Kurdistan Region Government/Iraq
Presidency of the Council of Ministers
Ministry of Higher Education & Scientific
Research Erbil Polytechnic University
Technical Engineering College
Information Systems Engineering Department

Enhancing Clickbait Detection Through Deep Learning and Language-specific Analysis in English and Kurdish

A Thesis

Submitted to the Council of the College of Technical College at Erbil Polytechnic University in Partial Fulfillment of the Requirements for the Degree of Master of Information Systems Engineering

By

Ibrahim Shamal Abdul Khaleq

B.Sc. Information Systems Engineering

Supervised by

Assist. Prof. Dr. Shahab Wahhab Kareem

Erbil, Kurdistan

September 2023

DECLARATION

I declare that the Master Thesis entitled: "Enhancing Clickbait Detection Through Deep Learning and Language-specific Analysis in English and Kurdish "is my own original work, and hereby I certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

Signature:

Student Name: Ibrahim Shamal Abdul Khaleq

Date:

CERTIFICATION OF PROOFREADING

This is to certify that this dissertation entitled "Enhancing Clickbait Detection Through Deep Learning and Language-specific Analysis in English and Kurdish" written by the postgraduate student (Ibrahim Shamal Abdul Khaleq) has been proofread and checked for grammatical. punctuation, and spelling mistakes. Therefore, after making all the required corrections by the student for further improvement. I confirm that this last copy of the dissertation is ready for submission.

Signature:

Name: Yassin Muhamad Awla

Lecturer, Applied linguistics.

Phone No.0750 488 2578

Email address: Yassin.muhamad.eng@gmail.com

Date: 17-09-2023

SUPERVISOR CERTIFICATE

This thesis has been written under my supervision and has been submitted for the award of the degree of Master of Information Systems Engineering with my approval as supervisor.

Signature:

Name: **Assist. Prof. Dr. Shahab Wahhab Kareem**

Date: /11/2023

I confirm that all requirements have been fulfilled.

Signature:

Name:

Head of the Department of Information Systems Engineering

Date:

I confirm that all requirements have been fulfilled.

Postgraduate Office

Signature:

Name:

Date:

EXAMINING COMMITTEE CERTIFICATION

We certify that we have read this thesis: "Enhancing Clickbait Detection Through Deep Learning and Language-specific Analysis in English and Kurdish" and as an examining committee examined the student (Ibrahim Shamal Abdul Khaleq) in its content and what related to it. We approve that it meets the standards of a thesis for the degree of Master in Information Systems Engineering.

Signature:	Signature:
Name: **Assist. Prof. Dr. Roojwan Scddeek**	Name: **Assist. Prof. Dr. Masoud Muhammed**
Member	Member
Date: / /2023	Date: / /2023
Signature:	Signature:
Name: **Assist. Prof. Dr. Shahab Wahab**	Name: **Assist. Prof. Dr. Abbas Mohamad**
Supervisor and member	Chairman
Date: / /2023	Date: / /2023

Approved by:

Dean of the College of Technical Engineering

Name: **Prof. Dr. Ayad Zaki Sabr**

Signature:

Date: / /2023

ACKNOWLEDGEMENTS

In the name of Allah, the Merciful, the Compassionate. We are grateful to Almighty Allah for supporting us in order to complete the requirement of this thesis. Many people have contributed in the completion of this work.

I am thankful to Dr. Shahab Wahab for being the best guide and advisor for this research work in every field I have taken to complete my requirement. His ideas and inspirations have helped me make this idea of mine into a fully-fledged project. Without presence of him I may never had tasted the flavor in a research work.

Again, I am thankful to my batch-mates to support me in my implementation part sometime. I am also grateful to all the professors of my department for being a constant source of inspiration and motivation during the course of the project.

Lastly, I thank my parents for their support and guidance to reach this far and encouraging me throughout this work.

DEDICATION

This thesis is dedicated to Almighty Allah

Asking for acceptance and this will be a good work for fellow scholars and researchers.

To all people who were believing in me whom never stopped believing

To my brother who supported me and helped me to accomplish this study

To my mother who raised me and guided me throughout my whole life never stopped supporting me.

ABSTRACT

In the rapidly evolving landscape of online content, the prevalence of clickbait poses a significant challenge for users seeking reliable and informative material. Clickbait, characterized by sensationalized headlines designed to attract attention and drive user engagement, has become a pervasive issue in various languages and cultural contexts. As digital platforms continue to host a vast array of content, the need for robust clickbait detection mechanisms becomes paramount to ensure a trustworthy online experience. This study aims to evaluate the performance of deep neural networks in clickbait detection for both English and Kurdish languages. To address clickbait in Kurdish, we collected 10,000 news articles from various Kurdish platforms, complemented by a dataset of 32,000 English headlines curated by Chakraborty. Utilizing Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BILSTM), Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and a hybrid CNN BILSTM model, we evaluated clickbait detection techniques. Findings underscore the importance of understanding language-specific traits and cultural norms in spotting clickbait across linguistic boundaries. The Bidirectional Long Short-Term Memory algorithm proved most effective in English, boasting a 99.23% accuracy rate, 95.33% precision, 94.33% recall rate, and a 95% F1 score. In Kurdish, the Gated Recurrent Unit algorithm excelled with a 93.93% accuracy rate, 93.13% precision, 95.17% recall rate, and a 94.13% F1 score. This study extends the application of recurrent neural network and deep learning methods in clickbait detection, showcasing their potential in analyzing textual data with nuanced semantic features, contributing valuable insights to the broader field of natural language processing.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Acronyms
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
CBCNN	Clickbait Convolutional Neural Network
CBFs	Content-Based Features
CBR	Case-Based Reasoning
CNN	Convolutional Neural Network
ELMO	Embeddings from Language Models
FOMO	Fear of Missing Out
GloVe	Global Vectors for Word Representation
GPT	Generative Pre-trained Transformer
GPUs	Graphics Processing Units
GRU	Gated Recurrent Unit
IR	Information Retrieval
LRP	Layer-Wise Relevance Propagation
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
N-gram	A sequence of N words
NLP	Natural Language Processing
OLSTM	Ontology-based LSTM Model
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SVM	Support Vector Machines
TDM	Term-Document Matrix
TF-IDF	Term Frequency-Inverse Document Frequency
UBFs	User-Based Features
URL	Uniform Resource Locator
W2V	Word2Vec

Chapter One

INTRODUCTION

1. INTRODUCTION

1.1 Overview

It is no doubt that social media has grown significantly to play major roles in people's social, economic, religious and political spheres. Subsequently, both the request and exchange of information have in parallel expanded following an increase in the number of users registering on social media platforms to satisfy their informational needs. Despite the existence of notable and crucial benefits being served by social media platforms, societies have also not been spared from the dangers of social media. One such danger that has attracted attention in academic spheres and the social media fraternity is clicking baits (Elyashar, Bendahan and Puzis, 2022) Drawing evidence from (Gosh, 2021) it is highlighted that inconveniences caused by click baits are a major concern to reckon with when using social media and websites. This is congruent with (Elyashar et al.,2022) observations denoting that click baits are always lurking high around almost any social media platform and website. This demands stern efforts to devise effective ways to cushion users from the adverse effects of click baits and calls for better, modern and sophisticated clickbait detection methods. hence, this underscores the importance of undertaking this study.

According to (Bieberich, 2002), clickbait refers to digital content that has been generated only for the aim of earning advertising money, sometimes at the expense of the content's worth and veracity. (Potthast et al., 2016) defined clickbait as short messages that lure readers to click a link. (Elyashar and Bendahan and Puzis, 2022) assert that clickbait posts are short, catchy phrases pointing to a longer online article. The magnitude of individuals exposed to the dangers and inconveniences of clickbait remains relatively high with social media platforms like Facebook registering a total of 2.8 billion users and YouTube with 2.2 billion users in 2022 (Dean, 2022), and an unaccountable number of people surfing the internet on various websites and social media platforms. For instance, (Dean, 2022) outlines that Facebook estimated that 5% of its 2.7 billion monthly active users are fake at any one time. Meanwhile, the efforts to prevent click baits are been countered by the lucrative benefits of luring users to increase site traffic. The Marketing Brew report uncovered those advertisers spent US$115 million on clickbait sites between January 2020 and May 2022, enough to buy some of the most expensive real estates in New York City (Barwick, 2022).

Given that numerous platforms consider “clicks” to be a measure of network flow, the process of click induction, known as “clickbait”, is widely prevalent. Click baiting is not aimed at providing high-quality content, but rather uses enticing headlines to lure users to click. Consequently, click-baiting tends to make users feel fooled or disappointed as poor-quality content is observed after clicking (Loewenstein, 1994). Other studies voice their concerns about click baiting citing that it poses serious harm to the society and economy through the spreading of fake information (Elyashar, Bendahan and Puzis, 2022; Indurthi et al., 2020; Jain et al., 2021; Kumar et al., 2018). Such has been the case with events observed during the COVID-19 pandemic (Moscadelli et al., 2020). (Breznitz, 2013) contends that clicking-baiting poses severe adverse effects during a public crisis when people’s attention is sincerely required but they have been frustrated by the previous clickbait because of “the cry wolf effect” to raise a false alarm. This further compound the introduction of various, unique and sophisticated click baits that are hard to detect and has been the case among Kurdish social media users. It is to the researcher’s attention that there are no existing clickbait detection methods trained to recognize the Kurdish language’s semantics and lexicon. Amid such observations, the next section of the study explores study problems linked to existing clickbait detection methods and proposes an alternative clickbait detection method that feasibly applies in the context of the Kurdish language's semantics and lexicon.

1.2 Problem Statement

Clickbait detection methods in their vast number have always been a subject of discussion in academic studies. With issues such as precision (Pujahari and Sisodia and Varshney and Vishwakarma, 2021) and accuracy (Jain et al., 2021) (Potthast et al., 2016) being raised against many clickbait detection methods, the quest to find new and better click bait detection methods is far from over and remains of huge value. Some of the existing methods like behavior analysis (Zheng et al., 2017), and cognitive evidence (Varshney and Vishwakarma, 2021) have succumbed to Objectivity problems. As a way of shedding lighter on the clickbait long-standing debate, (Elyashar, Bendahan and Puzis, 2022) proposed using training mainstream machine learning classifiers in detecting clickbait citing that some of the existing approaches were incapable of differentiating between clickbait and legitimate posts. Besides, a growing number of studies are pinpointing that a notable number of click baits are difficult to detect (Geçkil et al., 2020) (Naeem et al., 2020) (Pujahari and Sisodia, 2021). On a similar angle, (Pujahari and Sisodia, 2021) cite that click baits like headline cloning is difficult to detect. On a similar line, (Al-Sarem et al., 2021) mentioned that clickbait developers are increasingly becoming innovative and introducing complex and well-structured content that is difficult to detect. Besides, the entire click detection approach itself has been characterized by vital controversies. For instance, (Zhou et al., 2022) mentioned that certain existing clickbait detection studies regard the clickbait detection approach as a binary classification task and suffer from the problem of shallow usage of information that makes it easy to bypass them. It is in this regard that this study proposes the application of a deep learning approach and recurrent neural network towards clickbait detection.

Furthermore, a new development in text processing, deep learning-based methods, may assist increase the accuracy of clickbait detection algorithms and other strategies. In order to assess if a title is clickbait, current clickbait detection technologies need a vast amount of data. There are a variety of strategies for predicting the future based on the headlines of linked web pages. According to this, to determine whether or not the headline in question qualifies as "clickbait," one needs to look at what is being referred to in that headline (Setlur, 2018). If one of these ways works, it means a system must look at every single link connected to every single relevant headline on a site to determine whether or not it is clickbait. Additionally, if this technique were to be implemented, it would need a substantial amount of computational power. Consequently, the approaches' real-world applications are severely limited. A further problem is the ability of clickbait detection to adapt. Because of the subjective nature of clickbait, users may perceive the same title from a different perspective. As a result, a title might be categorized as either clickbait or a legitimate headline by different users (Zaremba et al., 2015). To improve their predictions over time, clickbait detection algorithms should be able to learn from user comments and alter their predictions appropriately. Text manipulation techniques, including machine learning, are also examined. Facebook, Google, and Microsoft are just a few of the companies searching for new methods of detecting fake material. Kurdish and other languages with fewer resources have been overlooked because of their availability as studies confine their analysis to the English language (Al-Sarem et al., 2021) (Jain et al., 2021) (Naeem et al., 2020) (Pujahari and Sisodia, 2021) (Varshney and Vishwakarma, 2021).

1.3 The Aim of This Thesis

The aim of this thesis is to develop an efficient and accurate clickbait detection system using deep learning algorithms. Clickbait refers to content that utilizes sensationalized or misleading headlines to attract users' attention and entice them to click on a link, often leading to low-quality or irrelevant content. The primary focus of this thesis is to create a robust clickbait detection model that can identify and distinguish clickbait from legitimate content across various online platforms, such as social media, news articles, and blog posts. As such, the study seeks to answer the following question;

1) How do cultural nuances, linguistic patterns, or language-specific characteristics impact the effectiveness of clickbait detection algorithms?
2) How does the performance of LSTM, BILSTM, CNN, GRU, and the hybrid CNN BILSTM deep learning techniques compare in clickbait detection for both English and Kurdish languages?

1.4 Thesis Objective

The main objective of the study is to design a model that can automatically differentiate between clickbait and legitimate content with high accuracy and efficiency. The study's secondary objectives are listed as follows:

1) Conduct a comprehensive review of clickbait detection methods, focusing on deep learning algorithms, in both English and Kurdish languages.
2) Identify cultural nuances, linguistic patterns or language-specific characteristics that impact the effectiveness of clickbait detection algorithms in English and Kurdish languages.

3) Develop and implement LSTM, BILSTM, CNN, GRU CNN-BILSTM, and hybrid CNN BILSTM models for clickbait detection in English and Kurdish languages.
4) Identify the most effective approach for clickbait detection in English and Kurdish languages in term of accuracy.

1.5 Scope of The Thesis

The study confines its examinations to 5,000 Real Kurdish news data collected from XENDAN, K24, RUDAW and other Kurdish news platforms. This also includes 16,001 non-clickbait and 15,999 clickbait English headlines retrieved from headlines comprising among others Wikinews, the New York Times, the Guardian, The Hindu, BuzzFeed, Upworthy, "ViralNova," "Thatscoop," "ScoopWhoop," and "ViralStories" headlines and a curated data set by (Chakraborty, Paranjape and Kakarla, 2016). Consequently, the study further restricts the examination of the 32 000 headlines to studying how linguistic analysis-trained LSTM, BILSTM, CNN and CNN LSTM rank in terms of accuracy and precision when used to detect click baits based on word embedding and gathered corpus of news headlines and contents.

1.6 Thesis Layout

The study ideas are structured into five chapters as follows:

Chapter 1: Pinpoints an overall picture of social media developments and the related inconveniences and dangers, notably, click baits. Additionally, the same chapter also provides insights into problems related to existing clickbait detection methods (study problems) and how they demand contemporary studies such as this one in developing new and effective clickbait detection methods that can cushion users, notably English and Kurdish users from click baits.

Chapter 2: Reviews related theoretical and empirical studies with the aim of identifying empirical gaps and devising effective ways of developing the desired clickbait detection method.

Chapter 3: Deals with the methodology of the study is structured.

Chapter 4: Is dedicated to the presentation of the empirical results.

Chapter 5: Concludes the study by inferring conclusions and recommending suggestions as per study findings and with the goals of furthering future studies.

Chapter Two

THEORETICAL BACKGROUND

2. THEORETICAL BACKGROUND

2.1 Introduction

Given the severity of research issues and empirical gaps linked to the detection of clickbait, especially strategies involving deep learning and neural network approaches, this chapter of the study applies related theoretical and empirical ideas intending to uncover and fill empirical gaps. Consequently, matters involving the concept of clickbait, clickbait structures and forms, the role of clickbait's cognitive mechanisms in clickbait detection, a deep learning approach to clickbait detection, a recurrent neural networks approach and related studies on clickbait detection (clickbait detection in English language and clickbait detection in other languages), are addressed in this chapter.

2.2 Clickbait

Given the nature and implications of clickbait, drawing further studies towards enhancing its understanding paves the way for numerous activities, methods and tools related to its solutions. To commence with the meaning of clickbait, Genç, 2021) defines clickbait as teaser messages or headlines that utilize complex or intriguing patterns or phrases designed to lure readers to specific content. Hence, in their deceptive nature, clickbait is presumed as having an ulterior motive of increasing social media users' or readers' click rates (Genç, 2021). However, the severity of clickbait implications is influenced by the type of clickbait strategy that often exists in the form of wrong, ambiguous, bait-and-switch, graphic clickbait, inflammatory, teasing, formatting and exaggeration (Biyani, Tsioutsiouliklis and Blackmer, 2016). The diversity of clickbait strategies broadens when (Pujahari and Sisodia's, 2021) established varieties involving Uniform Resource Locators (URL) redirection, headline cloning, and incomplete headlines are integrated into the discussion. By definition, URL redirection clickbait headlines redirect users to an unrelated website to increase its click rate (Pujahari and Sisodia, 2021). (Pujahari and Sisodia, 2021) also distinguished between headline cloning, and incomplete headlines citing that the former refers to using a headline for various contents without reflecting the content while the latter is intentionally left incomplete to increase user's curiosity.

