network to learn complex patterns from input data. Formally, an activation function f takes an input x and generates an output Y , that gets passed to the next layer of the network. In the case of a neuron, the output y is computed as: $y = f(wx + b)$ where w is a vector of learning weights and b is a bias term and f is the activation function. Frequently used activation functions include the sigmoid function, the hyperbolic tangent function, and the rectified linear unit (ReLU) function. [55]

Activation Functions

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

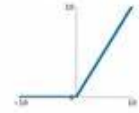


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Figure 2.6: Activation Function Types [56]

- **The Rectified Linear Unit (ReLU):** is the predominant activation function used in deep learning models. It works by producing a value of 0 for all input values that are negative, and for all non-negative input values, it produces the same value as the input [57]. This can be mathematically expressed as:

$$f(x) = \max(0, x) \quad (2.4)$$

- **Sigmoid:** The sigmoid function is a non-linear function that maps any input value to a value between 0 and 1, which makes it useful for a variety of tasks such as binary classification, where the output of the network represents the probability of a particular class. The formula for the sigmoid function is:

$$y = \frac{1}{1 + e^{-l \times s}} \quad (2.5)$$

with $s = \sum_{i=1}^n W_i x_i + b$ where b is the bias and l is a positive constant that determines the steepness of the sigmoid function. [58]

- **Hyperbolic tangent function (tanh):** The hyperbolic tangent function is a non-linear function that maps input values to a range between (-1) and (1). The formula for the hyperbolic tangent function is [57]:

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (2.6)$$

4 Artificial Neural Networks types

4.1 Deep Feed-Forward Neural Network(FNN)

Depending on the way the neurons are connected, FNN are one of the two main categories of network architectures in which information flows in a single direction, from the input layer to the output layer, with no feedback connections. An FNN consists of several layers of interconnected neurons, including an input layer, a hidden layer or layers, and an output layer. and its variants, called Convolutional Neural Networks (CNNs). [59]

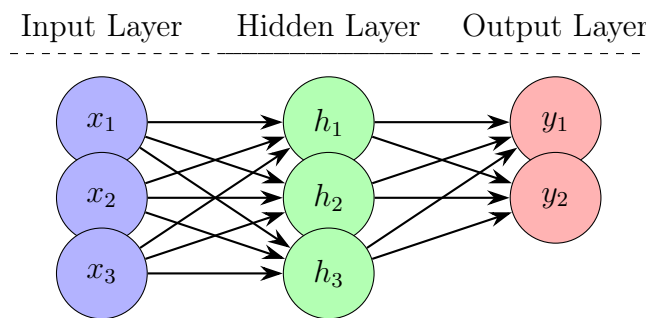


Figure 2.7: Structure of a hidden layer feed-forward neural network

4.2 Convolutional Neural Network

Convolutional Neural Networks (CNN) are multilayer artificial neural networks that currently outperform classical methods such as pattern recognition and image processing. A CNN has a number of layers. These include convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply various filters to the input data, allowing extraction of patterns and features such as edges and corners. The output of the convolution operation is typically fed through a nonlinear activation function and then altered by a pooling function that substitutes a value from nearby outputs for the output at a given location. This pooling function helps create an invariant representation that is insensitive to small input translations. It additionally decreases the total amount of input data. Convolution and pooling layers are stacked hierarchically. Convolution and pooling layers are hierarchically arranged. Features are learned in a stepwise process. After learning the desired training characteristics, their activations are used for input classification in the final layers, which are built from fully connected neurons. [60]

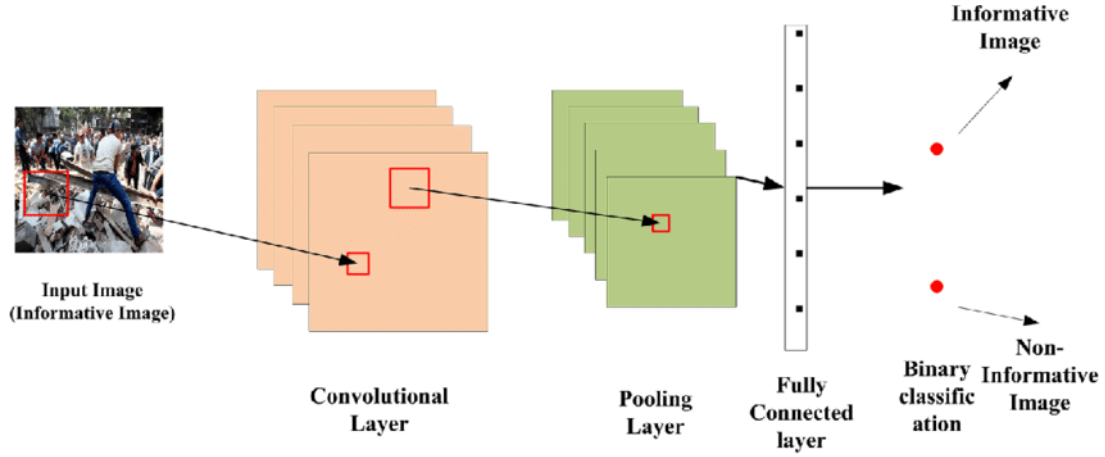


Figure 2.8: CNN Model Architecture [61]

4.3 Deep Recurrent Neural Network

Recurrent Neural Networks (RNNs) are designed to process sequential data where the input is a sequence of vectors and the output is a sequence of vectors. They are defined by recurrence, taking input vectors and previous hidden states, and producing new hidden states and current output vectors. The new hidden state is computed by applying a nonlinear function to a linear combination of the current input and the previous hidden state. RNNs can be trained using backpropagation through time (BPTT), and they have been used in many applications, including natural language processing [62]. The general formula for an RNN at each time step t is:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (2.7)$$

where:

- h_t is the hidden state vector at time t , which summarizes the network's memory of the past inputs up to that point.
- x_t is the input vector at time t .
- W_{hh} is the weight matrix that governs the recurrence of the network, controlling how much of the previous hidden state is retained and how much new information is added.
- W_{xh} is the weight matrix that maps the input vector to the hidden state.
- b_h is the bias vector that is added to the weighted sum of inputs and the previous hidden state.

- f is a non-linear activation function, such as the sigmoid or hyperbolic tangent function, that introduces non-linearity to the network.

The output of the network at time t can be computed as:

$$y_t = W_{hy}h_t + b_y$$

where:

- y_t is the output vector at time t .
- W_{hy} is the weight matrix that maps the hidden state to the output vector.
- b_y is the bias vector that is added to the weighted sum of the hidden state and output vector.

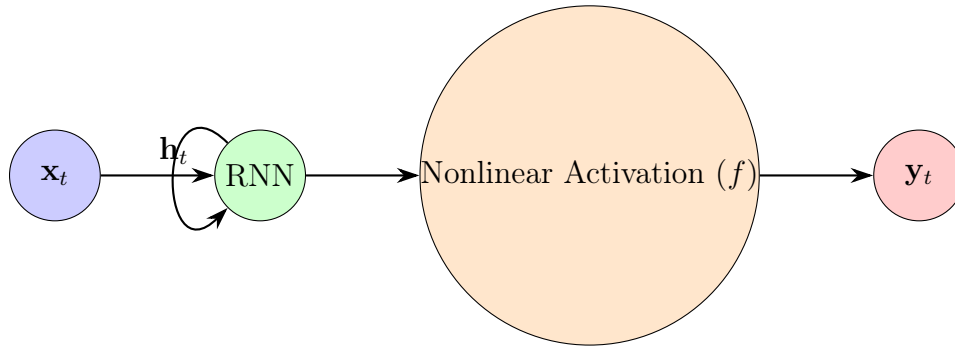


Figure 2.9: Deep Recurrent Neural Network schema

4.3.1 Long Short-Term Memory (LSTM):

Long Short-Term Memory (LSTM) is a type of memory meant to hold long-term dependencies in sequential data. The fundamental idea behind LSTM is to use a memory cell that can store information over an extended period of time, while being able to forget or update that information as needed. [63]

The memory cell:

The memory cell is a fixed size vector updated at each time-step on the basis of the input and the previous state of the LSTM. The update process is controlled by vectors of the same size as the memory cell are the input gate, the forget gate and the output gate. [63]