Nonetheless, the misleading nature of clickbait places a huge demand for contemporary examination to further explore not only their detection but also their implications. Highlighting the problems associated with clickbait, (Shu et al., 2017) underscored that clickbait tends to increase the spreading of misinformation. This is because enticing users with exaggerated claims or fake news can boost inaccurate information's reach and possibly create confusion and influence public opinion. In the context of Tabloid journalism, (Chen, Conroy and Rubin, 2015) opine that clickbait causes a deterioration of journalism standards. This problem increases in intensity when media houses focus on prioritizing clicks and advertising revenue generating at the expense of ethical and responsible journalism (Rolnik et al., 2019). However, problems such as wasted attention and time are also prominent in any context. For instance, (Munder, 2020) argued that grabbing attention and generating clicks is the sole intention of clickbait and offers dubious value as they pay no attention to the value users will derive from spending time on the clickbait. In certain instances, (Hassoun et al.'s, 2023) information sensibility examination of individuals born between the mid-1990s and the early 2010s (Gen Z), Hassoun outlines that users are often frustrated when they are forced to sift through several details and fail to find engaging content. Concerns such as credibility and distrust issues have also been raised in prior studies. In support of such arguments, (Li et al., 2022) opines that continuous clickbait exposure undermines readers' trust. Users tend to develop a skeptical approach to information, especially when they have been repeatedly exposed to clickbait. In a related incidence, (Zannettou et al., 2019) directed the shallow nature of propagated information that spreads faster citing that such clickbait lacks substantive or meaningful details. Consequently, in most cases, clickbait offers little or no value to users. As a result, the overall quality of information online declines due to the high prevalence of clickbait. Such issues tend to gain huge credence when deceptive or misleading practices are involved. To reinforce this notion, (Biyani, Tsioutsiouliklis and Blackmer, 2016) highlighted that clickbait utilizes misleading or exaggerated headlines to lure users to visit a particular information site. Given these above-highlighted clickbait issues, the importance of detecting clickbait is justified and gains significant credence when other vital aspects are incorporated into the discussion. Nonetheless, this revolves around a proper examination and understanding of clickbait structures. In that regard, the next section of the study examines the existing clickbait structures.

2.2.1 Clickbait Structures

Clickbait has recognizable structures that separate them from non-clickbait headlines or statements. to begin with (Su's, 2020) establishments, clickbait headlines or statements can easily be detected because they use text-related methods. As a result, this often sparks concern regarding the credibility of such headlines or statements. In support of this notion, (Hardalov et al., 2016) disclosed that certain credibility features can be used in differentiating between credible information and clickbait. Furthermore, their findings uncovered that clickbait often contains several exclamation marks, upper cases and words like nobody, no and not. As a result, their findings direct attention to the importance of raising clickbait awareness and how non-technical means can be integrated with technical or systematic clickbait detection methods.

Nonetheless, the design of effective clickbait detection methods relies significantly on understanding the existing vital clickbait properties. Along similar lines, by applying a Stanford CoreNLP method, (Chakraborty et al.'s, 2016) examination revealed that syntactic and semantic nuances are the key linguistic features essential for differentiating clickbait from non-clickbait at both word group level, word level and sentence level. Furthermore, they suggested that such comparative examination uncovers vital distinguishing clickbait distinguishing information when clickbait and non-clickbait are distinguished according to the length of the syntactic dependencies, the length of the words, and the length of the headlines (Chakraborty et al., 2016). (Genç's, 2021) further reinforcement of this notion shows that non-clickbait sentences often include words such as a person omitting the function words and words pointing to a particular location. On the contrary, clickbait sentences often comprise function words like content words, question words, qualifiers, models, auxiliary verbs, pronouns, prepositions, conjunctions and determiners. Further, information derived from (Genç's, 2021) shows that shorter function words are a common feature among several clickbait sentences causing their sentences to have a shorter average word length. Another distinction between clickbait and a non-clickbait is available in (Chkraborty et al.'s, 2016) denoting that clickbait has longer syntactic dependencies. As a result, clickbait sentences have a longer distance between dependent words and the governing words. Additionally, (Zhou et al., 2020) noted that clickbait headlines contain both function and content words. (Genç, 2021) was also in support of (Ngan's, 2020) insinuation of clickbait and non-clickbait headlines and contended that clickbait has numerous complex phrasal sentences.

Following (Chakraborty et al., 2016)'s investigations that use word n-grams and part of speech tags to examine clickbait and non-clickbait headlines' patterns, 65% of the clickbait were discovered to have n-grams like “see what happened”. On the other hand, 19% of the non-clickbait data was observed as having n-grams. Nonetheless, their speech tags investigations showed that non-clickbait headlines often contain pronouns while clickbait contained numerous adverbs and determiners (Chakraborty et al., 2016). Chakraborty and others also discovered that clickbait sentences tend to use possessive and personal pronouns. (Potthast et al.’s, 2016) clickbait detection examination of web page-related tweets shows that mean word length, number of dots, image tags, mentions (@), hashtags (#), word n-grams and n-grams are essential features for distinguishing between clickbait and non-clickbait. Such insights are in support of (Chakraborty et al.’s, 2016) establishments denoting that mean word length and n-grams are applicable when comparing clickbait and non-clickbait with the former having more n-grams than the latter. As a result, these examinations show that numerous linguistic elements can be applied to detect clickbait. In as much as their findings are instrumental, the demand for innovative technological systems and tools like deep learning capable of effectively detecting clickbait sentences is always high. Hence, efforts to satisfy such inquiries in the context of this study will incorporate the application of deep learning and RNN methods.

2.2.2 Types of Clickbait

Apart from clickbait that shows information in an exaggerated manner and those with headlines that do not show the main content, other several types of clickbait exist. Drawing from (Biyani, Tsioutsiouliklis, and Blackmer’s, 2016) study, eight forms of clickbait existing in the form of wrong, ambiguous, bait-and-switch, graphic clickbait, inflammatory, teasing formatting and exaggeration strategies were identified. In another study, URL redirection, headline cloning and incomplete headlines were listed as the variants of clickbait headlines (Pujahari and Sisodia, 2021).

Following these, the importance of incorporating ideas on various forms of clickbait is of huge importance. However, such has not been covered in academic studies as ideas are confined to mere clickbait detection (Genç, 2021) (Indurthi and Oota, 2017) (Razaque et al., 2022) (Zhou, 2017). Hence, the current study adds to the existing body of knowledge by providing insights into the possible benefits of examining the available forms of clickbait. In that regard, one of the notable benefits of examining the available forms of clickbait is effective detection. This is because having such understanding allows developers and researchers to develop highly effective and significantly accurate detection algorithms. This can aid in alleviating problems associated with the unnecessary and invaluable grabbing of users' attention and the manipulation of their curiosity. Furthermore, it becomes highly feasible to precisely identify and flag clickbait. Therefore, understanding the various forms of employed clickbait strategies as well as their features is essential for effective and accurate clickbait detection model development.

Though not yet fully incorporated in contemporary examinations, ideas concerning how understanding the existing forms of clickbait plays an instrumental role in training data creation are nascent. As such, the application of methods such as machine learning (Genç, 2021), deep recurrent neural network (Razaque et al., 2022), self-attentive network (Zhou, 2017) and word embeddings (Indurthi and Oota, 2017) in detecting clickbait opens new avenues for designing better and effective data training methods. Hence, when achieved, benefits such as the development of effective and accurate detection algorithms become highly conceivable. In support of this notion, (Zannettou et al. ,2018) stressed the importance of having a huge dataset of labelled examples so as to develop reliable clickbait detection models. Moreover, representative and distinct training datasets covering numerous clickbait tactics and styles can easily be curated by having a solid understanding of existing clickbait forms. As a result, such possibilities aid in training detection models so that they can effectively and accurately identify patterns and subtle cues linked to various forms of clickbait.

Extending ideas to establish novel insights that understanding various forms of clickbait is vital and contributes to the development of feature engineering activities and programs are highly inevitable. This is because the development of effective machine learning models capable of dealing with clickbait issues is influenced by feature engineering development activities. In alignment with this proposition (Coddington, 2019) contends that vital characteristics capturing identifying clickbait content or headline features are influenced by the availability of information and understanding of various forms of clickbait. For example, some clickbait significantly depends on enticing questions, exaggerated claims or sensational language. Hence, an understanding of these characteristics aids developers in developing highly effective and accurate detection models.

One of the paramount reasons why this study incorporates ideas on types of clickbait in clickbait detection follows attempts to develop effective and accurate countermeasures as has been sought by other previous studies (Genç, 2021) (Indurthi and Oota, 2017) (Razaque et al., 2022) (Zhou, 2017). Therefore, the demand and execution of studies such as this current study serve to inform developers on how best they can structure the development of robust strategies and countermeasures to deal with clickbait. By understanding the underlying techniques and motivations behind clickbait, researchers can devise approaches to mitigate its impacts, such as designing algorithms to identify and demote clickbait content in search results or social media feeds.

Ideas about educating users and raising awareness about clickbait are hard to ignore in this context. Following the establishment of problems such as the provision of dubious value (Munder, 2020), generating at the expense of ethical and responsible journalism (Rolnik et al., 2019), deterioration of journalism standards (Chen, Conroy and Rubin, 2015) and the spreading of misinformation (Shu et al., 2017), the importance of educating users and raising awareness about clickbait is highly vital. This is because educating users and raising awareness enhances users' discernment abilities to identify and avoid clickbait, which helps them avoid becoming clickbait victims. In other words, educating users and raising awareness about clickbait helps in ensuring that the online community remains highly responsible and more informed about accessing online information, especially that involving clickbait. Despite these ideas being instrumental in clickbait detection, management and eradication, further examinations are required to explore their cognitive mechanisms. Of paramount importance is the notion that the development of effective and highly accurate clickbait detection models revolves around a proper examination and understanding of clickbait's cognitive mechanisms. Such has not been the case with several prior studies (Genç, 2021) (Indurthi and Oota, 2017) (Razaque et al., 2022) (Zhou, 2017). Hence, the incorporation of such ideas adds to the current study's originality. In that regard, the next section of the study examines the existing clickbait's cognitive mechanisms.

2.3 The Role of Clickbait Cognitive Mechanisms and Interest in Clickbait Detection

The amalgamation of cognitive mechanisms and clickbait detection is a critical facet in the realm of computer science, where prevailing studies predominantly focus on scientific methodologies like machine learning (Genç, 2021), deep recurrent neural networks (Razaque et al., 2022), self-attentive networks

(Zhou, 2017), and word embeddings (Indurthi and Oota, 2017). Unfortunately, the role of cognitive mechanisms often takes a backseat in these discussions, leading to a notable gap in understanding. The current study aims to rectify this omission by shedding light on the pivotal role that cognitive mechanisms play in clickbait detection. (Paranjape,2015) underscores that the primary objective of clickbait is to capture attention and entice users into clicking on specific headlines. However, the cognitive mechanisms underlying this phenomenon are frequently overlooked. This study addresses this gap by delving into the cognitive intricacies involved in clickbait strategies. Foremost among these cognitive mechanisms is the exploitation of curiosity. Clickbait strategically presents incomplete statements or questions, leveraging the inherent human trait of curiosity. By promising a revelation or answers to a specific situation, clickbait capitalizes on users' curiosity, compelling them to click on the content. Sensationalism is another facet linked to cognitive mechanisms in clickbait. (Chen, Conroy, and Rubin, 2015) argue that clickbait content is often sensationalized or exaggerated to enhance its appeal. This taps into users' emotions, using provocative statements, shocking images, and dramatic language to lure them into clicking on the content.

The Fear of Missing Out (FOMO) also plays a significant role, despite receiving limited academic attention (Prentice, 2022). Clickbait often employs FOMO by creating a sense of urgency or exclusivity, making users believe that accessing the provided links is crucial to staying informed or not missing out on exciting opportunities. Emotional triggers are integral to clickbait detection, as clickbait is designed to elicit specific emotional responses such as curiosity, amusement, surprise, and anger. These emotions capture users' attention and drive them to click on the content to satisfy or understand their emotional reactions.

Personal relevance is not to be underestimated in the cognitive mechanism and clickbait detection discourse. Clickbait headlines are tailored to resonate with users' personal experiences or interests, creating a sense of connection and relevance that enhances engagement. Ambiguity, characterized by vague statements and ambiguous language, is a major force in clickbait strategies. By excluding vital details or using click-inducing phrases, clickbait compels users to access headlines to uncover the full story. This exploration of cognitive mechanisms in clickbait detection is a novel contribution to the field, filling empirical voids and enhancing the design of accurate clickbait detection models. To further enrich the study's contributions, the next section delves into the role of curiosity in clickbait strategy. Interest, as an instrumental element of curiosity, plays a crucial role in clickbait strategy, connecting users' interests with the mechanisms employed by clickbait creators. Drawing from prior research (Biyani, Tsioutsiouliklis, and Blackmer, 2016) (Genç, 2021) (Thiel, 2018), it becomes evident that clickbait aims to captivate users by aligning with their interests. Clickbait creators enhance the appeal of their content by learning about users' preferred subjects, trends, or types of information, creating a connection between users' interests and clickbait strategies.

Interest intersects with clickbait strategy in various ways, as highlighted by (Thiel ,2018):

- **Targeting specific interests**: Clickbait developers analyze popular trends, topics, and keywords to align their content with subjects likely to attract attention, such as sensational events, viral stories, controversial issues, or celebrity news.
- **Personalization**: Clickbait strategies often involve personalizing content based on individual users' interests and preferences, achieved through targeted advertising or personalized recommendations.

- **Emotional appeal**: Clickbait leverages strong emotions like curiosity, surprise, fear, or excitement by tapping into users' emotional interests, crafting headlines that trigger psychological responses and prompt clicks.
- **Exploiting niche interests**: Clickbait extends beyond broad topics, catering to specific interests and communities by identifying subcultures or audiences and creating content that resonates with them.
- **Content packaging**: Clickbait developers use strategies to package content in ways that align with users' interests, employing provocative videos, intriguing headlines, or attention-grabbing thumbnails.

By understanding and leveraging users' interests, clickbait creators enhance the effectiveness of their strategies, ultimately increasing engagement and click rates. This comprehensive examination of curiosity and interest in clickbait strategy further contributes to the evolving landscape of clickbait detection studies.

2.3.1 The Role of Curiosity in Clickbait Strategy

According to (Loewenstein, 1994), curiosity is an instrumental element in art, literature, and lifelong development, producing scientific knowledge and gaining new knowledge by motivating people to examine the new stimulus and learn more about it. Loewenstein further highlights that curiosity is widely utilized for commercial purposes just as much as it is used in clickbait to persuade individuals to seek and acquire additional knowledge. (Gottlieb et al.'s, 2013) further developments in this regard have led to the establishment that curiosity triggers individuals to engage in information-seeking behavior is also essential (Gottlieb et al., 2013). However, it is imperative that such information-seeking behavior does not change the external environment or situations but rather changes individuals' epistemic state. Therefore, by the same implications, clickbait is in this context regarded as stirring users to seek additional information, which causes them to click on provided links. Furthermore, it thus becomes apparent in this context as suggested by (Gottlieb et al., 2013) that clickbait does not alter individuals' external circumstances or situations but rather alters their epistemic state. Advancing further (Gottlieb et al. ,2013) ideologies, information-seeking behavior is presumed to lower or eradicate users' uncertainty by altering their epistemic state. Hence, environment features such as surprise, uncertainty, novelty and randomness are intrinsically motivated and tend to drive users' actions to click the provided links. For example, an attentional attraction is generated by being intrinsically motivated as a result of novelty, which is salient. However, (Gottlieb et al., 2013) contend that informational gaps are bound to exist because there is no guarantee that the information to be accessed will aid users to learn and this occurs regardless of whether the surprising or novel stimuli cause curiosity-relieving behaviors.

Meanwhile, a clickbait's deceptive and misleading nature tends to create a knowledge gap and the more such discrepancy widens, the more users are enticed to click the links. As stated by (Munder, 2020), clickbait entices users by providing misleading or exaggerated information of dubious value. Thus, their misleading nature triggers an information gap, which triggers or compels users to click certain links.