The Input Gate determines how much of the new input should be added to the memory cell by applying a sigmoid activation function to a linear combination of the input and the previous hidden state, which produces a vector of values between 0 and 1. This vector is then multiplied element-wise by the candidate memory cell vector, which is a vector of the same size as the memory cell obtained by applying the hyperbolic tangent (tanh) activation function to a linear combination of the input and the previous hidden state. [58]

The Output Gate determines how much of the current contents of the memory cell should be output by applying a sigmoid activation function to a linear combination of the input and the previous hidden state, which produces a vector of values between 0 and 1 representing how much of each element of the memory cell should be output, resulting in the final output vector. [58]

The Forget Gates determines how much of the previous contents of the memory cell should be retained by applying a sigmoid activation function to a linear combination of the input and the previous hidden state, which produces a vector of values between 0 and 1 representing how much of each element of the memory cell should be retained, resulting in the elements to be forgotten being set to zero. [58]

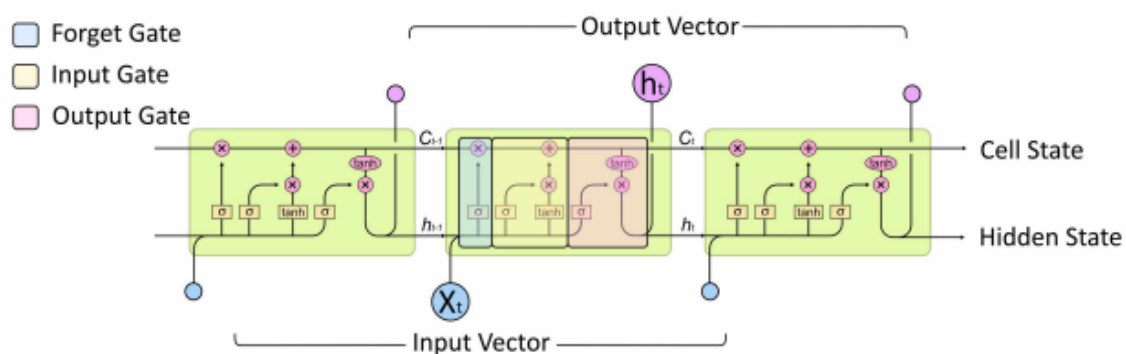


Figure 2.10: LSTM unit with input and output connections [64]

4.3.2 Bidirectional Long Short-Term Memory(BiLSTM)

In a BiLSTM network, the input sequence is processed both in the forward direction and in the backward direction, with separate hidden states for each direction. This allows the network to capture not only the dependencies between the current element and the previous elements, but also the dependencies between the current element and

the future elements [65]. By processing the input in both directions, BiLSTM is able to capture more complex patterns and dependencies in the input data, making it well-suited for tasks that require a deeper understanding of the context, such as named entity recognition and sentiment analysis. One thing to note is that BiLSTM typically requires more computational resources and has more parameters to learn than standard LSTM, which can make it slower and more prone to over-fitting. However, it can often achieve higher accuracy on certain tasks due to its ability to capture bidirectional dependencies in the input data. [66]

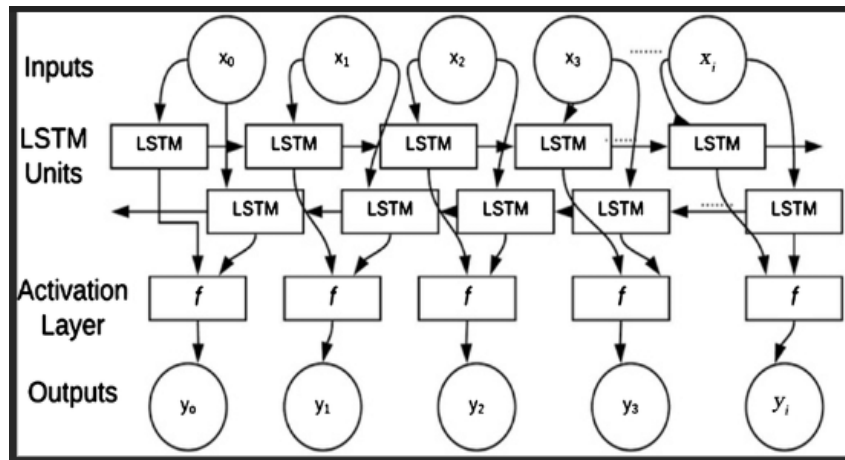


Figure 2.11: BiLSTM model with enabling feature. [67]

5 ML vs DL

Criteria	Machine Learning (ML)	Deep Learning (DL)
Definition	A subset of artificial intelligence (AI) that involves algorithms that learn from data to make decisions or predictions	A subset of machine learning that involves neural networks with multiple layers that can learn and extract features from complex data
Data requirements	Requires less data and can still provide good results	Requires large amounts of data to train models effectively
Feature extraction	Feature extraction is done manually by humans	Automatic feature extraction is performed by deep neural networks
Model complexity	Generally simpler models compared to DL	Models are more complex with many layers and parameters
Performance	Good performance for tasks with smaller amounts of data and fewer features	Performs well on large datasets with many features and can learn complex patterns
Applications	Used in a wide range of applications such as image recognition, spam filtering, and predictive maintenance	Used in applications such as speech recognition, natural language processing, and autonomous vehicles
Training time	Requires less training time compared to DL	Requires significant training time and computational resources

Table 2.1: Comparison between machine learning (ML) and deep learning (DL)

6 Conclusion

In conclusion, this chapter has provided a summary of the basic principles of machine learning and deep learning. We talked about the different classifications of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, and the commonly used algorithms in each category. We also introduced the notion of deep learning and its dominant structure, the artificial neural network. At its core, deep learning involves a great deal of field-specific terminology, with many technical concepts coming from other fields of study. It is important to recognize that common terms can have different meanings in the context of deep learning.

Chapter 3

Environment Tools

1 Introduction

This chapter explores the essential tools and resources required to accomplish our goals. It delves into the hardware and system requirements necessary to conduct comprehensive research in this field.

2 Tools Presentation

2.1 Hardware

The following system requirements were used for running the software tools and libraries in the machine learning experiments:

- Operating System: Windows 10 (64-bit).
- Central Processing Unit (CPU): Intel Core i5-4210U CPU @ 1.70GHz 2.40GHz.
- Random Access Memory(RAM): 8 GB
- Storage: C (145 GB), E (156 GB), and F (163 GB) drives.
- Graphics Card: Multiple NVIDIA GeForce RTX GPUs (GPU 0 and GPU1) with 10 GB of dedicated memory each, which were used to parallelize the training of deep learning models."64-bit Windows 10 operating system.

2.2 System Requirements

2.2.1 Python

Python is a widely used, high-profile programming language known for its simplicity and versatility in a range of applications, from web development to scientific computing. It has become a popular choice for developers worldwide due to its extensive library of modules. [\[68\]](#)

2.2.2 Anaconda Distribution

Anaconda is a free and open-source delivery of the Python programming language which comprises of several pre-installed data science tools and packages including Jupyter Notebook, Spyder IDLE, and renowned data science libraries like NumPy, Pandas, and Scikit-learn. Anaconda streamlines the process of installing and managing these packages, thus enabling data scientists and developers to conveniently establish a uniform and steady

working environment for their projects. It is compatible with Windows, macOS, and Linux operating systems and is extensively employed by the data science community for creating, evaluating, and implementing data-oriented applications. [69]

2.2.2.1 Anaconda Installation

The steps to install Anaconda:

1. Go to the Anaconda website (<https://www.anaconda.com/products/individual>) and download the appropriate installer for your operating system (Windows, macOS, or Linux).
2. Double-click the downloaded file to launch the Anaconda installer.
3. Follow the installer prompts to complete the installation process. You can choose to install Anaconda for a single user or all users on the system. You can also choose the location where you want to install Anaconda.
4. Once the installation is complete, you can launch Anaconda Navigator from the Start menu or by typing “anaconda-navigator” in a command prompt or terminal.

2.2.2.2 Updating Anaconda It’s important to keep Anaconda up-to-date to ensure that you have the latest packages and bug fixes. Follow these steps to update Anaconda:

1. Open the Anaconda Navigator.
2. Click on the “Environments” tab.
3. Select the environment that you want to update. By default, the base environment is selected.
4. Click on the “Update” button on the right-hand side.
5. Click on the “Apply” button to update the selected packages.