(Biyani et al., 2016) and (Pujahari and Sisodia ,2020) both found different varieties of clickbait. These clickbait categories can be compared based on curiosity qualities like ambiguity, surprisingness, complexity and novelty. For instance, gruesome names, and frightening or scandalous terms used in graphic clickbait might be connected with surprise and novelty to pique readers' interest. Similar to this, misleading clickbait that piques curiosity with deceptive information can be compared to surprisingness and novelty. On the other side, the clickbait exaggeration tactic may be linked to stimulus surprisingness. Complexity can be seen in the structuring of inflammatory clickbait that employs nasty or offensive language or clickbait where textual characteristics are changed incorrectly. These headlines may appear challenging and complex to grasp because they misuse punctuation rules or use unpronounceable or unpleasant language. Since they function by omitting details or incorporating ambiguous and perplexing terms, some clickbait types, such as teasing, bait-and-switch, ambiguous, and incomplete, can be considered ambiguous. It becomes clear that clickbait does not have a single mechanism and encompasses the feeling of intrigue in all of its facets when clickbait methods are evaluated in this manner from many qualities of curiosity.

Curiosity-driven behavior can be explained by two hypotheses: complexity theories and novelty-based theories. According to (Düzel et al., 2010), novelty-based ideas contend that curiosity drives an organism to seek out novel stimuli and that learning more about it is satisfying. Rewards-responsive brain regions are demonstrated to be sensitive to novel stimuli in several neuroscientific investigations (Düzel et al., 2010). Several computer studies have shown that searching for originality can also be effective (Düzel et al., 2010) (Tang et al., 2017). All of these investigations support novelty-based theories, but they also have certain flaws. For instance, despite the notion that the agent's discovery of novel stimuli is advantageous to the agent, new stimuli may not always be instructive and helpful. They may lead the agent to confusing findings (Dubey and Griffiths, 2020). (Dubey and Griffiths, 2020) findings demonstrate that context provided by the likelihood of subsequent exposure to a stimulus or task (future occurrences), frequency of prior exposure to a stimulus or task (past exposure) and the environment all influence curiosity. Low confidence improves the value of information when prior exposure and future occurrences of a task or stimulus are independent (Brändle et al., 2020). This relates to novelty-based curiosity. If these findings are viewed in the context of clickbait news headlines, for instance, those who are not interested in the information might be more interested in the clickbait headlines when there is an event on the agenda that they are likely to encounter in the future. On the other hand, if previous exposure and future occurrences have a strong relationship, then knowledge with a modest degree of certainty is more valuable, which relates to complexity-based curiosity (Brändle et al., 2020). If an individual who is very interested in the agenda, for instance, sees clickbait news headlines about things he or she has read about frequently in the past, he or she will be more interested in clickbait news headlines on things she has read about repeatedly in earlier times. Examining all of these conversations and findings reveals that curiosity is influenced by a variety of factors, including the agent's psychological condition and the qualities of the stimulus. This is rather than being determined solely by one factor. As a result, these elements should be taken into account when analyzing clickbait news headlines and research that can harmonize various ideas should be developed.

2.4 A Deep Learning Approach to Clickbait Detection

In the study by (Mekruksavanich, Jitpattanakul, 2021) presented in Figure 2-1, a deep learning approach is defined as the act of automatically learning hierarchical representations of data and using deep neural networks in solving complex tasks. Deep Neural networks are another form of an artificial neural network comprising several hidden layers stacked together (Shi et al., 2021). As a result, deep learning is now widely applied in numerous fields such as speech recognition, computer vision and Natural Language Processing (NLP).

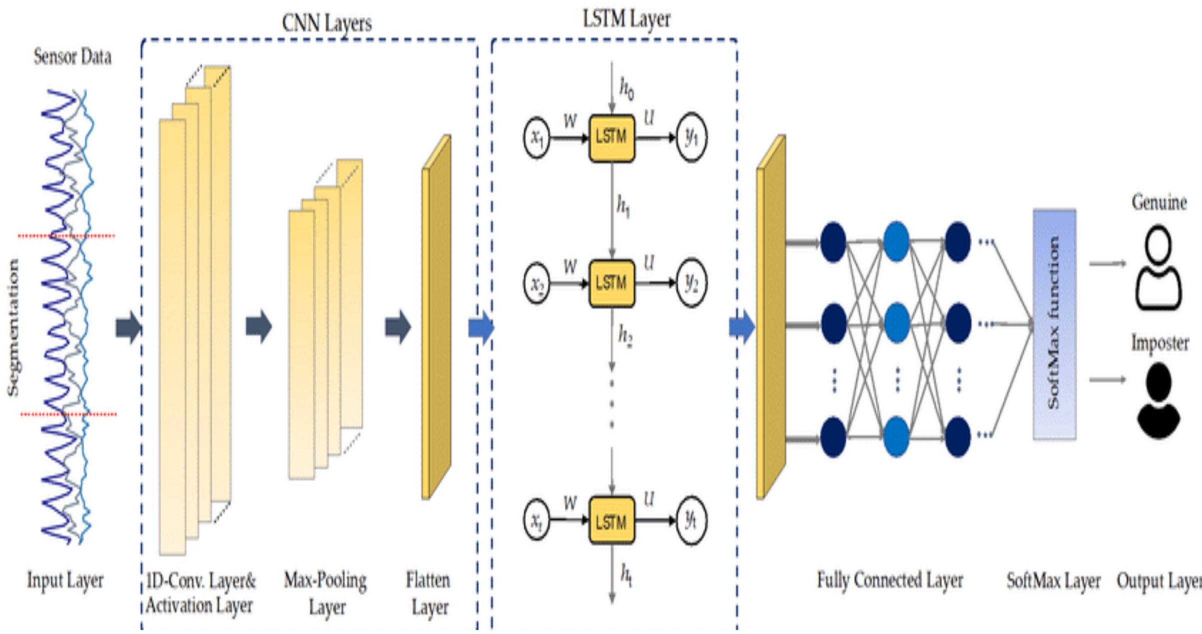


Fig. 2-1 Framework of Mekruksavanich and Jitpattanakul (2021)

In order to accurately classify clickbait and non-clickbait headlines or articles, a deep learning approach can be used to automatically learn and extract pertinent features from textual data. One of the key ways deep learning plays a crucial role in detecting clickbait is through feature representation. This follows observations made by (Gan, Sun and Sun, 2022) indicating that automatic learning of meaningful representations of data is a strength of deep learning algorithms. Furthermore, Martins, (Papa and Adeli, 2020) outlined that textual data is the main input for clickbait detection, and deep learning models can efficiently extract both low-level and high-level features from the text. These algorithms can uncover complex patterns and semantic links in the text through several hidden layers, facilitating the identification of clickbait-related indicators and traits.

The other benefit of applying a deep learning approach in clickbait detection is end-to-end learning. In support of this notion, Tian, (Chen and Shen, 2019) underscored that deep learning models can be taught end-to-end, which enables them to precisely relate desired outputs (such as clickbait or non-clickbait labels) to the intended inputs (such as textual headlines). Hence, once applied, the need for licit rule-based systems or manual feature engineering is eliminated. Therefore, a deep learning model will automatically learn to extract essential features and accurately predict the intended outcomes, especially when supplied with a large clickbait and non-clickbait dataset. This causes the entire clickbait detection process to depend less on human-designed features and more efficient. Figure 2-2 Model of (Chen and Shen, 2019) visually represents the discussed approach.

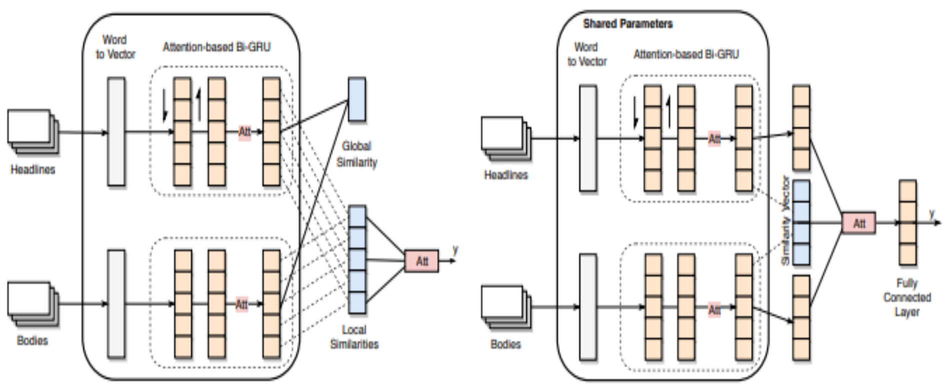


Fig. 2-2 Model of Chen and Shen (2019)

Benefits such as handling complex patterns can also be obtained by applying a deep learning approach to detecting clickbait. To draw attention and encourage clicks, clickbait headlines frequently use a variety of linguistic strategies, including insufficient information, ambiguity or sensationalism (Al-Sarem et al., 2021). These linguistic nuances can be effectively captured by deep learning models as they are well-designed to capture intricate dependencies and patterns. In addition, deep learning models can more accurately distinguish between clickbait and non-clickbait headlines because they can recognize complicated links among phrases and words by using the hierarchical representations they acquired in the hidden layers.

Scalability and adaptability are also some of the major benefits of applying deep learning models in detecting clickbait. In acknowledgement, (Long et al., 2016) cited that large-scale datasets can be handled by deep learning models and this makes them highly suitable for detecting clickbait. Furthermore, the fact that datasets collected from numerous domains can be used in training deep learning models entails that such models are well-equipped to generalize and adapt well to unseen and new clickbait patterns (Tian, Chen and Shen, 2019). Such an approach is of huge importance, especially given the evolving nature of clickbait methods (Pujahari and Sisodia, 2021). Therefore, practitioners and researchers can create more reliable and effective clickbait detection algorithms that capture the intricate and delicate character of clickbait headlines. This is done by utilizing deep learning and will increase the success rate of the campaign against clickbait content.

2.5 A Recurrent Neural Networks Approach to Clickbait Detection

An artificial neural network called a recurrent neural network (RNN) is first and foremost made to deal with sequential data (Mou, Ghamisi and Zhu, 2017). As such, an RNN is mainly ideal for use in activities involving time-series or time-series data like handwriting recognition, speech recognition, and natural language processing. When compared with traditional feedforward neural networks, which process data from input to output in a single pass (Choobi, Haghpanahi and Sedighi, 2012), RNNs' architecture includes loops that let them retain and use data from earlier time points or stages. Apart from being well-suited for activities where sequential and context patterns are vital, RNNs can also capture temporal dependencies. Furthermore, RNNs can maintain and update a hidden state whilst processing the input sequence step by step as illustrated in Figure 2-3. This is because of their recurrent unit, which serves as its vital building block.

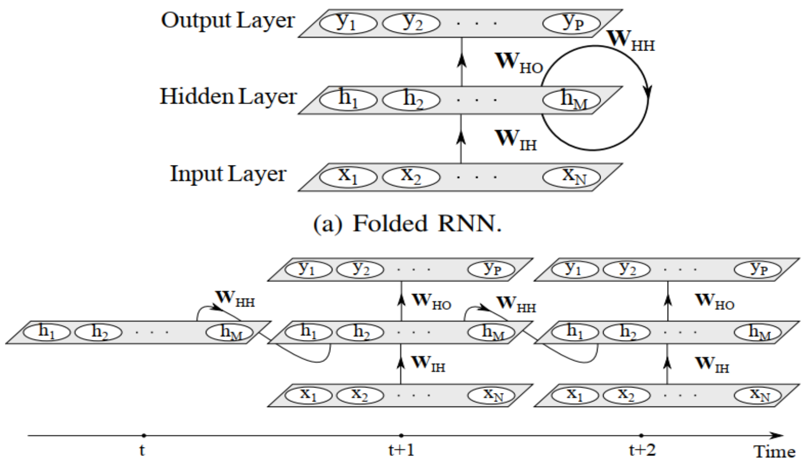


Fig. 2-3 Unfolded recurrent neural networks (RNN) through time.

According to (Qin et al., 2017), an output is produced and the hidden state is updated every time the recurrent unit uses the current input and combines it with the previous hidden state to produce an output and update the hidden state. Thus, the memory function of capturing information from prior time stages is performed by the hidden state. Consequently, this allows RNNs to recall prior events and utilize them to inform forecasts or make predictions. Two RNN variants in the form of a Long Short-Term Memory (LSTM) network capturing long-range dependencies and a Gated Recurrent Unit (GRU), which simplifies the LSTM architecture exist. They are also used in making predictions and dealing with sequential data (Mou, Ghamisi and Zhu, 2017).

Studies documenting the application of RNNs in clickbait detection have been on the rise in as much as the computer science field is concerned (Kumar et al., 2018) (Razaque et al., 2022) (Siregar, Habibie and Nababan, 2021). As such, various approaches to the application of RNNs in such regards have also been applied. Consequently, numerous outcomes were conceived and the divergence of outcomes can in most cases hinder consensus. Hence, the need to explore these studies and realign ideas to ensure that they converge to a widely accepted theoretical and empirical point becomes of instrumental importance.

Just like deep learning methods, the application of RNNs in detecting clickbait is essential. Foremost, RNNs possess a unique architecture and are effective in modelling sequential data (Mou, Ghamisi and Zhu, 2017). On the other hand, the entire clickbait detection process revolves around the analysis of sequential textual data like article content and headlines. By nature, RNNs can deal with sequential data as they can sequentially process each input element while keeping track of prior inputs. RNNs can capture dependencies and contextual information thanks to their capacity for serial modelling, which is essential for comprehending the complex patterns found in clickbait headlines. The language frameworks of clickbait headlines are frequently complex, requiring readers to comprehend the connections between phrases or words that may seem unrelated. Such long-term dependencies can be analyzed using RNNs with GRU or LSTM. These models are well-designed to capture long-term dependencies in the data and deal with problems associated with the vanishing gradient problem. RNNs are therefore very good at spotting clickbait headlines that rely on information or context that is dispersed throughout the entire sequence.

Benefits such as contextual understanding are also conceivable when RNNs are deployed in detecting clickbait. Given the idea that an effective approach to detecting clickbait places a huge demand for the texts' semantics and context (Chakraborty et al., 2017), the contextual representations can easily be learned by RNNs. This can be reinforced by assertions denoting that RNNs can preserve and convey data gathered from prior inputs and are particularly good at learning contextual representations (Choobi, Haghpanahi and Sedighi, 2012). These features enable RNNs to be highly effective when deployed in situations involving learning contextual representations. In addition, this makes it possible for them to accurately detect clickbait by capturing the context and meaning of phrases or words within a headline.

Drawing further, applying RNNs in detecting clickbait can also be said to be instrumental for variable-length input handling purposes. Following (Chakraborty et al.'s, 2017) observations denoting that clickbait headlines vary in length; handling inputs of variable lengths can prove to be a complex task. With the given nature of RNNs, handling inputs of variable lengths can easily be accomplished by applying RNNs. Besides, when compared with traditional bag-of-words models and other fixed-length methods, RNNs can dynamically adapt to headlines' internal states. This follows their ability to process inputs of variable lengths, which causes them to be more flexible (Bianchi et al., 2019).

Lastly, in support of the application of RNNs in detecting clickbait, (Mou, Ghamisi and Zhu, 2017) mentioned that RNN models are pre-trained on large-scale text datasets. Such datasets can include social media posts or news articles. Moreover, (Mou, Ghamisi and Zhu, 2017) contend that RNNs acquire generic language illustrations through pre-training on a large quantity of data, which may then be specialized for clickbait detection using a smaller labelled dataset. By utilizing the prior information that the RNN has already recorded, this method can enhance performance and efficiency in clickbait detection tasks. Hence, given all these highlighted advantages, it can be inferred that applying RNNs in detecting clickbait stands to be one of the effective methods that serve to deal with clickbait-related challenges.

2.6 Related Studies on Clickbait Detection

Given the existence of empirical voids related to clickbait detection in the English language, and other languages as well as detection precision, this section of the study is designed to fill such voids. Hence, related empirical examinations were conducted in this regard as follows:

2.6.1 Clickbait Detection in English Languages

Following the underlying observations denoting that clickbait has increased in number and form, the number of studies investigating clickbait detection has also increased. As such, researchers have been determined to identify clickbait in textual data or videos retrieved from various social media platforms like Facebook, YouTube and Twitter.

32,000 headlines for news articles were created by (Chakraborty et al., 2017) using data from ViralStories, Scoopwhoop, WikiNews, Thatscoop, ViralNova, Upworthy, BuzzFeed, New York Times and The Guardian. The articles annotated by the three annotators are reflected in the dataset as nearly equal amounts of clickbait and non-clickbait. Three machine learning algorithms; Random Forests, Decision Trees and Support Vector Machines (SVM) were used on this dataset, and SVM ranked the best with a precision of 93%. BILSTM was used on an identical dataset by (Anand, Chakraborty, and Park, 2017) and achieved an accuracy of 98%.

The dataset from (Chakraborty et al., 2017) was utilized in the study of Lopez-Sanchez et al. (2018) to present a strategy that adapts to users. For flexible clickbait identification, they created the Case-Based Reasoning (CBR) method, which they integrated with deep metric learning algorithms, word embeddings and wor2vec.