2.2.2.3 Python Libraries

- Numerical Python(NumPy)

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. It is a fundamental library for scientific computing in Python. [70]

- Python Data Analysis Library(Pandas)

Pandas is a Python library used for data manipulation and analysis. It offers data structures for efficiently storing large datasets and tools for working with them, including cleaning, merging, and reshaping data. Pandas is widely used in data science and is an essential tool for data cleaning and preparation. [71]

- TensorFlow Is an open source software library for numerical computation and machine learning, developed by the Google Brain team. It provides a flexible platform for building and training machine learning models across a range of tasks, including image and speech recognition, natural language processing, and reinforcement learning. TensorFlow is widely used in academia and industry, and is one of the most popular machine learning libraries in Python. [72]

- Key Enabling Rapid Applicability System (Keras)

Is an open source deep learning framework written in Python. It provides a high-level API(Application Programming Interface) for building and training deep learning models, and can run on top of various backends, including TensorFlow, Microsoft Cognitive Toolkit, and Theano. Keras was designed to be user-friendly, modular, and extensible, and is widely used in industry and academia. [73]

- Scientific Kit for Machine Learning(Scikit-learn)

Is a Python library for machine learning built on top of NumPy, SciPy, and Matplotlib. It provides efficient tools for data mining and data analysis, including classification, regression, clustering, and dimensionality reduction via a consistent interface. Scikit-learn is widely used in academia and industry, and is one of the most popular machine learning libraries in Python. [74]

- MATLAB Plotting Library (Matplotlib)

Matplotlib is a Python library used for creating static, animated, and interactive visualizations in Python. It provides a wide range of plotting functions, from basic line plots to advanced 2D plots, and offers customization options for fine-tuning visualizations. Matplotlib is one of the most widely used data visualization libraries in Python. [75]

- Scientific Python (SciPy)

SciPy is a library for scientific computing in Python that builds upon the NumPy library. It provides a variety of useful functions for optimization, integration, interpolation, linear algebra, signal and image processing, and more. SciPy is widely

used in scientific and engineering applications, and is an essential tool for many scientific computing tasks in Python. [76]

2.2.2.4 Anaconda Libraries Installation There are several methods to install libraries in Anaconda:

1. Using conda: Conda is a package manager that comes with Anaconda. To install a library using conda, open the Anaconda prompt and type:

```
conda install <library_name>
```

For example, to install pandas, type:

```
conda install pandas
```

Conda will automatically resolve dependencies and install the library.

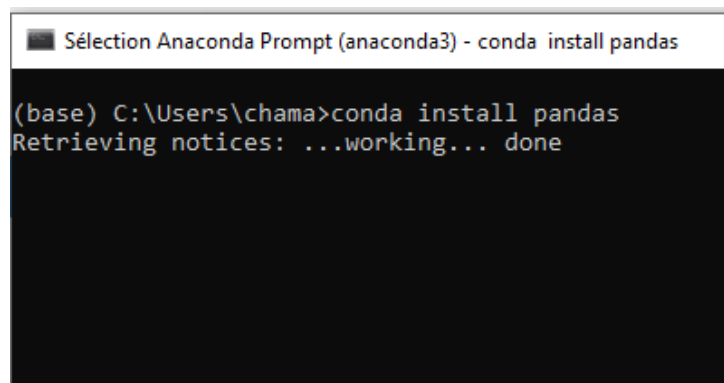


Figure 3.1: conda install pandas

2. Using pip: Pip is a Python package manager that is not installed by default with Anaconda. To install pip, open the Anaconda prompt and type:

```
conda install pip
```

Once pip is installed, you can install libraries using pip by typing:

```
pip install <library_name>
```

For example, to install matplotlib, type:

```
pip install matplotlib
```

```
Anaconda Prompt (anaconda3)

(base) C:\Users\chama>conda install pip
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\chama\anaconda3

  added / updated specs:
    - pip

The following packages will be downloaded:

  package                                     | build                                     |
  -----|-----|
  boltons-23.0.0                             | py39haa95532_0                         | 421 KB
  conda-23.3.1                               | py39haa95532_0                         | 972 KB
  jsonpatch-1.32                             | pyhd3eb1b0_0                           | 15 KB
  jsonpointer-2.1                             | pyhd3eb1b0_0                           | 9 KB
  -----|-----|
                                         Total:                               | 1.4 MB

The following NEW packages will be INSTALLED:

  boltons          pkgs/main/win-64::boltons-23.0.0-py39haa95532_0
  jsonpatch        pkgs/main/noarch::jsonpatch-1.32-pyhd3eb1b0_0
  jsonpointer      pkgs/main/noarch::jsonpointer-2.1-pyhd3eb1b0_0
```

Figure 3.2: install pip

3. Using Anaconda Navigator: Anaconda Navigator is a graphical user interface (GUI) that comes with Anaconda. To install a library using Anaconda Navigator, follow these steps:
 - (1) Open the Navigator and click on the "Environments" tab.
 - (2) Select the environment you want to install the library in.
 - (3) Click on "Channels" and select "conda-forge" from the drop-down menu.
 - (4) Type the name of the library you want to install in the search box and click on the checkbox next to it.
 - (5) Finally, click on the "Apply" button to install the library.

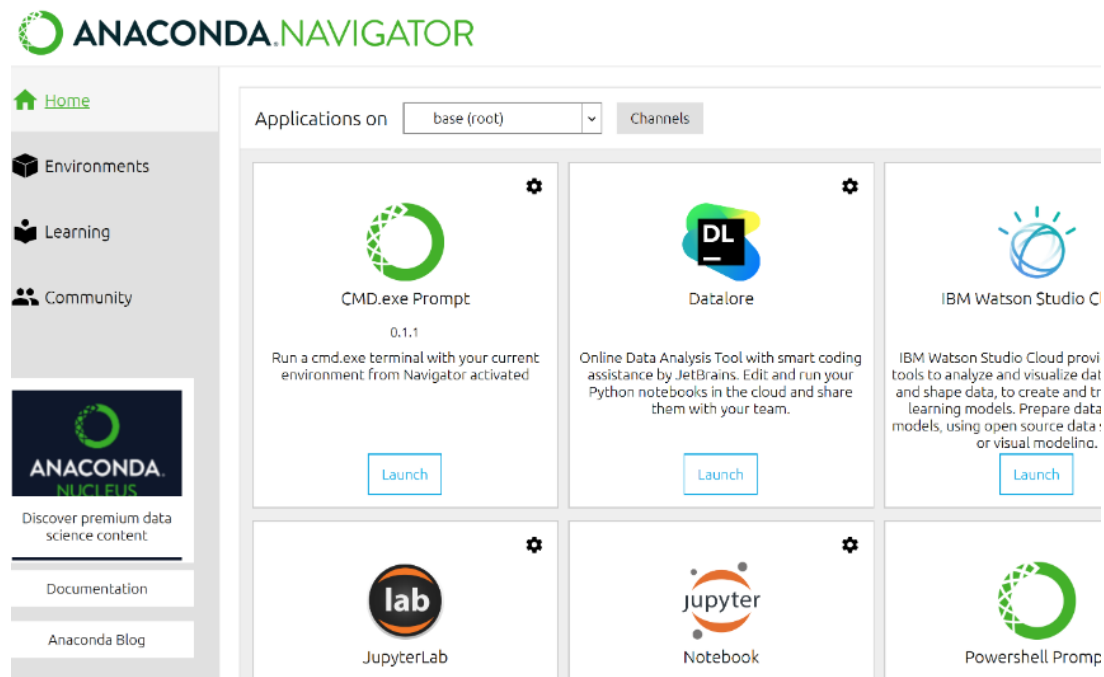


Figure 3.3: Anaconda Navigator

4. Using the Anaconda command prompt: You can also install libraries using the Anaconda command prompt. To do so, follow these steps:

- (1) Open the prompt and type:

```
anaconda search -t conda <library_name>
```

This will show you a list of available versions of the library. Choose the version you want to install and type:

```
conda install -c <channel_name> <library_name>
```

For example, to install numpy version 1.19.3 from the "conda-forge" channel, type:

```
conda install -c conda-forge numpy=1.19.3
```

3 Conclusion

In this chapter, we presented the environment and the tools used to develop machine learning and deep learning models, allowing us to focus on the task and achieve our goals efficiently.

Chapter 4

Realized work and Obtained Results

1 Introduction

This chapter focuses on the experimental part of the work, where the preprocessing of the used dataset is discussed, as well as the proposed model. Additionally, the chapter includes a discussion on the evolution of the model during the training process and the achieved results. Furthermore, the interface application was presented to showcase the practical implementation of the proposed model.

2 Dataset

2.1 Dataset Description

The dataset used for this study consisted of two CSV(Comma Separated Values)files of fake and real news articles, which were obtained from the Kaggle website. The Fake News dataset contained approximately 23,000 labeled news articles that were identified as fake, while the Real News dataset contained approximately 21,000 labeled news articles that were identified as real. Both datasets contains two main columns which are: Title contains the news headlines, Text contains the news, Content/article subject is the type of news, Date is the news published date.

```
1 title,text,subject,date
2 Donald Trump Sends Out Embarrassing New Year's Eve Message; This
3 Drunk Bragging Trump Staffer Started Russian Collusion Investiga
4 Sheriff David Clarke Becomes An Internet Joke For Threatening To
```

Figure 4.1: Dataset Capture

2.2 Dataset Preparation

To prepare the data for the model, the following steps were taken:

- Remove unnecessary columns: The columns 'date' and 'subject' were identified as unnecessary for the model's objective. The 'date' column typically represents the publication date of the news article, which is irrelevant to determining its authenticity. Similarly, the 'subject' column may contain information about the article's topic, but it does not directly contribute to distinguishing between real and fake news. Thus, these columns were removed from both datasets to streamline the data and focus solely on the relevant information.
- Add 'class' column: To enable supervised learning, a new column called 'class' was introduced to both datasets. This column assigns a binary label to each news article,

indicating whether it is real or fake. The addition of this column allows the model to learn from labeled data, enabling it to make predictions on unseen articles based on the patterns it learns during training.

- Visualize data distribution: Bar graphs were created to visualize the distribution of real and fake news articles within the dataset. This visualization helps to gain insights into the balance between the two classes and identify any potential class imbalances.
- Concatenate datasets: The two separate datasets were concatenated into a single unified dataset. This process involved combining the instances from both datasets while maintaining their respective attributes. Concatenating the datasets helps to create a more comprehensive and diverse dataset by pooling information from multiple sources. By incorporating data from different origins, the model can learn from a broader range of examples and potentially improve its ability to generalize to unseen data.
- Drop 'title' column: The decision to drop the 'title' column was based on the scope and focus of the study. Since the primary objective is to classify news articles as real or fake based on their content, the 'title' column was considered extraneous and potentially noisy. Titles are often concise and may not provide sufficient context for accurate classification. Therefore, removing the 'title' column from the concatenated dataset helps to streamline the data, reducing any potential noise and focusing solely on the article's body text.

2.3 Dataset Preprocessing

The following preprocessing steps were taken on the data:

- Text normalization: The textual data within the news articles underwent normalization to ensure consistency and reduce the complexity of the text. This normalization process involved converting all text to lowercase, eliminating any variations in capitalization that might hinder the model's ability to recognize word patterns. Additionally, non-word characters, such as punctuation marks, URLs, and other special symbols, were removed. Furthermore, extra spaces within the text were eliminated to standardize the format and facilitate subsequent processing steps.
- Splitting into training and testing sets: The dataset was divided into separate training and testing sets to assess the model's performance accurately. Typically, a certain percentage of the dataset, such as 90%, is allocated for training the model,

while the remaining percentage, e.g., 10%, is reserved for evaluating the model's performance on unseen data. This separation ensures that the model is not biased towards the examples it has already seen during training and allows for an objective evaluation of its generalization abilities.

- **Tokenization and vocabulary creation:** The training set was tokenized, which involved breaking down the textual data into individual words or tokens. This step is crucial as it converts the text into a format that the model can process. Subsequently, a dictionary or vocabulary of the most frequent words was created based on the training set. This dictionary contains a mapping of words to unique indices, allowing the model to convert text into numerical representations.
- **Converting text to sequences of integers:** Both the training and testing sets were converted into sequences of integers based on the word index obtained from the dictionary. This conversion enables the model to work with numerical data, which is essential for training machine learning models. Each word in the sequences is replaced with its corresponding integer index from the dictionary, allowing the model to process the text as a series of numbers.
- **Padding sequences:** To ensure uniformity in input dimensions, the sequences were padded to have the same length. This step involves adding zeros or truncating the sequences to a maximum length of 256 tokens. Padding is necessary because neural networks typically require fixed-length inputs. By padding or truncating the sequences, all inputs have consistent dimensions, allowing the model to efficiently process them. The maximum length of 256 tokens was chosen based on a suitable trade-off between preserving information and managing computational resources.

3 The Proposed BiLSTM Model

3.1 Build the Model

The model is a Sequential model that consists of four layers. The first layer is an Embedding layer with a maximum vocabulary size of 10,000 and an output dimension of 512. The second and third layers are Bidirectional LSTM layers with 256 and 128 units, respectively. The fourth layer is a Dense layer with 64 units and a ReLU activation function. A Dropout layer with a rate of 0.5 is added after the Dense layer to prevent overfitting. The final layer is a Dense layer with a single output, which represents the predicted class of the input.

Layer (type)	Output Shape	Param #	Activation Function
Embedding	(None, max_len, 512)	39372800	-
Bidirectional LSTM	(None, max_len, 512)	1574912	-
Bidirectional LSTM	(None, 256)	656384	-
Dense	(None, 64)	16448	ReLU
Dropout	(None, 64)	0	-
Dense	(None, 1)	65	-

Table 4.1: Description of the proposed LSTM Model

The previous architecture of the model is summarized using the `model.summary()` method, which provides a concise description of the layers and their output shapes.

```
model.summary()

early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2, restore_best_weights=True)
model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              optimizer=tf.keras.optimizers.Adam(1e-4),
              metrics=['accuracy'])
```

Figure 4.2: Model Architecture

3.1.1 Train the Model

During training, the model used the BinaryCrossentropy loss function and the Adam optimizer with a learning rate of $1e-4$. The model was trained for 5 epochs with a batch size of 30 and a validation split of 0.1. Early stopping was used with a patience of 2 to prevent overfitting and restore the best weights.

The history object of the trained model contains information about the training and validation accuracy and loss for each epoch.

```
history = model.fit(X_train, y_train, epochs=5, validation_split=0.1, batch_size=30, shuffle=True, callbacks=[early_stop])
```

Figure 4.3: Model Training

3.2 The Obtained Results

The model was trained on the preprocessed dataset, and its performance was evaluated using two main metrics: accuracy and loss.

The training and validation accuracy were plotted over the number of epochs to provide an overview of the model's performance during training. Similarly, the training and validation loss were also plotted over the number of epochs to assess the model's ability to minimize the loss function.

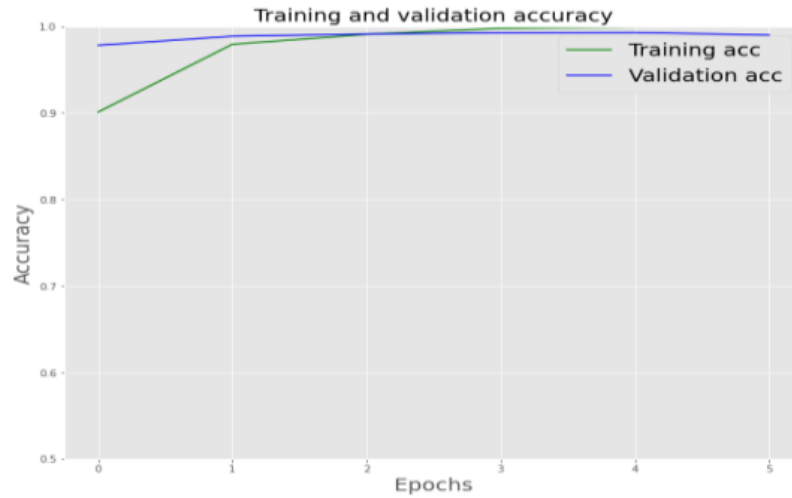


Figure 4.4: Training and validation accuracy

The first graph illustrates the training and validation accuracy of our model over multiple epochs. The training accuracy represents the model's ability to correctly classify the training data, while the validation accuracy measures its performance on unseen data from the validation set. In epoch 0, the training accuracy is reported to be 90%. As the model undergoes training, the accuracy steadily improves and reaches 99% by epoch 5. This signifies that the model becomes increasingly proficient at accurately classifying the training data as the training progresses.