Word2Vec is a word embedding technique that represents words as numerical vectors, capturing semantic relationships between words based on their context in large text corpora.

Twitter has also been utilized to gather data for clickbait research, such as this thesis. In addition, websites like Facebook and YouTube. Twitter is a particularly useful tool for data collection because it frequently contains tweets about news or clickbait teasers. (Potthast et al., 2016) conducted one of the earliest studies in this field. generated a dataset using Twitter data and applied three different machine learning algorithms to it to detect clickbait in a social media stream. First, depending on the accounts' receivable turnover (RT counts), they chose which ones to employ for data extraction. In order to create the Twitter Clickbait Corpus, three reviewers annotated the data taken from the official Twitter accounts of news organizations like BuzzFeed, the Huffington Post, Business Insider and BBC News as being clickbait or not. They used three learning algorithms; Random Forest, Naive Bayes and Logistic Regression on this dataset. The Random Forest classifier outperformed all others with a precision and recall of 0.76.

Using headlines obtained from Twitter, (Chakraborty et al., 2017) produced a different dataset. The top three newspapers (India Times, Washington Post and New York Times) as well as one online news source (Huffington Post), which also has 38 subsidiary accounts, provided clickbait headlines for this dataset. The researcher also gathered headlines from these 38 supplementary accounts. The 27 primary and secondary accounts of the five outlets (Viral Stories, Scoop Whoop, Viral Nova, Upworthy and BuzzFeed), on the other hand, yielded non-clickbait data that was extracted. Over the course of eight months, tweets and retweets from these accounts were collected. This created the largest English dataset with 288K tweets and 11.4M retweets.

There are numerous clickbait datasets made using Facebook data in addition to YouTube datasets. A dataset comprising 1.67 million Facebook postings from 153 media companies was produced by (Rony et al., 2017). When they used this dataset to test their word-embedding-based clickbait detection model, it had an 84% accuracy rate.

(Qin et al., 2017) created a YouTube clickbait dataset intending to analyze clickbait in video teaser information. The first 123 characters, cover photos, thumbnail and title of the video description were all included in the teaser material for videos. They understood that this teaser information is insufficient for clickbait detection and that additional information from the video needs to be considered as well. Two reviewers annotated 109 YouTube videos in this dataset as clickbait or non-clickbait based on the video's title, description, thumbnail, and viewer comments.

(Potthast et al., 2018) developed an updated Webis Clickbait Corpus 2017 by enlarging the prior dataset and describing the annotation procedure. The accounts from which the data would be gathered were chosen. They were based on the number of retweets and the 38,517 teaser messages posted by accounts of 27 news publishers. From the Twitter accounts of a few news organizations, including tweets, Washington Post, CNN, Fox News, Business Insider, Guardian and Independent, and their media attachments were archived for six months. For payment, crowd workers initially assessed how clickbait the tweets were throughout the review process. Annotators were cautioned to pay attention to the visuals that accompany the headlines and avoid mistaking gossip tweets for clickbait. Second, the terms in the tweets that prompted the referees to categorize them as clickbait were noted. Finally, rather than completing a binary classification test, students were asked to rate the tweets on a four-point Likert scale. One of the most noteworthy aspects of this study is the use of a non-binary task to determine if the headlines are clickbait.

Online Video Clickbait Protector (OVCP), created by (Shang et al., 2019), can detect clickbait videos by looking at viewers' comments. To assess OVCP effectiveness, they gathered a dataset of YouTube videos. The videos in the collection were labelled as clickbait by three annotators. This dataset includes comment threads, comments, thumbnails, descriptions and the title for videos as well as information about how clickbait-worthy each video is.

There are studies where clickbait is employed in a different context outside of clickbait detection and prevention work in English. For example, the study by (Bhowmik et al., 2019) suggested that rather than deceptive products, clickbait messages may engage readers with trustworthy health-related information. They created an experimental setting where participants were shown articles with clickbait-style and non-clickbait-style titles and then asked if they wanted to read the content. If readers thought the story was trustworthy and wanted to share it, they intended to ask them. By suggesting that such research be conducted using a dataset specially constructed for this purpose, they provided a different viewpoint from clickbait headlines.

(Elyashar, Bendahan and Puzis, 2022) conducted a study on detecting clickbait in online social media. Their suggested method employs popular machine learning (ML) classifiers to distinguish between clickbait and authentic posts. The suggested classifiers are trained using a variety of linguistic, behavioral, and image-extracted data. We evaluated using two datasets from the 2017 Clickbait Challenge. With an AUC of 0.8, an accuracy of 0.812, a precision of 0.819, and a recall of 0.966, the Extreme Gradient Boosting (XGBoost) classifier performed the best. Finally, we discovered that determining the percentage of formal English words in the provided content is useful for identifying clickbait.

(Bronakowski, Al-khassaweneh and Al Bataineh, 2023) presented machine learning and semantic analysis methods for identifying clickbait headlines in the English language. The technique entails examining six different machine learning classification algorithms both alone and collectively while assessing thirty distinct semantic characteristics. According to the results, the most accurate algorithms identify clickbait headlines 98% accurately. The suggested models can be used as a blueprint for creating useful programs that easily identify clickbait headlines. Table 2-1 presents an overview of related works.

Table 2-1 Related work

Author	Dataset	Algorithms	Accuracy
(Bronakowski, Al-khassaweneh and Al Bataineh, 2023)	The dataset consists of 32,000 headlines collected from net	decision tree, logistic regression, naïve Bayes, support vector machine (svm), knn, GBDT	SVM and GBDT algorithm, achieved an accuracy of 98%
(Elyashar, Bendahan and Puzis, 2022)	it includes 2,495 posts, among them 762 clickbait posts, and 1,697 legitimate posts	XGBoost, Random Forest	an accuracy of 0.812, a precision of 0.819, and a recall of 0.966
(Rony et al.,2017)	dataset comprising 1.67 million Facebook postings from 153 media companies was produced by author	BERT, RoBERTa, GPT2	RoBERTa has best accuracy of 84%

(Chakraborty et al., 2018)	The author used (Chakraborty, Paranjape, & Kakarla, 2016) dataset, which has 32,000 headlines (clickbait and non-clickbait).	CBR + CNN	Using TF-IDF, Word2vec and n-gram count, the suggested method obtained 0.994, 0.95, and 0.90 average area under the ROC curve.

2.6.2 Clickbait Detection in Other Languages

Clickbait detection studies are of no doubt concentrated in the English language and their prominence serves to enhance the validation of proposed methodologies and practical suggestions. Nonetheless, the incorporation of other languages in long-standing debates is gaining huge prominence in academic studies. To support this notion, clickbait studies in Thai (Wongsap et al., 2018) and Chinese (Zheng et al., 2018) have also emerged in contemporary examinations. Hence, their inclusion including the current study into clickbait discussions broadens perspectives and approaches to clickbait detection.

(Zheng et al., 2018) constructed a dataset including 14,922 headlines taken from four famous Chinese news websites (Tencent, 163, Sohu, Sina) and well-known blogs. They proposed a Clickbait Convolutional Neural Network (CBCNN) consisting of Word2Vec models and a CNN model. They applied this model to Chinese news headlines preprocessed with stop-word filtering, part-of-speech filtering, and segmentation. CBCNN model performed with an 80.50% accuracy.

(Wongsap et al., 2018) applied Decision Tree, Support Vector Machine, and Naïve Bayes on a dataset with 5,000 Thai news headlines which were labelled as clickbait and non-clickbait by two users, and another dataset consisting of special characters such as '!', '?', and '#.' The results showed that the decision tree classifier gave 99.90% accuracy on the special characters dataset, which is the best performance in their study. On the other hand, the decision tree classifier performed 84.79% accuracy on the news headlines.

(Geçkil et al., 2018) formed a Turkish dataset with 2000 clickbait and non-clickbait news headlines extracted from Twitter, and 2000 clickbait and non-clickbait news gathered from the websites of the selected news outlet. The data of BBC Turkish and Anadolu Agency were labelled as non-clickbait since they were considered as being unlikely to be clickbait, while the data of other media institutions such as Hurriyet, Vatan, and Sabah Daily Newspaper were labelled as clickbait. They applied the TF-IDF method on this dataset, which gave an 87% accuracy.

(William and Sari, 2020) constructed a dataset, CLICK-ID, consisting of 15,000 annotated Indonesian news headlines gathered from 12 Indonesian online news publishers. They applied BILSTM and CNN to this dataset. Results show that BILSTM performs with an 81% accuracy while CNN performs with a 79% accuracy on the stemmed words of the dataset.

(Al-Sarem et al., 2021) used a machine learning-based approach and improved multiple features for Detecting Clickbait News on Social Networks in the Arabic language. This was done using 54,893 Arabic news data items collected from Twitter with 56.31% being legitimate news and 43.69% comprising clickbait news. According to the experimental findings, the SVM with the top 10% of ANOVA F-test features (Content-Based Features (CBFs) and User-Based Features (UBFs)) had the greatest results and had a detection accuracy of 92.16%. This indicates the unending possibilities of achieving higher accuracy rates using other clickbait detection methods as intended by this study.

(Fakhruzzaman, Sa'idah and Ningrum, 2023) applied fine-tuned transformers in studying the flagging of clickbait in Indonesian online news websites. The classifier performed remarkably well with a training dataset of 6,632 headlines. It received an accuracy score of 0.914, an F1 score of 0.914, a precision score of 0.916, and a receiver operating characteristic-area under curve (ROC-AUC) score of 0.92 when evaluated with 5-fold cross-validation. This reinforces the current study's argument that there are unending possibilities for achieving higher accuracy rates using other clickbait detection methods.

2.6.3 Clickbait Detection Precision

Subject matters involving the detection of clickbait are not merely a matter of identifying clickbait but rather extend to encompass accuracy and degree of detection. Hence, attempts to develop models that can accurately detect clickbait to a higher degree revolve around the examination of the related accuracy and degree of other detection methods. In that regard, previous related examinations of this nature will be examined.

Foremost, attempts to determine the accuracy of clickbait detection methods have been characterized by the application of various labelled datasets. Consequently, this has a profound effect on the established findings. For instance, by using 762 evaluation and 4761 Train clickbait datasets and 1697 evaluation and 14777 Train non-clickbait datasets, Thomas' system (2017) system ranked 6th of the 13 participating teams. Additionally, the system also achieved an accuracy of 0.826, an F1 score of 0.564 and a mean squared error of 0.0428. (Setlur, 2018) applied a semi-supervised confidence network and a gated attention-based network in assessing labelled samples for clickbait. The results of their study indicated that a detection accuracy rate of 97% can be achieved from 30% of strongly labelled samples we can achieve over. In another related instance, (Nadia and Iswanto, 2021) applied the Backpropagation Neural Network in analyzing Indonesian data for clickbait. The standard algorithm achieved a precision score of 67% and a recall and F1 score of 66%, while the modified algorithm had a precision score of 78% and a recall and F1 score of 76%. These examinations are in support of notion highlighting that in the digital age, clickbait is a rising issue, and experts are continuously looking for new solutions to address it (Naeem et al., 2020) (Potthast et al., 2016) (Vorakitphan, Leu and Fan, 2019). Using neural networks, such as the Backpropagation Neural Network, is one possible strategy. In a recent study, (Nadia and Iswanto, 2021) used Indonesian data using this technique to great effect. With a precision score of 67%, the conventional algorithm correctly classified 67% of the items as clickbait. Two-thirds of all clickbait items could be successfully identified by the algorithm, according to the recall and F1 scores, which were both 66%. (Nadia and Iswanto, 2021) also created a revised algorithm that produced even better outcomes. This novel method has a precision score of 78%, a recall score of 76%, and an F1 score of 76%. Thus, for anyone concerned about the proliferation of false or sensational content online, these findings have significant ramifications.

Using a clickbait corpus of 2992 Twitter tweets comprising 1697 non-clickbait and 762 clickbait, (Potthast et al., 2016) used a random forest classifier to determine the precision of the clickbait detection mechanism with 215 features. The applied logistic regression analysis results produced a 0.79 Area under the ROC Curve (ROC-AUC) at 0.76 recall and 0.76 precision. The results of their study also indicate that the model's performance varied depending on the type of clickbait being classified. That is, the model struggled with tweets that used made vague promises or rhetorical questions. Overall, the study highlights the potential for machine learning to identify and combat clickbait on social media platforms. However, it also underscores the need for continued research and development in this area to improve these models' accuracy and effectiveness. Additionally, it raises significant questions about algorithms' role in shaping users' online experiences. It also raises important questions about how we can ensure they are used ethically and responsibly.

(Khater et al., 2019) exploration of clickbait detection involving 24 trained features extracted from a social media posts dataset achieved an area under the ROC curve of 0.7 and an F1-score of 79%. The ROC is higher than (Potthast et al.'s, 2016) ROC of 76% and signals improvements in the applied clickbait detection methods. Hence, the possibility of conceiving better detection performance remains feasible and this study intends to achieve highly effective clickbait detection in both English and Kurdish languages.

(Klairith and Tanachutiwat, 2018) applied natural language processing with machine learning methods with the aim of assessing Thai clickbait detection algorithms involving crowdsourced 30,000 headlines. Their findings point to BILSTM with word-level embedding demonstrating superior performance compared to other models with an F1-score of 98% associated with an accuracy rate of 98%. When compared with (Potthast et al.'s, 2016) 79% ROC-AUC and (Khater et al.'s, 2019) ROC curve of 70% and an F1-score of 79%, substantial differences in performance are a major force to reckon with. This gains huge credence especially when a different language like the Kurdish language is introduced into the long-standing clickbait detection debate. Hence, these findings reinforce the current study's attempts to apply different algorithms in detecting clickbait in English and Kurdish languages.

(Vorakitphan, Leu and Fan, 2019) applied word embedding models in the form of an Ontology-based LSTM Model (OLSTM) to detect clickbait. The model's performance was validated using real data from news websites and Twitter. Clickbait detection accuracies ranging from 80% to 90% compared with other prior related methods were achieved. Again, incorporating (Potthast et al.'s, 2016) 0.79 ROC-AUC and (Khater et al.'s, 2019). In that manner, the application of the LSTM, BILSTM, GRU, CNN and the hybrid CNN BILSTM model in this study was highly justified.

(Naeem et al., 2020) conducted a linguistic analysis involving natural language cues using a deep learning framework for clickbait detection on social area networks. LSTM was used for classifying the decision-making task. Their model achieved a higher performance level of 97% accuracy compared to other study models. At this juncture, the reviewed studies highlight that higher clickbait detection accuracy rates are conceivable when new and innovative methods are applied to detecting clickbait, especially in English and Kurdish languages.

2.7 Summary

In this chapter, clickbait cognitive mechanisms mainly curiosity and interest were established as having an influence on clickbait. In that manner, the roles and importance of deep learning and RNNs methods in detecting clickbait were established. These include among others feature representation, end-to-end learning, handling complex patterns, scalability and adaptability, modelling sequential data, contextual understanding, variable-length input handling purposes, and pre-trained on large-scale text datasets. The review proceeds to outline the existence of clickbait detection in the English language and other languages such as Arabic, Chinese, Turkish and Indonesian. Amid such attempts, no study had yet analyzed clickbait detection in the Kurdish language nor compared models detecting both clickbait in Kurdish and English languages at the same time. In that regard, the next section proceeds to examine how deep learning and RNN methods will be used in detecting clickbait in the Kurdish and English languages.

Chapter Three

THESIS METHODOLOGY

3. THESIS METHODOLOGY

3.1 Introduction

This chapter of the study is dedicated towards the examination of methods and procedures carried out in executing the application of deep learning and neural network methods in clickbait detection.

Hence, it is in connection with the above-mentioned research inquiries that this chapter was organized to incorporate data collection, text classification, text processing and the architecture of the proposed frameworks.

3.2 Data Collection

In order to aid the detection of Fake News in the Kurdish language, 5,000 Real and 5000 Fake Kurdish news data were collected from XENDAN, K24, RUDAW and other Kurdish news platforms, Table 3-2 shows some sample of collected data. However, there existed no website assessing the validity of Kurdish news at present the study was conducted. As part of initiatives aimed at enhancing the study's validity, fake news from prior datasets was not translated.

Meanwhile, concerning the English data sets, 32,000 unprejudiced headlines on famous websites were compiled into a single data set and each class comprised 7,500 headlines chosen among its members. The headlines comprised 15,999 clickbait headlines and 16,001 non-clickbait headlines. Among the collected 32,000 unprejudiced headlines were "Viral Stories", "Scoop Whoop", "That scoop", "Viral Nova", the New York Times, the Guardian, the Hindu, BuzzFeed, Upworthy and Wiki News headlines data. The selected dataset was curated by (Chakraborty, Paranjape and Kakarla, 2016) and rated by at least three individuals as either clickbait or not. Subsequently, four algorithms namely; CNN, CNN BILSTM,

BILSTM and LSTM were applied to randomly select the unprejudiced sample of headlines for categorization. Table 3-1 presents illustrative instances of both clickbait and non-clickbait content in the English language.

Table 3-1 English Dataset Samples.

Headline	Clickbait
1- Can You Guess These Celebrity Fragrances by Their Descriptions.	YES
2- Here Are the Bronzers and Blushes That Actually Look Amazing on Women of Color.	YES
3- YouTube to reward users for posting creative videos	NO
4- Two British hostages feared dead after bodies found in Iraq	NO

Table 3-2 Kurdish Dataset Samples.

Headline	Clickbait
1- له ئەمریکا له پاشماوەی خۆراک پەین و وزه بەرهەمدەهێنرێت	NO
2- تۆپخانەکانی سوپای تورکیا ناوەندی گوندی هروور تۆپباران دەکەن	NO
3- دوو چەکداری داعش له بەکرەجۆ کوژران	YES
4- کریستیانۆ رۆنالدۆ دەچیته یانەی بارشلۆنه	YES

3.3 Text Classification Process

Given that the study relies on a pool of Kurdish and English language datasets, the text classification process was structured in a manner that accommodates differences between the two datasets. The process was conducted as follows:

3.3.1 English Text Classification Process

The data preprocessing procedure includes the processing of words into segments, feature vector representation, and the elimination of stop words and punctuation. The larger and more selective pre-processed text either emphasizes the greatest features of the data in a more prominent manner (feature selection) or reduces the complexity of high-dimensional data by describing it in fewer dimensions (feature extraction).

3.3.2 Kurdish Text Classification Process

Text processing is an essential Natural language processing (NLP) element that separates relevant data from irrelevant data before feature extraction (Altheneyan and Alhadlaq, 2023). The motive is to ensure that quality textual input is offered before being classified. In this regard, a Kurdish Language Processing Toolkit was used (Ahmadi, 2020). In order to enhance the quality of the text data and the reliability of the applied statistical examination approaches, special characters like extra space, emojis, foreign word languages, URL, &, % and @ were removed. Subsequently, the text was converted to UTF-8 Unicode. The text data was also subjected to a tokenization process through which sentences were split into words (Domingo et al., 2023). After having subjected the text data to a tokenization process, the study proceeded to remove frequently used words that are known as stop words (so on, the, and a). According to (Domingo et al., 2023), these words are worth removing because they do aid in distinguishing between two. After that, the study proceeded further to conduct what is known as stemming so as to reduce inflexion in words to their root form categorization. Table 3-3 and 3-4 shows Data Samples after text cleaning process.

Table 3-3 English Dataset Samples after text cleaning.

Headline
1- guess celebrity fragrances descriptions.
2- bronzers blush actually look amazing women color.
3- YouTube reward users posting creative videos.
4- two British hostages feared dead bodies found Iraq.

Table 3-4 Kurdish Dataset Samples after text cleaning.

Headline
1- ئەمریکا پاشماوە خۆراک پەین وزە بەرهەم
2- تۆپخانە سوپا تورکیا ناوەند گوند هروور تۆپباران کردن
3- دوو چەکدار داعش بەکرەجۆ کوژران
4- کریستیانۆ رۆنالدۆ چوون یانە بارشلۆنە

3.4 Text Processing

Models of language structure based on vector spaces semantics, such as the GloVe Model, represent each word as a single real-valued vector of length. (Chakraborty, Paranjape and Kakarla, 2016). They can be used for a variety of tasks, including parsing, named entity identification, question answering, document categorization and information retrieval (Yang, 2013). In many cases, a corpus of words is represented in this way; a particular set of "word vectors" are used to indicate how near or far distant the words are from one another. The total quantity of dimensions that divide two-word vectors from one another in addition to their distance from one another is counted in an innovative evaluation method that utilizes word similarities. All unsupervised word representation learning techniques are built on data on term distributions gathered in a corpus. However, others have made assumptions about how to interpret this data and how word vectors could represent that interpretation. Glove stands for "Global Vectors", a novel word-representation technique that automatically records global corpus data.

3.5 The Architecture of The Proposed System Frameworks

In this section, a system framework was proposed. The architecture of the system frameworks comprised an English system framework and a Kurdish system framework. The design or architectures of the two systems made it possible to compare and determine how LSTM, BILSTM, CNN, GRU, CNN-BILSTM and hybrid CNN BILSTM algorithms rank in performance in terms of detection clickbait in Kurdish and English languages. Figure 3-1 illustrates Framework of proposed system.

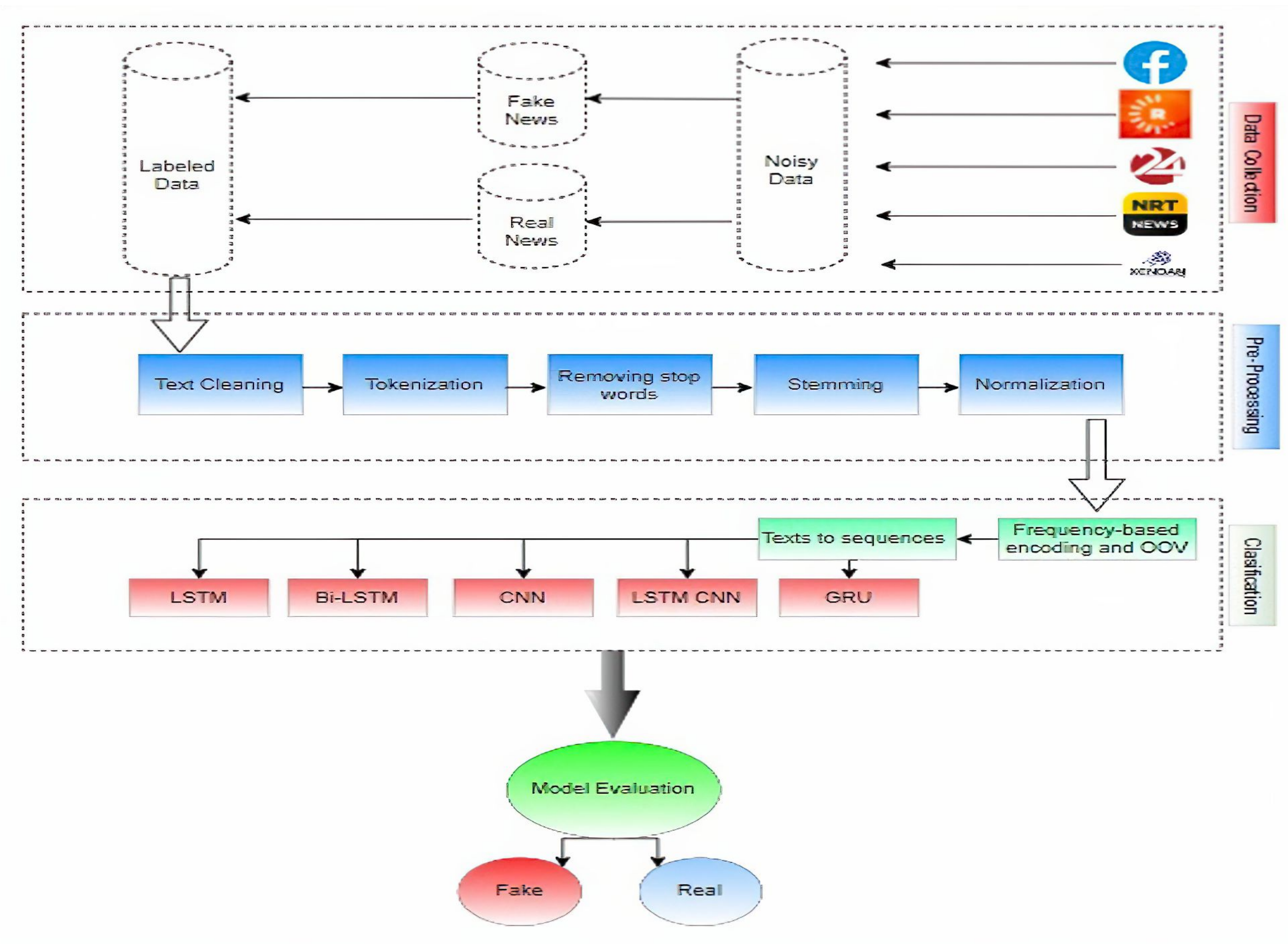


Fig. 3-1 Framework for English and Kurdish fake news detection

In this study, the author proposed a method for detecting clickbait that makes use of Deep Learning classifiers and features that are independent of language. In light of this, the current study's method may be used to train and classify articles written in any language. The technique is based on Convolutional Neural Network (CNN), LSTM, BILSTM, and GRU models that have been trained using data from a distributed feature space (Biyani, Tsioutsiouliklis and Blackmer, 2016), such as word or character embeddings. Below, we'll describe the various steps we took to create the current study's "clickbait" detection algorithm and the rationale behind each one. At the same time, evaluate the results of the investigations and compare them to those in the academic literature.

Our methods for identifying clickbait are based on a number of decisions including the classification model, the use of a smart classifier, the choice of the most effective algorithm, and the construction of accurate assessments of accuracy. By focusing on universal traits, this sophisticated approach may be easily extended to include new types of content and new languages. For the proposed model to be useful in a context where many languages are present, it must be trained on more than just English. This process of proposed system represents the stages in this model, shows how we arrived at the current study's final product (colored in green). The selected lexical qualities are organized into four groups, while the extracted characteristics are presented in grey (shown in orange). We started by looking at the qualities and datasets that have previously been utilized in the literature, and then we picked the one that was the greatest fit for these purposes. In the steps that followed, through cleaning up the data by eliminating duplicates and outliers and by isolating linguistic characteristics. All of the obtained characteristics were normalized such that they fell within the range [0,1] before being used in a smart classifier. The best findings found in the study were used to choose the intelligent algorithms (LSTM, BILSTM, CNN, GRU, CNN BILSTM) that were put through their paces in this evaluation. The last steps included adjusting the smart models' settings to optimize performance and determining the metrics.

3.5.1 General Steps to Check Headlines for Clickbait Detection:

1- Data Collection and Preprocessing:

For this example, let's assume we have a small dataset with the following labeled headlines:

- "5 Shocking Ways to Lose Weight Fast!" (Clickbait)
- "How to Cook a Healthy Meal in 30 Minutes" (Not Clickbait)
- "10 Celebrities Who Look Unrecognizable Now!" (Clickbait)

- "The Science of Climate Change" (Not Clickbait)
- "دوو چەکداری داعش لە بەکرەجۆ کوژران" (Clickbait)
- "لە ئەمریکا لە پاشماوەی خۆراک پەین و وزە بەرهەمدەهێنرێت" (Not Clickbait)

We'll preprocess this data into numerical format and train the model.

- Remove punctuation and special characters using online python library for English language (Porter Stemmer).
- For Kurdish language we use Kurdish Language Processing Toolkit—KLPT toolkit for preprocess, stem, transliterate and tokenize and addresses basic language processing tasks such as text preprocessing, stemming, tokenization, spell-checking and morphological analysis for the Sorani and the Kurmanji dialects of Kurdish.

2- Tokenization and Padding:

First, we need to tokenize the text and convert it into numerical format. We'll assume a simple vocabulary for this example:

Vocabulary English: {"5", "shocking", "ways", "to", "lose", "weight", "fast", "how", "cook", "a", "healthy", "meal", "in", "30", "minutes", "10", "celebrities", "who", "look", "unrecognizable", "now", "the", "science", "of", "climate", "change"}

Vocabulary Kurdish: {"ئەمریکا ", "پاشماوە", "خواردن", "پەین", "وزە", "بەرهرمهێنان", "چەکدار", "داعش", "بەکرەجۆ", "کۆژران"}

After tokenization, our English examples look like:

Clickbait: [1, 2, 3, 4, 5, 6]

Not Clickbait: [7, 8, 9, 10, 11, 12, 13, 14, 15]

various numerical representations are used to convert textual data into a format that can be processed by machine learning algorithms. Each word is represented as a binary vector where all elements are zero except for the index that corresponds to the word's position in the vocabulary.

We'll pad the sequences to a fixed length, for example, 10:

Clickbait: [1, 2, 3, 4, 5, 6, 0, 0, 0, 0]

Not Clickbait: [7, 8, 9, 10, 11, 12, 13, 14, 15, 0]

We do same thing for Kurdish headlines

3- Define the Model:

Create a neural network. text classification may consist of:

- An embedding layer: Converts words into dense vectors.
- One or more LSTM layers: Captures sequential information.
- Dense layers with activation functions (e.g., ReLU or sigmoid) for classification.
- Output layer with a sigmoid activation function for binary classification (clickbait or not).

4- Compile and Train the Model

5- Make Predictions

3.5.2 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is specifically designed to capture long-term dependencies and patterns in sequential data (Zheng et al., 2017). As a result, LSTM is designed to deal with challenges associated with the use of traditional RNNs. In support of such a notion, (Singh et al., 2023) contends that RNNs are limited in terms of their ability to retain information over long sequences. Thus, by structure, LSTMs have memory cells capable of storing information over extended periods and can optionally forget or recall information, and convey necessary information to future time steps. According to (Singh et al., 2023), such memory cells comprise an input gate, a forget gate, and an output gate used for regulating the way information flows within the cells, thereby conferring them with an ability to regulate the input, leaving unnecessary information, and output relevant information.

In a text representation learned from word embeddings, words with the same semantic meaning are represented comparably. Word embedding methods can be employed to extract a vocabulary's vector representation from a corpus of text. When the present state of a cell shifts, the forget gate (ft) regulates how much information can be sent to the subsequent time axis. This gate adds the newly acquired data to the cell's state following a forget operation. Due to its focus on memory updating, this gate differs from RNN's typical design. As shown in Figure 3-2, the first stage contains an embedding layer with an input and an output shape. An LSTM layer and a sigmoid neuron layer with input forms of 100 are positioned between the other two layers due to binary classification. To facilitate multiclass categorization, Softmax activation is employed in the final layer. Softmax converts the raw output scores into probabilities, ensuring that the sum of probabilities across all classes is equal to 1. This allows the model to make confident predictions among multiple classes based on the learned features.

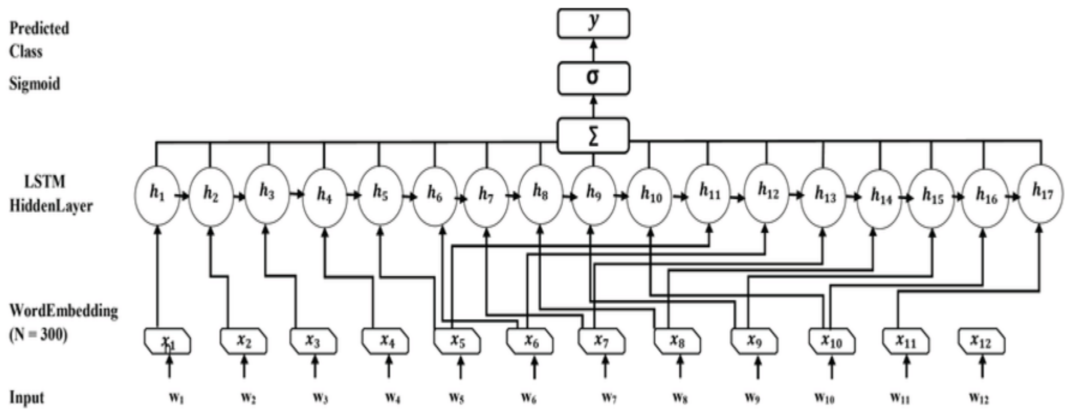


Fig. 3-2 Model configuration of LSTM

Concerning the detection of clickbait, the applications of LSTM in this study serve to enhance the analysis of article titles or headlines' textual content so as to ascertain whether they can be classified as clickbait or non-clickbait. There are other benefits tied to the application of the LSTM in this study. For instance, by applying an LSTM in detecting clickbait, the study was able to capture sequential patterns. This is because LSTM models are specifically developed to deal with sequential data (Zheng et al., 2017). As a result, this allows them to effectively capture the contextual dependencies present in clickbait headlines. Additionally, because LSTM can spot patterns across longer sequences (Zheng et al., 2017), they are appropriate for examining the clickbait titles' structure and composition. Furthermore, LSTM was applied in this context because of their capacity to handle variable-length inputs. That is, compared to fixed-length models such as traditional feedforward neural networks, LSTM models are well posed to sequentially process and handle different headline lengths. Consequently, there is an acceptable level of flexibility in dealing with text data that is obtained from the application of LSTM models, which deals with significantly lengthy headlines. Apart from this, by applying LSTM models, it became possible to learn long-term dependencies. This stems from the fact that they are excellent for capturing long-term dependencies in the text since they can hold information for long periods (Singh et al., 2023). This plays an instrumental role in enhancing the model's ability to comprehend the headlines' context and make highly informed decisions concerning clickbait classification. In connection with this, the incorporation of LSTM models in this context follows their capacity to learn clickbait features' general representations (Singh et al., 2023). This allows them to effectively generalize to new or unseen headlines. Such an ability to adaptation ability is significantly vital in contemporary real-world situations as sophisticated and other new forms of clickbait are continuously emerging over time. However, it is instrumental to note that hyperparameter tuning, selected architecture, the training data's diversity and quality, and other related implementation factors tend to influence LSTM models' performance.