Similarly, the validation accuracy starts at 98% in epoch 0 and also reaches 99% by epoch 5. This indicates that the model's performance on previously unseen data, represented by the validation set, closely aligns with its performance on the training data. The validation accuracy provides an estimate of how well the model is expected to perform on new, unseen data.



Figure 4.5: Training and validation loss

The second graph presents the training and validation loss values of the BiLSTM model over the course of training. The training loss represents the average discrepancy between the model's predicted outputs and the true labels for the training data, while the validation loss quantifies the model's performance on the validation set. In epoch 0, the training loss is reported to be 0.200. This indicates that, on average, the model's predictions deviate from the true labels by a value of 0.200. As the model undergoes training, the training loss steadily decreases and reaches 0.010 by epoch 5. A lower training loss suggests that the model is effectively minimizing the differences between its predictions and the true labels for the training data.

Regarding the validation loss, it starts at 0.073 in epoch 0 and decreases to 0.025 in epochs 2 and 3, indicating that the model performs well on the validation set during these epochs. However, by epoch 5, the validation loss increases to 0.051. This slight increase suggests that the model may be starting to overfit the training data, as its performance on the validation set begins to degrade.

3.3 The Evaluation of the Model

- **Accuracy:** is a measure used to evaluate the efficiency with which a classification model performs. It denotes the proportion of correctly classified instances among all instances [77]. Accuracy is typically calculated using the following equation:

$$\text{Accuracy} = \frac{\text{Number of correctly classified instances}}{\text{Total number of instances}}$$

The resulting value is expressed as a percentage, ranging from 0% to 100%. Higher

accuracy indicates better classification performance, with 100% accuracy representing perfect predictions.

- **Precision:** is a metric that focuses on the ratio of correct predicted positive instances to the total number of predicted positive ones. It assesses the ability of the model to minimize false positive errors [78]. Precision is calculated using the formula :

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Loss:** It quantifies the discrepancy between predicted and true values. The loss function maps parameter values to scalar values, indicating how well the parameters perform. Minimizing the loss improves model performance [79]. The mean squared error (MSE) is a common loss function:

$$\text{Loss} = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

where N is the number of training examples, y_i is the true value, and $\bar{y}_i$ is the predicted value.

- **Recall:** is a metric used to assess the accuracy of the model's performance in identifying all positive occurrences among all [78]. Recall is calculated using the following formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

A high recall score indicates that the model is efficient at correctly identifying positive instances.

- **Adam Optimizer:** a technique for improving the performance of deep learning. It is a form of stochastic gradient descent (SGD) with built-in flexible learning rates. of the popular optimization algorithms in deep learning. [80]
- **F1-score:** is a balanced average of precision and recall, commonly used in binary categorization tasks. It is a neutral method for the evaluation of precision and recall. [78]
- **Epoch:** it refers to the frequency with which the full dataset is fed through the model during the training process. [81]

- **Batch size:** is a parameter that determines the number of data instances that are processed together in each iteration during training of a deep learning model. Rather than having to use the complete training data set. [81]
- **Batch normalization:** is a technique applied in deep learning for normalizing the training data in each mini-batch. it is used to accelerate training and improve model performance. It is typically applied after the activation function in a neural network. [82]

4 Comparing Obtained Results of ML Models with BiLSTM model

we assessed the accuracies of several machine learning (ML) algorithms. The ML models, including Logistic Regression (LR), Decision Trees (DT), and K-Nearest Neighbors (KNN), were trained and tested on our dataset. The following table summarizes the accuracies achieved by each ML algorithm in our study.

Algorithm	Accuracy
LR	100%
DT	100%
KNN	84.04%
BiLSTM	99%

Table 4.2: Accuracy Comparison of Machine Learning Models

Logistic Regression (LR): The LR model achieves an accuracy of 100%. This means that the LR model is able to correctly classify all instances in the dataset.

Decision Tree (DT): The DT model also achieves an accuracy of 100%. This indicates that the DT model is able to accurately classify all instances in the dataset, similar to the LR model.

k-Nearest Neighbors (KNN): The KNN model achieves an accuracy of 84.04%. This means that the KNN model is able to correctly classify approximately 84.04

Bidirectional LSTM (BiLSTM): The BiLSTM model achieves an accuracy of 99%. This indicates that the BiLSTM model is able to accurately classify 99% of the instances in the dataset.

In summary, based on the provided accuracy values, both the LR and DT models achieve perfect accuracy, correctly classifying all instances in the dataset. The BiLSTM model achieves a high accuracy of 99%, indicating its strong performance in accurately classifying the dataset. The KNN model achieves an accuracy of 84.04%, which is lower compared to the other models.

It is important to note that accuracy alone may not provide a complete picture of the models' performance, and it is advisable to consider other evaluation metrics and potentially perform statistical significance tests to make a more comprehensive comparison.

5 Application Interface

The application consists of three main pages: the Home Page, the Tools Page, and the About Page.

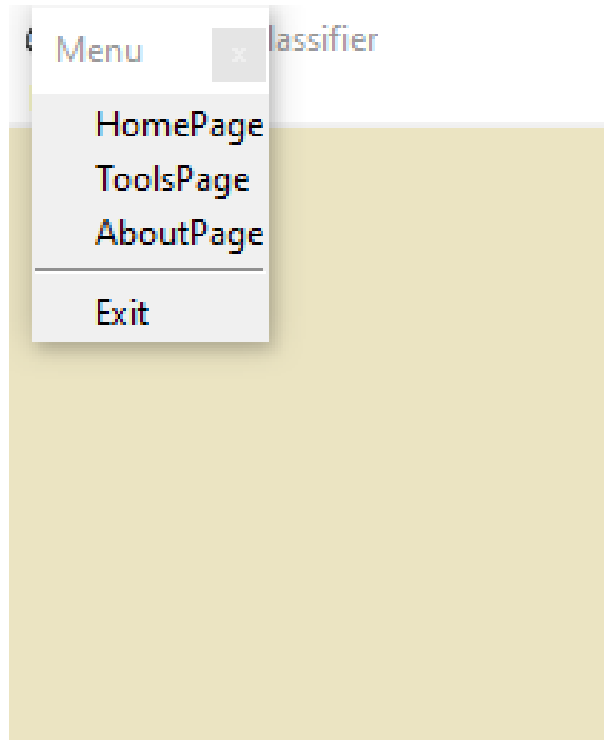


Figure 4.6: Capture of the Menu of the application interface

5.1 Home Page

The Home Page is the main page of the application, where users can input text or URLs for analysis and classification. The interface elements on this page include:

- A message frame displaying a prompt for users to upload or enter the article news they want to classify.
- Three buttons: "Upload," "Enter," and "Classify." The "Upload" button allows users to upload an article file, the "Enter" button displays a text box for manual entry of the article, and the "Classify" button triggers the article classification process.