3.5.2.1 Production Mode of LSTM Model:

This Streamlit application, "Clickbait Detective," stands at the intersection of cutting-edge deep learning technology and user-friendly web interfaces, offering a practical solution to the clickbait conundrum. Designed as an intuitive and accessible tool, this application empowers users from all walks of life to identify and combat clickbait in real-time, arming them with the knowledge and discernment necessary to navigate the digital landscape with confidence.

At its core lies the power of Long Short-Term Memory (LSTM) models, a category of deep learning algorithms renowned for their ability to understand and process sequential data effectively. These LSTM models have been meticulously trained on a wealth of textual data, enabling them to identify patterns and attributes that are indicative of clickbait content. Through the application's user-friendly interface, users can submit headlines or text snippets for analysis, instantly receiving a verdict on whether the content exhibits clickbait tendencies. Figure 3-3, Figure 3-4 shows Streamlit application web design.

Fig. 3-3 and 3-4 Web application mode of LSTM model

3.5.3 BILSTM

By definition, Bidirectional Long Short-Term Memory (BILSTM) algorithms are a form of RNN architecture that are prevalently applied for sequence processing activities like natural language processing (NLP), (Moon et al., 2021). In other words, BILSTMs represent an advanced form of traditional LSTM model that allows a bidirectional flow of information from future to past and from past to future using two linear support vector machines. According to (Moon et al., 2021), such a bidirectional flow of information enhances BILSTMs' capacity to capture succeeding and preceding contexts. This is crucial for activities and situations requiring a comprehensive understanding of a sequence of tasks or activities. Figure 3.5 shows the BILSTM system configuration while Figure 3.6 provides a graphical representation of Bi-directional LSTM.

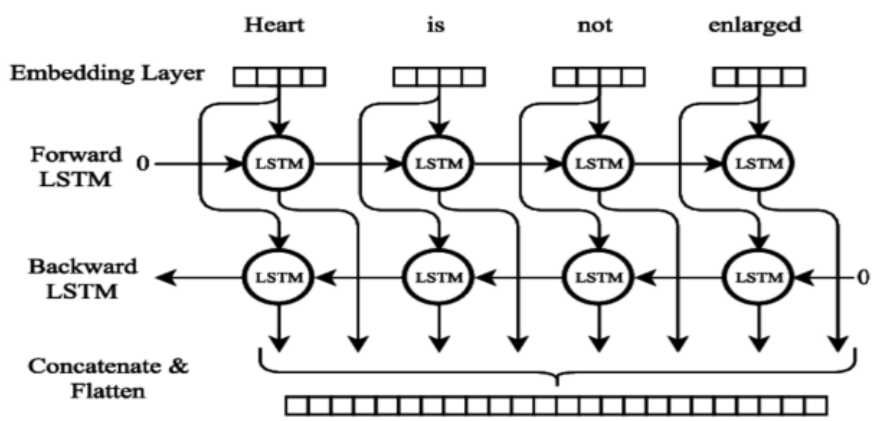


Fig. 3-5 BILSTM system configuration

The amount of information transferable from one-time axis to another in the LSTM can be limited by the forget (ft) gates. (Shang et al., 2019) contend that such an input gate's main is designed to restore the information cell's state to its original state before the information was lost.

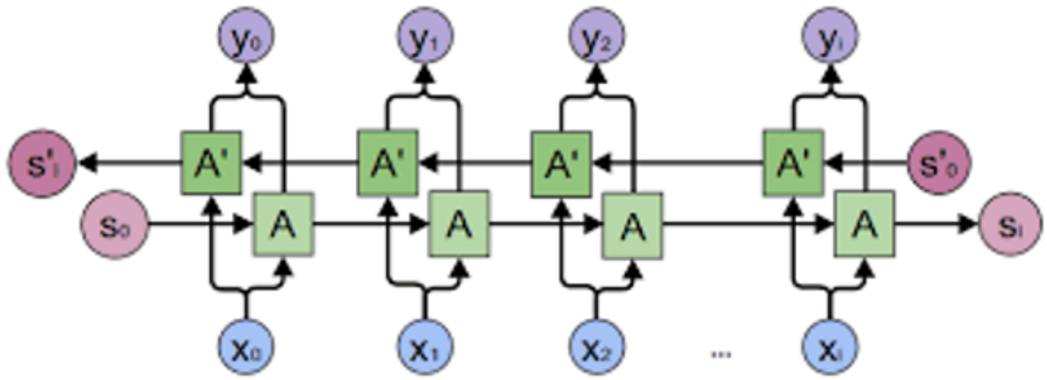


Fig. 3-6 Bi-directional LSTM

In the context of detecting clickbait, a BILSTM algorithm can be employed to analyze the textual content of headlines or articles and predict whether they exhibit clickbait characteristics. Following the huge significance attached to clickbait detection, combating misinformation, enhancing user experience and content moderation benefits are conceivable. The application of a BILSTM algorithm in clickbait detection in this regard offers considerable benefits. For instance, (Jang et al., 2020) outline that BILSTM can capture contextual dependencies between adjacent words. Hence, by applying the BILSTM algorithm in this context, it was possible to capture complex associations between words in an article or headline. Therefore, by applying BILSTM, the detection model was able to understand better the semantic meaning and identify clickbait content patterns.

In another instance, benefits such as the capacity to keep information over longer sequences are attributed to BILSTM and this is because of the contained LSTM units (Singh et al., 2023). Therefore, when applied in the context of clickbait detection, the model is well-positioned to recall the necessary dependencies and context spanning numerous words or phrases. (Drawing from Feng et al.'s, 2020) study, it can be noted that BILSTMs are well poised to effectively handle variable-length sequences. This plays a vital role in clickbait detection as clickbait articles and headlines often vary in length. Furthermore, Feng and others argued that BILSTMs are capable of processing all the input sequences and producing predictions according to the overall context instead of depending on a limited context or fixed-size windows (Feng et al., 2020).

(Atila and Sabaz, 2022) listed advantages such as automatic learning using the necessary input data and the elimination of manual feature engineering as the other benefits of using BILSTM algorithms. That is, BILSTM algorithms provide feature extraction benefits that are instrumental in detecting clickbait and the models can identify various linguistic cues and informative patterns related to clickbait. There are also generalization benefits linked to the application of BILSTM models. For instance, (Sheng et al., 2021) outlined that once trained on representative and diverse datasets, BILSTM models are effective in generalizing hidden news posts or clickbait examples irrespective of whether new clickbait methods or they show various linguistic styles. In other words, by applying a BILSTM algorithm, clickbait detection models' performance and accuracy can be enhanced because of the capturing of global and local dependencies and enhanced sequence modelling capabilities.

3.5.4 Gated Recurrent Unit (GRU)

According to (Wang et al., 2022), GRU is a variant of the conventional RNN that enhances the modeling of long-term relationships in sequential data by addressing the problem of vanishing gradients. The gating mechanism is a component of the GRU algorithm that regulates information flow throughout the network (Wang et al., 2022). According to (Wang et al., 2019), the GRU comprises of:

1) Update Gate: (Wang et al., 2019) opine that how much of the prior hidden state is to be transferred to the current time step is decided by the update gate. It uses a sigmoid activation function to combine the input from the most recent hidden state and the input from the current time step. This produces an update gate value between 0 and 1. A number of 0 indicates that no data from the previous hidden state is transferred, while a value of 1 indicates that all data is maintained.

2) Reset Gate: (Wang et al., 2019) also noted that the amount of the previous hidden state that should be erased and not used in calculating the current hidden state is decided by the reset gate. To determine the value of the reset gate, the input and the previous concealed state are used in the same manner as the update gate. This is done applying a sigmoid activation function.

The reset gate value, input, and prior hidden state are added to calculate the updated hidden state. The update gate decides how much new input information should be added to the updated hidden state. The reset gate controls which portion of the prior hidden state is used. Figure 3.7 shows the GRU structure and its gates. In comparison to conventional RNNs, the GRU algorithm enables effective training and enhanced capture of long-term dependencies (Wang et al., 2022). Gate mechanisms mitigate the vanishing gradient problem by allowing the network to choose whether to update or forget information based on inputs and hidden states currently in effect.

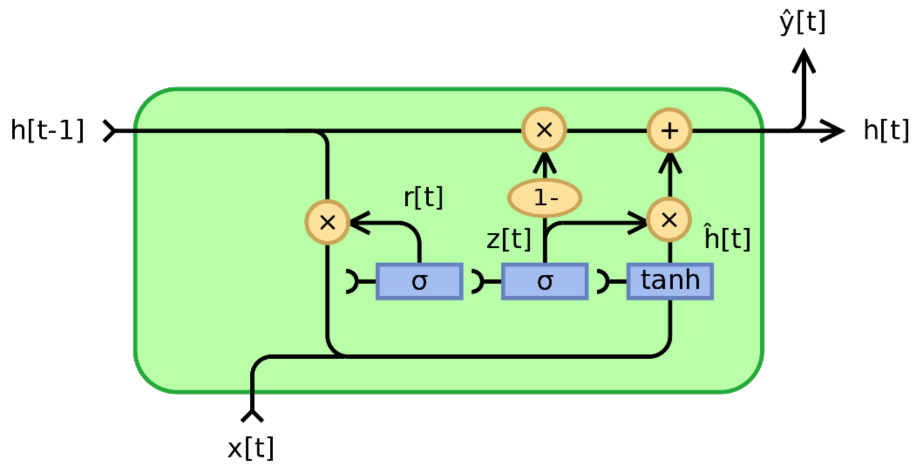


Fig. 3-7 GRU

3.5.5 CNN

According to (Deperlioglu et al., 2022), Convolutional Neural Network (CNN) is a deep learning model specifically designed for processing structured grid-like data like images. Because of their inherent nature, CNNs have witnessed increasingly high and successful application in numerous computer vision tasks such as image segmentation (Bullock, Cuesta-Lázaro and Quera-Bofarull, 2019), object detection (Sultana, Sufian and Dutta, 2020), and image classification (Yadav and Jadhav, 2019). By nature, CNNs tend to use a series of fully connected layers, pooling layers and convolutional layers to automatically learn the input data's hierarchical representations.

(Tiwari et al., 2020) outline that convolutional layers contain filters found inside the input image that use convolution operations to extract local features. In that regard, translational invariance is provided by lowering the computational complexity, down sampling the spatial dimensions and pooling layers as shown in Figure 3.8. As such, a word embedding layer is necessary for text CNN to operate effectively.

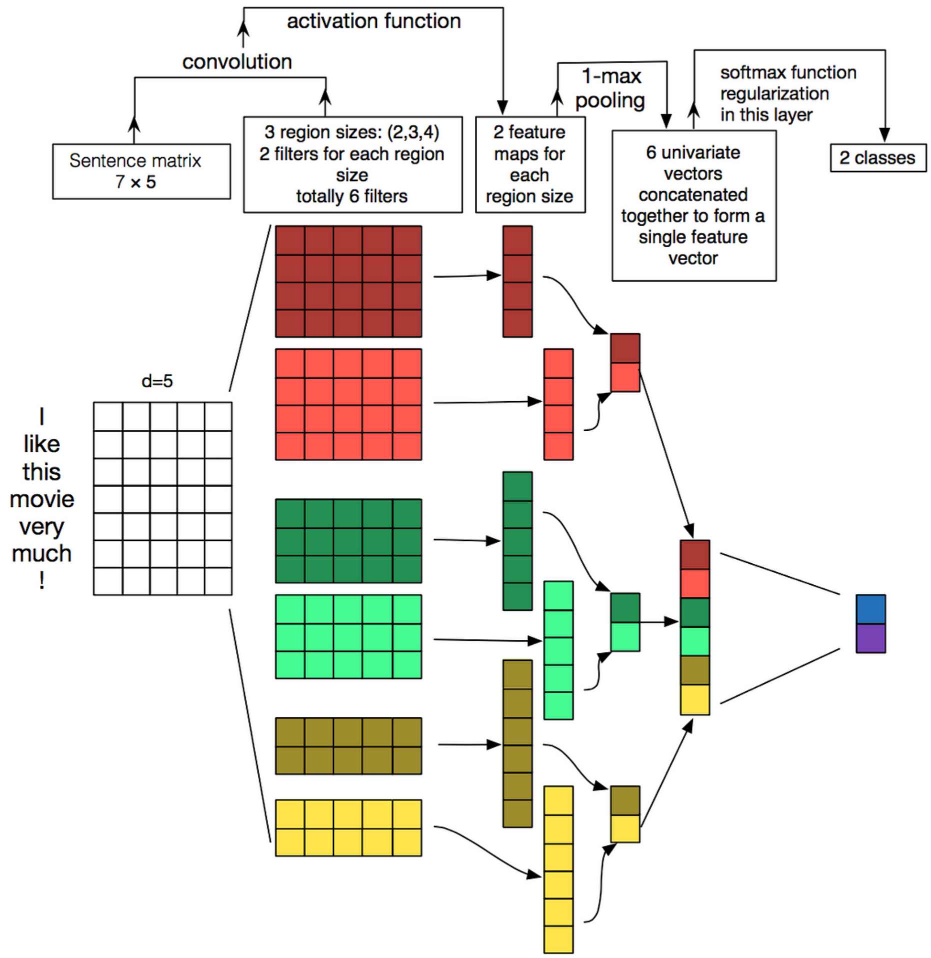


Fig. 3-8 CNN structure.

CNNs were applied in this study because of the immense benefits they provide, especially in detecting clickbait. Drawing from (Yuan et al.'s, 2020) outline, CNNs are regarded as having excellent local automatic feature-extracting abilities. This entails that when used in detecting clickbait, CNNs will use the provided raw input data and automatically extract the necessary clickbait content's features, patterns and visual cues. Benefits such as translation invariance are conceivable when CNNs are applied in clickbait detection exercises. This is because naturally, CNNs contain inherent translation invariance (Tiwari et al., 2020), which is essential for recognizing features irrespective of their location in the input. Such abilities carry huge empirical weight in clickbait detection as clickbait headlines can appear in various locations within images. The other notable benefit is that CNNs are well posed to handle imperfect or noisy input data (Chen et al., 2019). That is CNNs are robust to noise and this is instrumental in detecting clickbait as clickbait headings may vary in positioning, color, size and font. Furthermore, (Avazov et al., 2022) opine that CNNs are effective in generalizing from a large dataset to find true labels from untrained, unseen examples. Thus, when properly trained using various clickbait and non-clickbait database examples, CNNs can learn to generalize and recognize clickbait features in new situations.

In clickbait detection situations, CNNs will treat clickbait headings as images and converts them into visual representations. As a result, the textual content is considered as pixels. Subsequently, CNN is trained using clickbait and non-clickbait labelled datasets to distinguish clickbait from non-clickbait. However, it is crucial to reckon that the mere act of relying on image content to detect clickbait suffers from a series of limitations that demand solutions. For instance, clickbait exists in different forms like social media posts and text-based articles. As a result, this current study proposed to combine CNN with LSTM, Bi LTSM and a hybrid CNN BILSTM to analyze the 5,000 Real Kurdish news data and the 32,000 unprejudiced headlines.

3.5.6 Hybrid CNN BILSTM

- **Convolutional neural network (CNN):** According to (Ahmad et al., 2020) the CNN is made up of a fully linked layer, pooling and layers of convolution. The CNN may extract latent characteristics from the input data by running convolution and pooling procedures. After extraction, the traits are integrated and added to a completely linked layer. Last but not least, an activation function adds nonlinearity to a neuron's output. In order for CNN to function, the convolution layer is essential (Elyashar, Bendahan and Puzis, 2022). Every convolutional layer has a collection of convolutional kernels to reveal latent characteristics and produce feature maps. The feature mappings in the convolutional layer are triggered using a non-linear function for the end result. The convolutional layer is described by the following phrase

- $ci = f(wi * xi * bi)$ (3.1)

- where xi is the input variable for the convolution layer. In this case, the terms ci stand for Ith's feature map, *wi* for a weight matrix, * for the symbol for the dot product, bi for the bias vector, and f(.) for the activation function. CNNs use the rectified linear unit (ReLU) as their activation function.