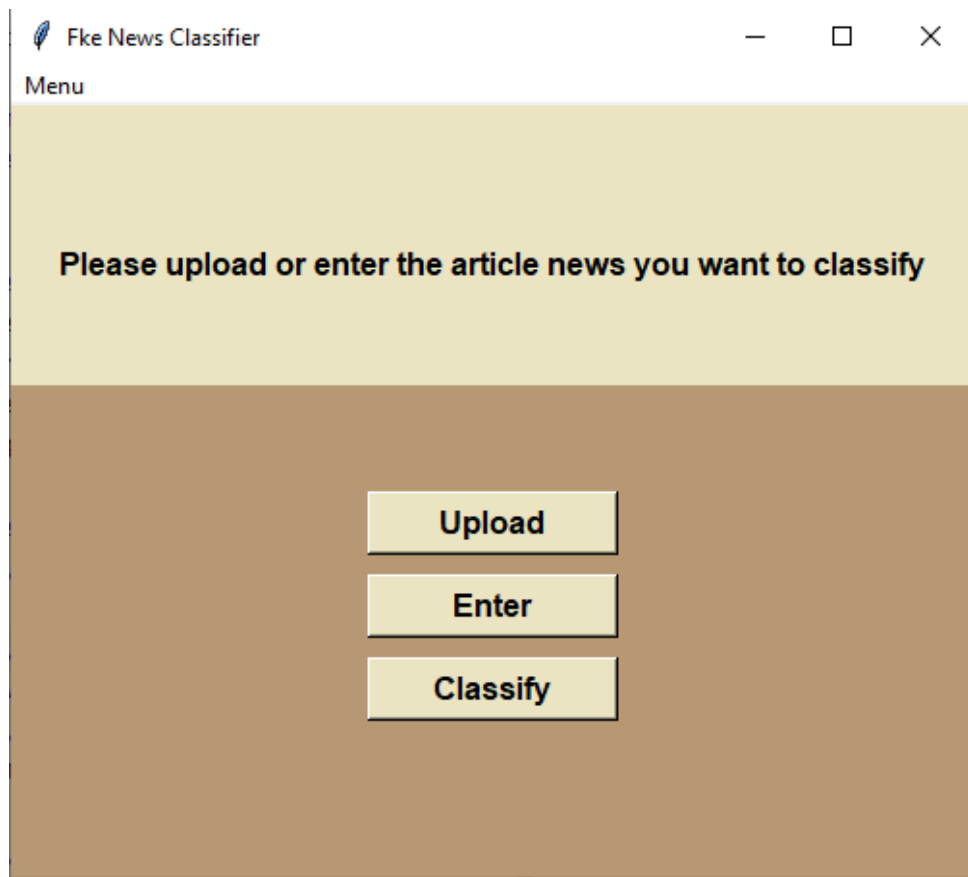


Figure 4.7: Capture of the Home Page interface

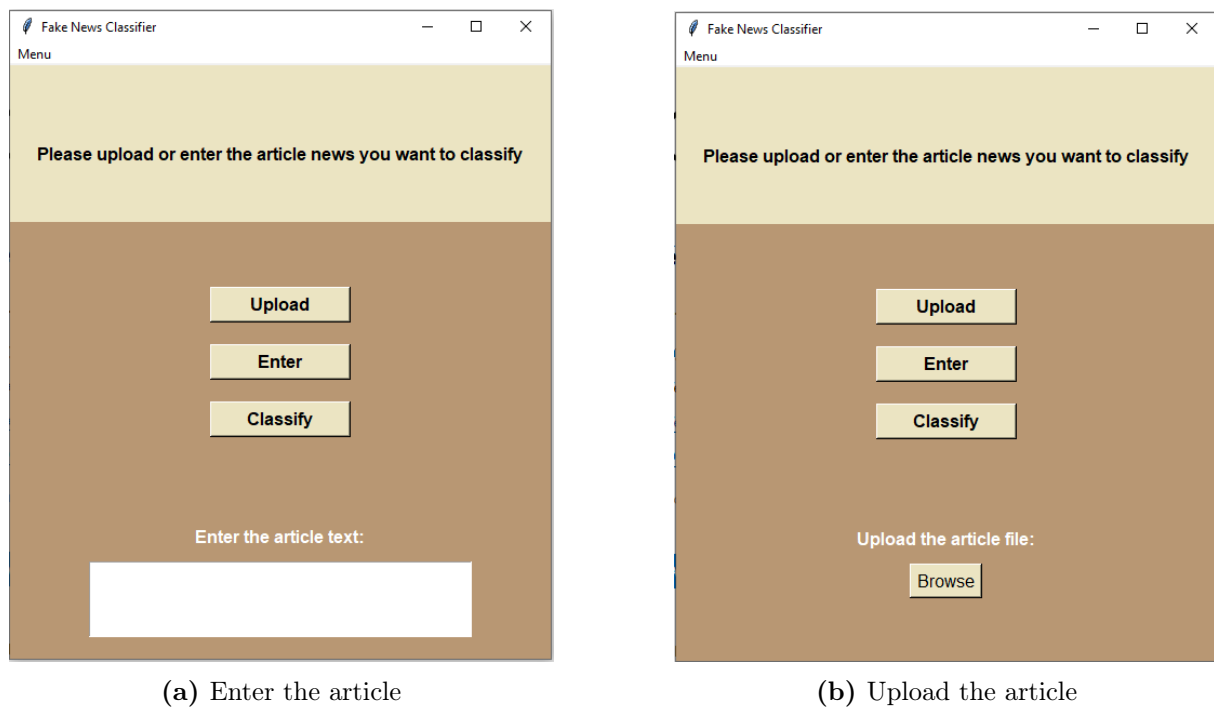


Figure 4.8: Capture of "Enter" and "Upload" functions

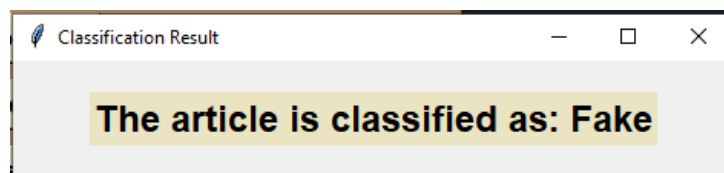


Figure 4.9: Capture classifying the article as fake

5.2 Tools Page

The Tools Page showcases the various technologies used in building the application. It includes the following information:

- BiLSTM Model: This section provides details about the model used in the application, such as its accuracy in training and testing data.
- The Used Dataset: This section describes the dataset used in the application and provides a link to download it.
- Environment Tool: This section provides a link to download Anaconda.

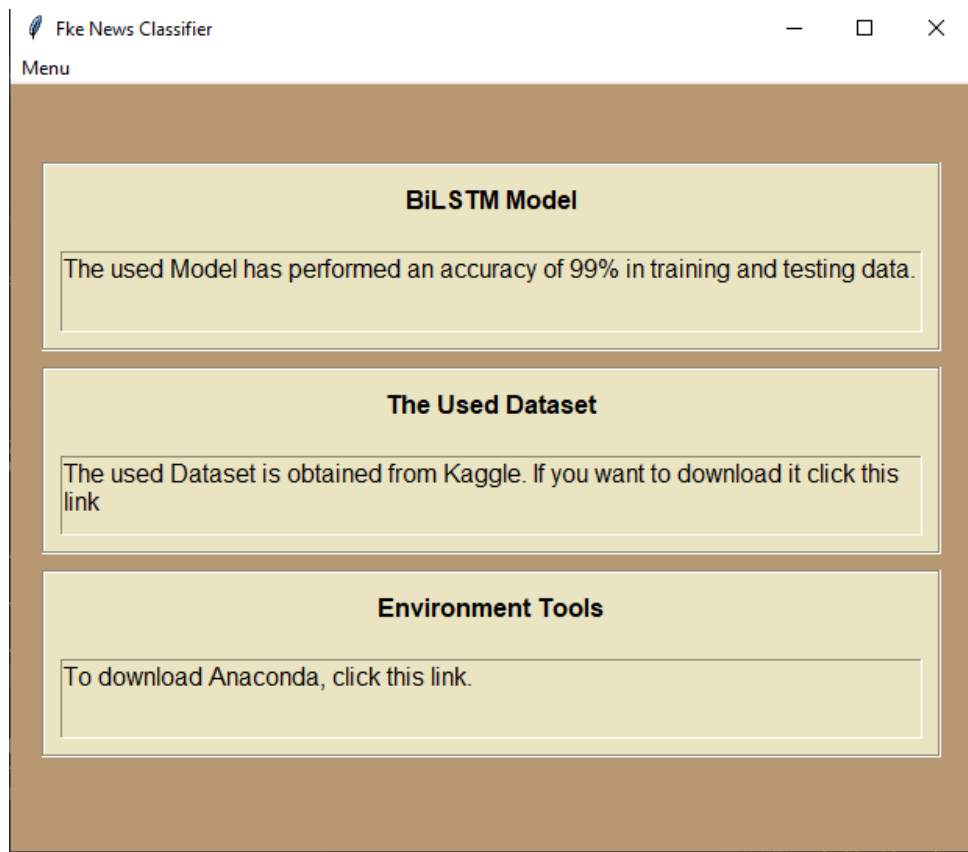


Figure 4.10: Capture of the Tools Page interface

5.3 About Page

The About Page provides an overview of the application and its purpose. It includes the following information:

- The application's goal: To classify news articles as real or fake using a BiLSTM model.
- The user-friendly interface: Built using Tkinter to provide a seamless experience.
- Application pages: Home Page, Tools Page, and About Page.
- The aim: Combating misinformation and empowering individuals to make informed decisions about the news they consume.