- **2. Bidirectional LSTM network:** A bidirectional design can simultaneously extract contextual information in both ways using backwards and forward hidden layers. The BILSTM output of the forward and backward hidden layers as ($ht \rightarrow$ and $ht \leftarrow$), respectively. The output and hidden sequence for each front layer is calculated iteratively from step 1 to step t, whereas the output and hidden sequence for each reverse layer is calculated iteratively from step t to step 1. The outputs of the forward layer and the backward layer were calculated using the usual LSTM. Every element of the output vector Y produced by the BILSTM layer is calculated using Equ. (3.2)

$$Yt = \partial(ht, \longrightarrow ht \leftarrow) \quad (3.2)$$

Where the ($ht \rightarrow$ and $ht \leftarrow$) sequences are both coupled using a function. The function can be an average, concatenated, multiplication, or sum function. The output of the BILSTM layer can be expressed as a vector, where Y = [y1, y2, yt], and yt is the value predicted for the well log at the following depth when employing BILSTM. Recently, several applications for neural networks based on the attention mechanism have yielded encouraging results. The BILSTM implicit state or the BILSTM state from the previous step is used by the attention approach in a BILSTM network to align with the cell state of the input at the current phase. Then, one assesses how closely each intermediate stage is related to the final state. It is advisable to stress vital information during the learning process while ignoring irrelevant aspects to increase the accuracy and effectiveness of forecasts.

3. **CNN-BILSTM-AT hybrid model:** When the values of the logs along depth are viewed as ordered sequences, the BILSTM method transforms into an ideal method to construct synthetic well logging curves due to its capacity to capture information from a sequence of data but also propagate information from adjacent depths with depth-term dependencies. In the meantime, the proposed CNN-BILSTM technique uses a hybrid architecture to extract the advantages of using a CNN and an attention mechanism. A CNN is used in one portion of this architecture to learn about the properties, and a two-layer RNN is employed in the other, together with an attention mechanism, to make feature selections.

- **Hybrid CNN-BILSTM:** Character-level traits were modelled using CNN. In the past, (Santos et al., 2020) successfully extracted character-level features using CNNs. A convolution layer and a max layer were applied to each word to extract character-level properties.

- This study's tests, which involved both networks, used a real-world false news dataset and the proposed CNN-BILSTM network. The layered architecture of the current study’s deep neural network is depicted in Figure 3-9.

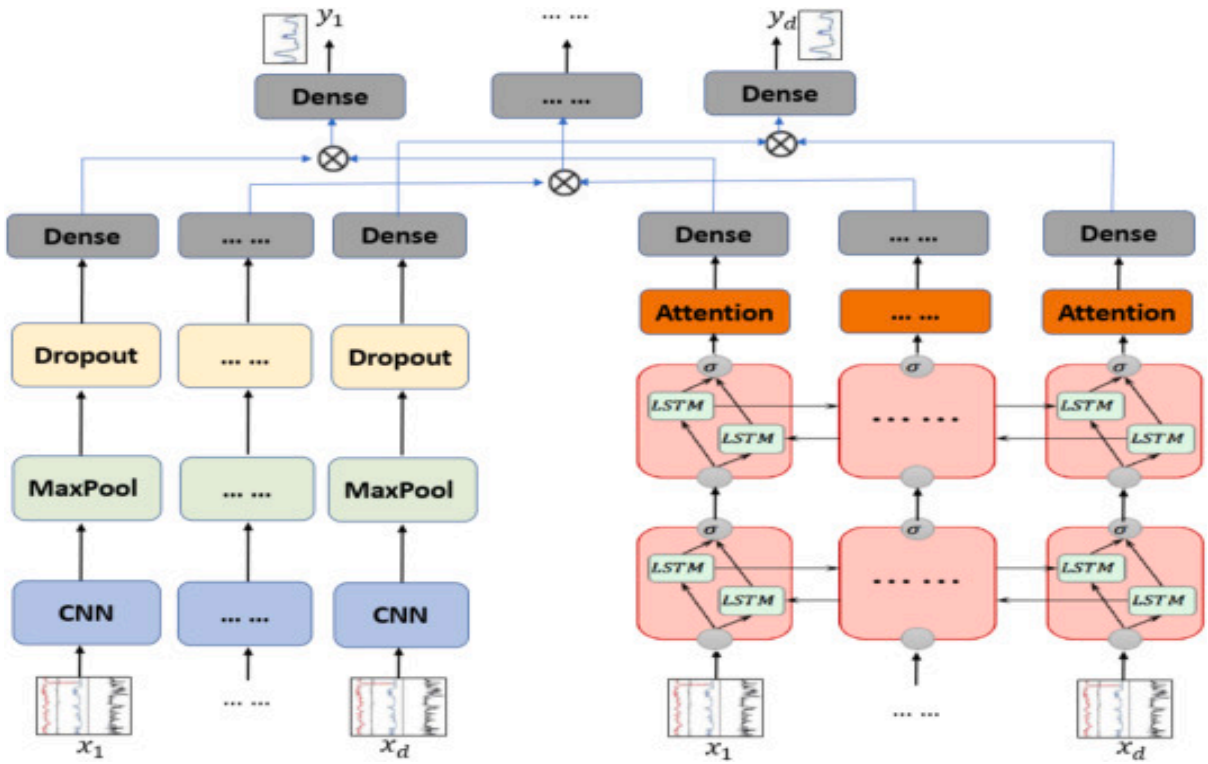


Fig. 3-9 Analysis of models

- This study’s design consists of two layers: a convolutional layer that executes operations based on matrix multiplications and an embedding layer that accepts a vector of 22169-word indices as input. The next step is to add a dropout layer to reduce overfitting. That idea is followed by a 1D convolutional layer with 32 filters, which is frequently used in NLP and binary classification tasks. Each filter creates a feature map in which the neurons triggered in a convolutional layer correspond to particular input word patterns:

- Rectified Linear Unit (ReLU) can be calculated very quickly and simply by comparing the input to the number 0.

- It additionally has a derivative that is either 0 or 1, depending on whether the input is negative or not.
- The layer output will be sent to the max pooling layer. This layer uses a pooling process to calculate the highest, or most significant, value for each patch of each feature map. In order to determine whether the end result is clickbait or not, the CNN output will be passed to a BILSTM layer with 100 neurons and a sigmoid activation function.

3.6 Summary

five algorithms were selected for detecting clickbait in 15,999 clickbait English headlines and 16,001 non-clickbait English headlines, and 5,000 real and 5,000 Fake Kurdish news data. The data went through feature extraction, text classification and text processing procedures. Once completed, clickbait detection performance metrics in the form of accuracy, precision, recall and F1 score were applied. This was essential for determining which deep learning algorithm(s) demonstrate superior performance in clickbait detection for each language.

Chapter Four

Results and Discussion

4.1 Introduction

This chapter provides results established from the English and Kurdish clickbait detection models that were developed using LSTM, BILSTM, CNN, GRU and hybrid CNN BILSTM algorithms. The English data set comprised 15,999 clickbait headlines and 16,001 non-clickbait headlines while the Kurdish fake news detection was carried out using 5,000 Real Kurdish news headlines.

Thus, by satisfying the above-listed research inquiries the study hopes to enhance understanding of the opportunities and challenges in detecting clickbait in both English and Kurdish languages. Consequently, this aids in designing algorithms that are highly effective in addressing clickbait problems and consequences. This chapter shows the results of the study's algorithms and accuracy score of each algorithm.

4.2 English Clickbait Detection Results

The system aims to contextually categorize clickbait titles. word embeddings that are learned from scratch during the model training process are utilized in LSTM, BILSTM, CNN, GRU and CNN BILSTM. The established results are presented in the subsequent sections.

4.2.1 LSTM Results

Foremost, the presented Table 4-1 results indicate that all the 3 LSTMs have a high accuracy rate of 99%. This is higher than the 98% accuracy rate achieved by (Gamage et al.'s, 2021) deep learning LSTM clickbait detection results. Therefore, this indicates that the LSTM models perform extremely well in identifying English clickbait. However, the Double-layer LSTM performed in terms of precision with a precision rate of 95% compared to the Single-layer LSTM and Triple-layer LSTM each with a precision rate of 94%. This implies that identifying positive instances can be effectively achieved using the Double-layer LSTM compared to the Single-layer LSTM and Triple-layer LSTM.

Table 4-1 English LSTM detection performance results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
1	***LSTM***	***One-Layer***	99.0%	94.0%	96.0%	95.0%
	LSTM	***Two-Layer***	99.0%	95.0%	95.0%	95.0%
	LSTM	***Three-Layer***	99.0%	94.0%	95.0%	95.0%

Drawing further, Table 4-1 also shows that the Single-layer LSTM and Triple-layer LSTM are highly capable of capturing all relevant positive instances than the Double-layer LSTM. In support of this notion, the provided results show that the Single-layer LSTM, the Double-layer LSTM and Triple-layer LSTM have recall rates of 96%, 95% and 95%, respectively. Thus, it can be inferred that there is a consistent performance in terms of both identifying positive instances (precision) and capturing all relevant positive instances (recall).

In overall, the applied LSTM models correctly classified English clickbait as evidenced by their high accuracy rate. Additionally, the applied LSTM models are capable of identifying positive instances accurately while minimizing false positives and negatives. However, considerable attention is drawn towards the Double-layer LSTM, which is highly effective in correctly identifying positive instances than the Single-layer and Triple-layer LSTMs. But all the relevant positive instances can be highly captured by the Single-layer and Triple-layer LSTMs.

4.2.2 BILSTM Results

Following the establishment of the first algorithm results, the study proceeded further to determine the performance of the LSTM algorithm. Using the provided Table 4-2 results, it can be noted that the three BILSTM layers are highly accurate in detecting clickbait as evidenced by their accuracy rates, which range from 99% to

99.6%. As a result, the BILSTM algorithm can be said to be highly accurate in correctly classifying clickbait instances. However, both model results are higher than the 98% accuracy rate established by (Dimpas, Po and Sabellano, 2017).

Table 4-2 English BILSTM detection performance results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
2	***BILSTM***	***One-Layer***	99.0%	96.0%	94.0%	95.0%
	BILSTM	***Two-Layer***	99.1%	95.0%	94.0%	95.0%
	BILSTM	***Three-Layer***	99.6%	95.0%	95.0%	95.0%

Furthermore, it can be established from Table 4-2 that the BILSTM models can correctly identify clickbait instances as evidenced by their precision results that ranged from 95% to 96%. With recall rates ranging from 94% to 95%, the BILSTM models effectively captured several necessary clickbait instances. Lastly, consistent F1 scores of 95% were observed across the single, double and triple-layer BILSTM models. This shows that there was a good balance between precision and recall.

In summary, the BILSTM model results denote that the model is highly capable of accurately classifying clickbait instances with a minimal number of misclassifications. However, attention is directed to the differences in the precision scores of the three models with the double and triple Layer BILSTM models having slightly lower precision rates of 95%. As a result, such findings may demand that measures be enacted to ensure that the selected configuration layer does not significantly affect the BILSTM's overall performance in terms of accuracy, precision, recall, and F1 score. Additionally, factors such as model complexity and computational resources can influence the decision to use a specific configuration layer. Hence, to make rational decisions about the BILSTM optimal configuration for clickbait detection, factors such as specific requirements and dataset characteristics may require immediate consideration. Nonetheless, further tests were conducted to ascertain further the performance of existing algorithms in detecting English clickbait. To accomplish such a task, the next section analyses the CNN model results.

4.2.3 CNN Results

Concerning the CNN results, accuracy rates ranging from 98.9% to 99.1% were established. Hence, by implication, Table 4-3 results indicate that the single, double and triple Layer CNN models were highly accurate in correctly classifying English clickbait instances. The current study's accuracy rates are higher than Zheng et al.'s (2018) accuracy rate of 78.18%. Furthermore, in terms of precision, the developed 3 CNN models ranked better than (Zheng et al.'s, 2018) precision rate of 73.37%, which is lower than the established rates of 93%, 95% and 95%, respectively.

Table 4-3 English CNN detection performance results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
3	***CNN***	***One-Layer***	98.9%	93.0%	96.0%	94.0%
	CNN	***Two-Layer***	99.1%	95.0%	96.0%	95.0%
	CNN	***Three-Layer***	99.1%	95.0%	96.0%	94.0%

Concerning the models' recall rates, the Single-layer CNN, the Double-layer CNN and Triple-layer CNN had recall rates of 96% each. This implies that the models effectively captured several necessary clickbait instances. Again, this is higher than a recall rate of 86.48% established by (Zheng et al.'s, 2018) in a related examination. Table 4-3 also shows that there was a good balance between precision and recall with the Single-layer CNN, the Double-layer CNN and Triple-layer CNN having F1 scores of 94%, 95% and 94%, respectively.

4.2.4 Hybrid CNN BILSTM Results

As part of the study's algorithm performance assessment, Hybrid CNN BILSTM examinations were also conducted on the 15,999 English clickbait headlines. All the 3 CNN BILSTM models were highly accurate in correctly classifying English clickbait instances. Supporting evidence presented in Table 4-4 shows accuracy rates of 99.1% (the Single-layer CNN BILSTM), 98.9% (the Double-layer CNN BILSTM) and 98.9% (Triple-layer CNN BILSTM). The accompanying precision rates of 96% (the Single-layer CNN BILSTM), 91% (the Double-layer CNN BILSTM) and 95% (Triple-layer CNN BILSTM) were also observed. Related prior examinations achieved accuracy rates of 95% when machine learning methods for fake news detection were applied (Khan et al., 2019) and 96% within the context of Bangla fake news detection using hybrid deep learning models.

Table 4-4 English hybrid CNN BILSTM detection performance results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
4	***CNN BILSTM***	***One-Layer***	99.1%	96.0%	94.0%	95.0%
	CNN BILSTM	***Two-Layer***	98.9%	91.0%	98.0%	94.0%
	CNN BILSTM	***Three-Layer***	98.9%	95.0%	95.0%	95.0%

Recall rates of 94% (the Single-layer CNN BILSTM), 98% (the Double-layer CNN BILSTM) and 95% (Triple-layer CNN BILSTM) were achieved. This denotes that the Single-layer CNN BILSTM model effectively captured several necessary clickbait instances than the Triple-layer CNN BILSTM and the Double-layer CNN BILSTM orderly. However, good balances between precision and recall were highly achieved through the application of the Single-layer CNN BILSTM algorithms.

4.2.5 GRU Results

As part of the study's algorithm performance assessment, GRU examinations were also conducted on the 15,999 English clickbait headlines. All the 3 GRU models were highly accurate in correctly classifying English clickbait instances. Supporting evidence presented in Table 4-5 shows accuracy rates of 93.1% (the Single-layer GRU), 94.5% (the Double-layer GRU) and 97.0% (Triple-layer GRU). The accompanying precision rates of 96% (the Single-layer GRU), 91% (the Double-layer GRU) and 95% (Triple-layer GRU) were also observed. Related prior examinations achieved accuracy rates of 95% when machine learning methods for fake news detection were applied (Khan et al., 2019) and 96% within the context of Bangla fake news detection using deep learning models.

Table 4-5 English GRU detection performance results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
5	***GRU***	***One-Layer***	93.1%	96.0%	94.0%	95.0%
	GRU	***Two-Layer***	94.5%	91.0%	98.0%	94.0%
	GRU	***Three-Layer***	97.0%	95.0%	95.0%	95.0%

Recall rates of 94% (the Single-layer GRU), 98% (the Double-layer GRU) and 95% (Triple-layer GRU) were achieved. This denotes that the Triple -layer GRU model effectively captured several necessary clickbait instances than the Single-layer GRU and the Double-layer GRU orderly. However, good balances between precision and recall were highly achieved through the application of the Triple -layer GRU algorithms.

4.2.6 Model Comparisons

Following the established LSTM, BILSTM, CNN, CNN BILSTM and hybrid CNN BILSTM algorithms results, the findings infer that CNN BILSTM model with pre-trained GloVe capture much more context. The findings also uncover that BILSTM models (accuracy=99.23%) outperform the CNN (accuracy=99.03%), LSTM (accuracy=99%) and Hybrid CNN BILSTM (accuracy=98.97%) algorithms in terms of accuracy. In comparison, both models applied in this study have higher accuracy rates compared to other related studies such as Gamage et al.'s (2021) deep learning LSTM results that achieved an accuracy rate of 98%. Similarly, when weighed in terms of precision, the current study's findings revealed that BILSTM models (precision = 95.33%) outperforms the CNN (precision= 94.33%), LSTM (precision = 94.33%) and Hybrid CNN BILSTM (precision = 94.00%). Such findings are exceptionally higher than (Zheng et al.'s, 2018) precision rate of 73.37%. Hence, such findings reinforce scientific efforts involving the application of machine learning classification methods, which have risen in prominence in identifying clickbait (Al-Sarem et al., 2021) (Pujahari and Sisodia, 2021). As such, the current study compares the Random Forest, Naïve Bayesian and other machine learning methods. Meanwhile, similar tasks can use a task-specific embedding that is trained and saved using large datasets. The datasets are then applied to new situations because of pre-trained word embeddings, which are another form of transfer learning.

By comparing the algorithms in terms of highly capturing all relevant positive instances, the CNN model (recall=96%) outperforms the Hybrid CNN BILSTM (recall=95.66%), LSTM (recall = 95.33%) and BILSTM models (recall=94.33%).