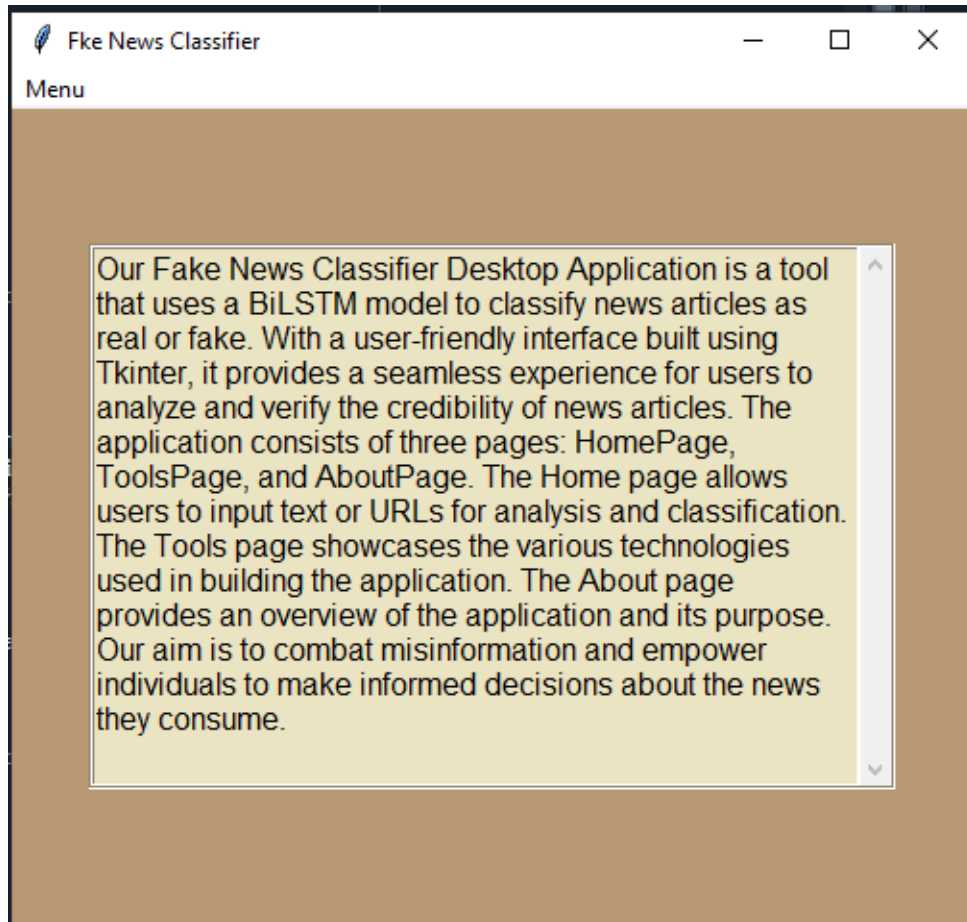


Figure 4.11: Screenshot of the About Page interface

6 Discussion

Our experimental results showcase the outstanding performance of the proposed BiLSTM model, which achieved an impressive accuracy of 99% on both the training and validation sets. This indicates the model's strong capability to effectively discern between real and fake news articles.

Furthermore, the ML models, including LR, DT, and KNN, exhibited remarkable accuracies of 100%, emphasizing their effectiveness in our study.

7 Conclusion

This chapter encompasses two key components. The first focuses on the construction of the BiLSTM model and its comparison with ML models, while the second pertains to the development of the application interface. Furthermore, we discuss the dataset preparation and preprocessing steps undertaken, outline the proposed model, present the obtained results, and provide definitions for important concepts. Finally, we showcase our application interface.

General Conclusion

In Our work, we focused on the crucial task of fake news detection using machine learning and deep learning algorithms, with a specific emphasis on employing a BiLSTM model. Our research aimed to contribute to the ongoing efforts in combatting the spread of misinformation in the digital age.

Through extensive experimentation and analysis, we have achieved promising results in fake news detection. Our findings demonstrate the effectiveness of the BiLSTM model in capturing the contextual dependencies and semantic patterns present in textual data, enabling accurate classification between fake and real news articles.

The evaluation of our model on the datasets revealed its ability to generalize well and outperform several baseline classifiers. By leveraging the power of deep learning and sequential modeling, our BiLSTM model successfully captured the nuanced linguistic cues and temporal dependencies that are often indicative of fake news.

However, it is essential to acknowledge the limitations of our study. The performance of our model may be influenced by factors such as dataset bias, the evolving nature of fake news techniques, and the generalizability of the model to different languages and domains. These limitations open up avenues for future research to address these challenges and further improve the effectiveness of fake news detection systems.

As the battle against fake news continues to evolve, we hope that this research serves as a stepping stone for further advancements in the field. By building upon our work and exploring innovative approaches, we can develop more robust and efficient fake news detection systems, ultimately contributing to a more trustworthy and informed information ecosystem in the digital era.

Bibliography

- [1] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483, 2020.
- [2] G. Said, O.M. Sara, H. Khadidja, and C. Safia. An efficient system to predict customers' satisfaction on touristic services using ml and dl approaches. In *22nd Arabic International Conference on Information Technology (ACIT 2021)*, pages 1–5, Quabos University, Sultanate Oman, December 2021. IEEE.
- [3] L. Borges, B. Martins, and P. Calado. Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *Journal of Data and Information Quality (JDIQ)*, 11(3):14, 2020.
- [4] S. Gadri and N.E. Adouane. Efficient traffic signs recognition based on cnn model for self-driving cars. In P. Vasant, I. Zelinka, and G.W. Weber, editors, *Intelligent Computing & Optimization*, volume 371 of *Lecture Notes in Networks and Systems*, Cham, December 2022. Springer.
- [5] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.
- [6] S. Gadri, S. Chabira, S. Ould Mehieddine, and Kh. Herizi. Sentiment analysis: Developing an efficient model based on machine learning and deep learning approaches. In P. Vasant, I. Zelinka, and G.W. Weber, editors, *Intelligent Computing & Optimization*, volume 371 of *Lecture Notes in Networks and Systems*, Cham, December 2022. Springer.
- [7] Chunqiu Zhang, Alok Gupta, Christian Kauten, Amit V. Deokar, and Xiaolin Qin. Detecting fake news for reducing misinformation risks using analytics approaches. *European Journal of Operational Research*, 279(3):1036–1052, 2019.

- [8] Javier Monti, Pablo Castells, and Francisco Rodríguez. Convolutional neural networks for stance detection on fake news. In *Proceedings of the 12th International Conference on Web Search and Data Mining*, pages 755–763. ACM, 2019.
- [9] G. Said. Efficient arabic handwritten character recognition based on machine learning and deep learning approaches. *Journal of Advanced Research in Dynamical & Control Systems*, 12(07-Special Issue):9–17, August 2020. (Scopus indexed Journal).
- [10] G. Said. Developing an efficient predictive model based on ml and dl approaches to detect diabetes. *The International Journal of Computing and Informatics Informatica*, 45(3), 2021.
- [11] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 112(18):5552–5557, 2015.
- [12] Michael D Conover, Jacob Ratkiewicz, Matei Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. *Proceedings of the fifth international AAAI conference on weblogs and social media*, 133:89–96, 2011.
- [13] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. Rumor cascades. In *Proceedings of the eighth international conference on weblogs and social media*, pages 101–110, 2014.
- [14] Gordon Pennycook, Adam Bear, Evan T Collins, and David G Rand. The implied truth effect: Attaching warnings to a subset of fake news headlines increases perceived accuracy of headlines without warnings. *Management science*, 66(11):4944–4957, 2020.
- [15] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018.
- [16] Liang Wu, Fred Morstatter, Kathleen M Carley, and Huan Liu. Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD explorations newsletter*, 21(2):80–90, 2019.

- [17] Edson C Tandoc Jr, Zheng Wei Lim, and Richard Ling. Defining “fake news” a typology of scholarly definitions. *Digital journalism*, 6(2):137–153, 2018.
- [18] Yimin Chen, Niall J Conroy, and Victoria L Rubin. Misleading online content: recognizing clickbait as "false news". In *Proceedings of the 2015 ACM on workshop on multimodal deception detection*, pages 15–19, 2015.
- [19] Gordon Pennycook and David G Rand. Lazy, not biased: Susceptibility to partisan fake news is better explained by lack of reasoning than by motivated reasoning. *Cognition*, 188:39–50, 2019.
- [20] Jeannette Paschen. Investigating the emotional appeal of fake news using artificial intelligence and human contributions. *Journal of Product & Brand Management*, 29(2):223–233, 2020.
- [21] John M MacKenzie. Propaganda and empire: the manipulation of british public opinion, 1880–1960. In *Propaganda and Empire*. Manchester University Press, 2017.
- [22] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017.
- [23] Eytan Bakshy, Solomon Messing, and Lada A. Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.
- [24] Stephan Lewandowsky, Ullrich K. Ecker, and John Cook. Beyond misinformation: Understanding and coping with the "post-truth" era. *Journal of Applied Research in Memory and Cognition*, 6(4):353–369, 2017.
- [25] Gordon Pennycook and David G. Rand. The implied truth effect: Attaching warnings to a subset of fake news stories increases perceived accuracy of stories without warnings. *Management Science*, 66(11):4944–4957, 2019.
- [26] Kai Shu et al. Fake news detection (fnd). *Journal of Information Science*, 2017.
- [27] Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. Detecting fake news using machine learning: A systematic literature review. *arXiv preprint arXiv:2102.04458*, 2021.
- [28] Martin J. Eppler and Jeanne Mengis. Information overload: A systematic literature review. *International Journal of Information Management*, 24(5):431–444, 2004.

Bibliography

- [29] Gordon Pennycook, Tyrone D Cannon, and David G Rand. The implied truth effect: Attaching warnings to a subset of fake news stories increases perceived accuracy of stories without warnings. *Management Science*, 66(11):4944–4957, 2020.
- [30] Hugo Mercier and Dan Sperber. Why do humans reason? arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34(2):57–111, 2011.
- [31] Karine Nahon and Jeff Hemsley. The disinformation ecosystem: A structural analysis. *Big Data & Society*, 6(2), 2019.
- [32] Wlodzislaw Duch, Rudy Setiono, and Jacek M Zurada. Computational intelligence methods for rule-based data understanding. *Proceedings of the IEEE*, 92(5):771–805, 2004.
- [33] Rajshree S Dubey, Rajnish Choubey, and Joy Bhattacharjee. Multi feature content based image retrieval. *International Journal on Computer Science and Engineering*, 2(6):2145–2149, 2010.
- [34] Saud Althabiti, Mohammad Ammar Alsalka, and Eric Atwell. Detecting arabic fake news on social media using sarcasm and hate speech in comments.
- [35] Nihel Fatima Baarir and Abdelhamid Djeflal. Fake news detection using machine learning. In *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, pages 125–130. IEEE, 2021.
- [36] Bradley C Love. Comparing supervised and unsupervised category learning. *Psychonomic bulletin & review*, 9(4):829–835, 2002.
- [37] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*.*[Internet]*, 9:381–386, 2020.
- [38] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3–4):197–387, 2014.
- [39] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.
- [40] Machine learning types. Figure source: <https://www.educba.com/types-of-machine-learning-algorithms/>. Accessed on 20/04/2023.
- [41] Daniel Jurafsky and James H. Martin. Chapter 5: Logistic regression. Retrieved from <https://www.web.stanford.edu/~jurafsky/slp3/5.pdf>, 2023.

Bibliography

- [42] Decision tree. Figure source: <https://blog.davidvassallo.me/2019/08/06/3-uses-for-random-decision-trees-forests-you-maybe-didnt-know-about/>. Accessed on 20/04/2023.
- [43] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.
- [44] Logistic regression example. Figure source: <https://blog.goodaudience.com/machine-learning-using-logistic-regression-in-python-with-code-ab3c7f5f3bed?gi=57a35733588e>. Accessed on 20/04/2023.
- [45] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.
- [46] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.
- [47] K-nearest neighbors (knn) model. Figure source: <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>. Accessed on 20/04/2023.
- [48] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin, 2008.
- [49] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1476–1485, 2019.
- [50] Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. Semi-supervised learning with multi-head co-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6278–6286, 2022.
- [51] Ximing Li, Yuanzhi Jiang, Changchun Li, Yiyuan Wang, and Jihong Ouyang. Learning with partial labels from semi-supervised perspective. *arXiv preprint arXiv:2211.13655*, 2022.
- [52] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [53] Reinforcement in ml. Figure source: <https://techvidvan.com/tutorials/reinforcement-learning/>, Year. Accessed on 20/04/2023.

Bibliography

- [54] S Agatonovic-Kustrin and Rosemary Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5):717–727, 2000.
- [55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [56] Activation function types. Figure source: <https://analyticsindiamag.com/what-are-activation-functions-and-when-to-use-them/>. Accessed on 20/04/2023.
- [57] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.
- [58] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [59] Murat H Sazli. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 50(01), 2006.
- [60] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [61] The basic CNN model for image classification. 10,06,2023.
- [62] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.
- [63] Pritika Bahad, Preeti Saxena, and Raj Kamal. Fake news detection using bi-directional lstm-recurrent neural network. *Procedia Computer Science*, 165:74–82, 2019.
- [64] Lstm unit with input and output connections. Figure source: https://www.researchgate.net/publication/349271188_Understanding_LSTM_Network_Behaviour_of_IMU-Based_Locomotion_Mode_Recognition_for_Applications_in_Prostheses_and_Wearables. Accessed on 20/04/2023.

Bibliography

- [65] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [66] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.
- [67] Bilstm model with enabling feature. Figure source: https://www.researchgate.net/figure/BiLSTM-model-with-activation-function_fig2_340896412. Accessed on 20/04/2023.
- [68] Raphael Vallat. Pingouin: statistics in python. *J. Open Source Softw.*, 3(31):1026, 2018.
- [69] Hetu Bhavsar and Richa Manglani. Sentiment analysis of twitter data using python. *International Research Journal of Engineering and Technology (IRJET)*, 6(3):510–511, 2019.
- [70] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.
- [71] Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.
- [72] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.
- [73] Daniele Grattarola and Cesare Alippi. Graph neural networks in tensorflow and keras with spektral [application notes]. *IEEE Computational Intelligence Magazine*, 16(1):99–106, 2021.
- [74] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [75] Paul Barrett, John Hunter, J Todd Miller, J-C Hsu, and Perry Greenfield. matplotlib—a portable python plotting package. In *Astronomical data analysis software and systems XIV*, volume 347, page 91, 2005.

- [76] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.
- [77] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
- [78] Marina Sokolova and Guy Lapalme. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 977–984. ACM, 2006.
- [79] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.
- [80] Ange Tato and Roger Nkambou. Improving adam optimizer. 2018.
- [81] Jason Brownlee. What is the difference between a batch and an epoch in a neural network. *Machine Learning Mastery*, 20, 2018.
- [82] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

ملخص

أصبح الكشف عن الأخبار المزيفة تحديا حاسما في العصر الرقمي، حيث تنتشر المعلومات بسرعة عبر منصات الإنترنت المختلفة. يركز هذا العمل على تطبيق تقنيات التعلم الآلي والتعلم العميق وتطوير نموذج لكشف الأخبار المزيفة يعتمد على الشبكات العصبية المتكررة. هدف هذا النموذج هو تحديد وتصنيف الأخبار المزيفة بدقة. من خلال تجربتنا، أكدنا فعالية ودقة هذا النموذج في التمييز بين الأخبار المزيفة والمعلومات الحقيقية. **الكلمات المفتاحية:** كشف الأخبار المزيفة، التعلم العميق، التعلم الآلي، الشبكات العصبية الاصطناعية، الشبكة العصبية المتكررة

Abstract

Detecting fake news has become a critical challenge in the digital era, where information spreads rapidly through various online platforms. This work focuses on the development and application of machine learning and deep learning techniques and develops a recurrent neural network (RNN)-based fake news detection model. The goal of this model is to accurately identify and classify fake news. Through our experimental work, we have demonstrated the high effectiveness and accuracy of this model in distinguishing fake news from genuine information.

Keywords: Fake News Detection, Deep Learning, Machine Learning, Artificial Neural Networks, Recurrent Neural Network.

Résumé

La détection des fausses informations est devenue un défi critique à l'ère numérique, où l'information se propage rapidement à travers diverses plateformes en ligne. Ce travail se concentre sur le développement de techniques d'apprentissage automatique et d'apprentissage profond et développer un modèle basé sur les réseaux neuronaux récurrents (RNN). L'objectif de ce modèle est d'identifier et de classer avec précision les fausses informations. À travers notre expérimentation, nous avons démontré l'efficacité élevée et l'exactitude de ce modèle.

Mots Clés: Détection des fausses informations, Apprentissage profond, Apprentissage automatique, Réseaux de neurones artificiels, Réseau de neurones récurrents.