When analyzed in terms of establishing a good balance between precision and recall, LSTM (F1 score=95%) and BILSTM models (F1 score=95%) algorithms outperform the Hybrid CNN BILSTM (F1 score l=94.66%) and the CNN model (F1 score=94.33%). Therefore, the study upholds that higher consistent performance in terms of both identifying positive instances (precision) and capturing all relevant positive instances (recall) is effectively established when the LSTM and BILSTM algorithms are applied.

Meanwhile, the embedding layer is the network's initial layer and the ANN is created using the LSTM. As a result, the LSTM's integration allows the network to meaningfully represent words by enhancing each token's vector. Typical word choices like where, who and which are often used when the main phrase's pronouns or subjects are to be referred to in an embedded clause. In certain cases, questions can be part of a statement and when a person is asked these types of questions, they are termed embedded questions and can be used to further discuss other questions. On the other hand, asking a question indirectly is polite. In contrast, embedded questions are worded the same as regular questions. Questions within a statement are punctuated the same as regular sentences. Instead of using a question mark, a particular inquiry's conclusion is indicated with a period.

4.3 Kurdish Clickbait Detection Results

As part of testing the performance of LSTM, BILSTM, CNN, GRU and hybrid CNN BILSTM algorithms in detecting Kurdish fake news, 5,000 Real and 5,000 Fake Kurdish news headlines were utilized. As a result, this section of the study presents the established performance results of the five algorithms as follows;

4.3.1 LSTM Results

Commencing with the LSTM, though high, the 3 LSTMs performed relatively lower than Gamage et al.'s (2021) deep learning LSTM clickbait detection results that achieved an accuracy rate of 99%. As such, the Double-layer LSTM outperformed the Single-layer LSTM and Triple-layer LSTM in terms of prediction accuracy with accuracy rates of 94.7%, 93.7% and 93% being achieved respectively.

Furthermore, similar findings are observable when comparisons are made with respect to the algorithms' precision. That is, the Double-layer LSTM (precision=96.7%) identified positive instances in Kurdish headlines more effectively compared to the Single-layer LSTM and Triple-layer LSTM that registered precision rates of 92.9% and 90.9%, respectively as indicated in Table 4-6.

Table 4-6 Kurdish LSTM Detection Performance Results

	Algorithm	Layers	Accuracy	Precision	Recall	F1 score
1	***LSTM***	***One-Layer***	93.0%	90.9%	95.5%	93.1%
	LSTM	***Two-Layer***	94.7%	96.7%	92.4%	94.5%
	LSTM	***Three-Layer***	93.7%	92.9%	94.5%	93.7%

By using recall as the detection performance metric, the provided Table 4-6 results show that the Single-layer LSTM (recall=95.5%) and the Triple-layer LSTM (recall=94.5%) are highly capable of capturing all relevant positive instances than the Double-layer LSTM (recall=92.4%). However, the Single-layer LSTM (F1 score=94.5%) performs better in terms of a good balance between precision and recall. The Triple-layer LSTM ranks second with an F1 score of 93.7% followed by the Single-layer LSTM with an F1 score of 93.1%. Nonetheless, given the fact that both models' F1 scores are above 93%, it can, therefore, be inferred that the LSTM algorithm or model achieves a higher consistent level of performance in terms of both identifying positive instances (precision) and capturing all relevant positive instances (recall) in Kurdish headlines. By using a similar approach that was applied in testing the algorithms' performance in detecting clickbait in English headlines, LSTM was examined next.

4.3.2 BILSTM results

After subsequent examinations that were carried out to test the performance of the BILSTM algorithm in detecting fake Kurdish news, lower accuracy rates of 94.8% (Triple-layer BILSTM), 93.6% (Double-layer BILSTM) and 92.5% (Single-layer BILSTM) were recorded. The established accuracy rates are lower than the 98% accuracy rate established by (Dimpas, Po and Sabellano, 2017). However, the fact that such accuracy rates surpass the 90%-mark entails that the BILSTM algorithm or model is highly accurate in detecting clickbait in the Kurdish language. When analyzed in terms of precision, Table 4-7 shows that the Triple-layer BILSTM (precision=96.2%) correctly identified clickbait instances in the Kurdish language better than the Single-layer BILSTM (precision=92.9%) and the Double-layer BILSTM (precision=92.8%). Again, an accuracy of more than 90% entails that the overall performance of the BILSTM model is extremely high in terms of correctly identified clickbait instances in the Kurdish language.

Table 4-7 Kurdish BILSTM detection performance results

	Algorithm	**Layers**	**Accuracy**	**Precision**	**Recall**	**F1 score**
2	***BILSTM***	***One-Layer***	92.5%	92.9%	92.0%	92.4%
	BILSTM	***Two-Layer***	93.6%	92.8%	94.5%	93.6%
	BILSTM	***Three-Layer***	94.8%	96.2%	93.2%	94.6%

With recall rates ranging from 92% to 94.5%, the study outlines that the Single-layer, Double-layer and Triple-later BILSTMs are highly capable of capturing all relevant positive instances in the Kurdish language. In that regard, the Double-layer LSTM (recall=94.5) ranks first followed by the Triple-layer LSTM (recall=93.2%) and the Single-layer BILSTM (recall=92%).

Differences are observed when the F1 score metric is used to compare the three models' performance. That is, the Triple-layer LSTM (F1 score=94.6%)

achieves a good balance between precision and recall compared to the Double-layer LSTM (F1 score=93.6%) and Single-layer LSTM (F1 score=92.4%). Nonetheless, given the fact that both models' F1 scores are above 90%, the study upholds that the BILSTM algorithm or model achieves a higher consistent level of performance in terms of both identifying positive instances (precision) and capturing all relevant positive instances (recall) in Kurdish headlines.

4.3.3 GRU

After achieving higher clickbait detection performance results using the BILSTM model, the study proceeded further to examine the GRU's performance using similar metrics. As presented in Table 4-8, the Single-layer GRU (accuracy=94.3%) and the Double-layer GRU (accuracy=94.3%) recorded higher levels of accuracy compared to the Triple-layer GRU (accuracy=93.2%). Despite, the fact that these accuracy rates are more than 90%, they are lower than the 98% accuracy rate established by (Dimpas, Po and Sabellano, 2017). This implies that the GRU encounters challenges that undermine its accuracy when used to detect Kurdish clickbait. However, this does not discount the fact that the high accuracy rate of the entire GRU model entails that the model is highly accurate in detecting clickbait in the Kurdish language.

Table 4-8 Kurdish GRU detection performance results

	Algorithm	**Layers**	**Accuracy**	**Precision**	**Recall**	**F1 score**
3	***GRU***	***One-Layer***	94.3%	94.7%	94.0%	94.4%
	GRU	***Two-Layer***	94.3%	93.3%	95.8%	94.5%
	GRU	***Three-Layer***	93.2%	91.4%	95.7%	93.5%

The Single-layer GRU assumes a first position when analyzed in terms of precision with a precision rate of 94.7% compared to 93.3% and 91.4% achieved when the Double-layer GRU and the Triple-layer GRU are applied. Table 4-8 also shows that the Double-layer GRU (recall=95.8%) followed by the Triple-layer GRU (recall=95.7%) and the Single-layer GRU (recall=94%) are highly capable of capturing all relevant positive instances in the Kurdish language. In terms of achieving a good balance between precision and recall, Double-layer GRU (F1 score=94.5%) outperforms both the Triple-layer GRU (F1 score=93.5%) and the Single-layer GRU (F1 score=94.4%).

4.3.4 CNN Results

An assessment of algorithms in detecting clickbait in the Kurdish language was further carried out using the established CNN results. The single-layer CNN (accuracy=93.7%) and Triple-layer CNN (accuracy=93.7%) models outperformed the Double-layer CNN (accuracy=93.3%). Thus, Table 4-9 results indicate that attempts to correctly classify Kurdish clickbait instances are highly achievable when the Single-layer CNN is applied. Similarly, the same results also indicate that the Single-layer CNN (precision=93.9%) is effective in identifying positive instances in the Kurdish language compared to the Triple-layer CNN (precision=93.1%) and the Double-layer CNN (precision=91.3%).

Table 4-9 Kurdish CNN detection performance results

	Algorithm	**Layers**	**Accuracy**	**Precision**	**Recall**	**F1 score**
4	***CNN***	***One-Layer***	93.7%	93.9%	92.9%	93.4%
	CNN	***Two-Layer***	93.3%	91.3%	95.2%	93.2%
	CNN	***Three-Layer***	93.7%	93.1%	93.8%	93.5%

Like the English CNN model, the Kurdish CNN model results have higher accuracy rates than (Zheng et al.'s, 2018) accuracy rate of 78.18%. The model layer interchangeably provides different results when their performance is compared in terms of recall and F1 scores. That is, the Double-layer CNN (recall=95.2%) outperforms the Triple-layer CNN (recall=93.8%) and the Single-layer CNN (recall=92.9%). That is, Triple-layer CNN (F1 score=93.5%) outperforms outperforms the Single-layer CNN (F1 score=93.4%) and the Double-layer CNN (F1 score=93.2%). This implies that the Double-layer CNN model effectively captured several necessary clickbait instances compared to the Triple-layer CNN and the Single-layer CNN models. In addition, a good balance between precision and recall was effectively achieved by using the Triple-layer CNN as depicted in Table 4-9.

4.3.5 Hybrid CNN BILSTM Results

Having tested the other four the study proceeded to examine the performance of the hybrid CNN BILSTM model in detecting Kurdish clickbait. As such, Table 4-10 denotes that the Triple-layer CNN BILSTM is highly preferable as evidenced by its high accuracy rate of 93.5%. Similarly, the Triple-layer CNN BILSTM gains huge favor in identifying positive instances in the Kurdish language (precision= 92.7%). Further inferences drawn from Table 4-10 also show that using the Double-layer CNN BILSTM enhances effectiveness in capturing all relevant positive instances in the Kurdish language (recall=96.4%). However, a good balance between precision and recall is effectively achieved by applying the Triple-layer CNN BILSTM (F1 score=93.6%) depicted in Table 4-10.

Table 4-10 Kurdish hybrid CNN BILSTM detection performance results

	Algorithm	**Layers**	**Accuracy**	**Precision**	**Recall**	**F1 score**
5	***CNN BILSTM***	***One-Layer***	93.0%	92.1%	94.3%	93.4%
	CNN BILSTM	***Two-Layer***	91.4%	87.8%	96.4%	91.9%
	CNN BILSTM	***Three-Layer***	93.5%	92.7%	94.4%	93.6%

In overall, all the performance metrics average around 90% and this suggests good performance in detecting clickbait in the Kurdish language. Having evaluated the performance of all five models, the next section of the study proceeds to compare their performance in detecting clickbait in the Kurdish language. Such comparisons set a stage for effectively comparing both models' performance in detecting clickbait in both English and Kurdish languages.

4.3.6 Model Comparisons

Following the application of the LSTM, BILSTM, GRU, CNN and hybrid CNN BILSTM algorithms in detecting fake Kurdish news, a decision was made to choose the best-performing algorithm. Performance-wise, the GRU ranks higher (accuracy=93.9%) than other the LSTM (accuracy=93.8%), BILSTM (accuracy=93.6%), CNN (accuracy=93.6%) and hybrid CNN BILSTM (accuracy=92.6%) algorithms. This entails that when it comes to extremely identifying well Kurdish clickbait, the GRU offers optimal detection results. Figure 4-1 provides pictorial insights into the models' accuracies.

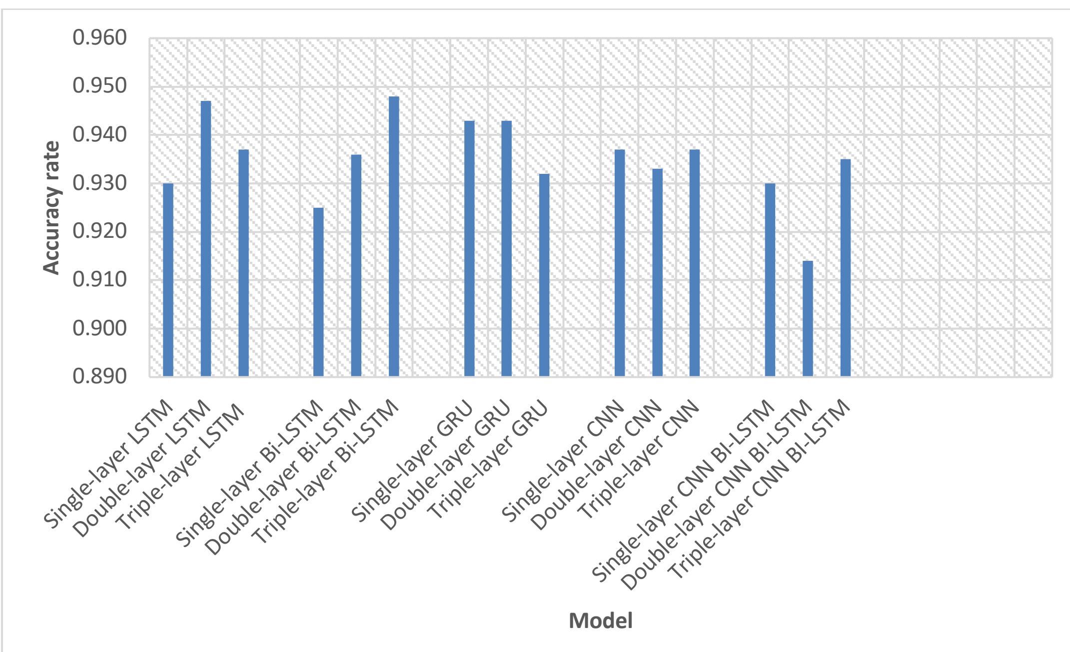


Fig. 4-1 Summary of the algorithms' accuracy performance metric results

Significant achievements in accuracy rates are high when the Single-layer GRU and the Double-layer GRU are applied as evidenced by accuracy rates of 93.8%. From this examination, the LSTM is recommended as the second-best alternative. Furthermore, the current study's applied algorithms offer superior clickbait detection accuracy compared to other previous related examinations. For instance, (López-Sánchez et al., 2018) achieved an accuracy rate of 88.58% using the Naïve Bayes, 87.58% with logistic regression and 88.78% with SVM. As such, the current study recorded an accuracy rate of 90.65% using Logistic Regression, 92.35% with SVM and 92.9% using the Naïve Bayes.

Different inferences and decisions can be made when other detection performance metrics are put into consideration. For instance, by taking into consideration of the algorithms' precision, the BILSTM offers superior detection precision (precision = 94%). The CNN ranks second (precision = 93.6%) followed by the LSTM (precision = 93.5%), the GRU (precision = 93.1%) and the hybrid CNN BILSTM. Given that both algorithms have precision rates that are higher than 90%, it can be inferred that the algorithms correctly identified clickbait instances in the Kurdish language.

Additional inferences concerning differences in the algorithms' performance are established when recall rates are considered. That is, the GRU has a higher recall rate of 95.2%. This is higher than LSTM (recall rate = 94.1%), BILSTM (recall rate = 93.2%), CNN (recall rate = 92.8%) and hybrid CNN BILSTM (recall rate = 90.9%). In this regard, the findings imply that the GRU is highly capable of capturing all relevant positive instances in the Kurdish language compared to other algorithms. Contributing to the GRU's higher recall rates is the Double-layer GRU (recall=95.8%) followed by the Triple-layer GRU (recall=95.7%). When compared to other studies, these recall rate performance metrics are higher than recall rates established in prior studies.

Other differences in the algorithm's clickbait detection performance can be established by applying the F1 score. Given the fact that both models' F1 scores are above 90%, the study upholds that the GRU algorithm achieves a higher consistent level of performance in terms of both identifying positive instances (precision) and capturing all relevant positive instances (recall) in Kurdish headlines. Though relatively lower than that of the GRU algorithm (F1 score = 94.1%), F1 scores of 93.8% (LSTM), 93.8% (BILSTM), 93.4% (CNN) and 93% (hybrid CNN BILSTM) were recorded. These scores are higher than F1 scores of 66% and 76% (Nadia and Iswanto, 2021) and 79% (Khater et al., 2019) (Klairith and Tanachutiwat, 2018).

In overall, with accuracy, precision, recall and F1 score performance metrics of more than 90% being recorded, the study therefore upholds that the applied LSTM, BILSTM, CNN, GRU and hybrid CNN BILSTM algorithms are effective in detecting clickbait in Kurdish languages.

The accuracy, Val-accuracy and validation-loss of the LSTM model epochs are provided in Figures 4-2 to 4-6. According to Figures 4-2 to 4-6, the LSTM's model's accuracy exceeds the Val-accuracy, loss and Val-loss at epoch values of at least 0.5. The accuracy rate rises to 1 when the epoch value reaches. Hence, the LSTM can be said to be a higher performance, generalization capability and convergence of learning and predicting Clickbait in the Kurdish language.

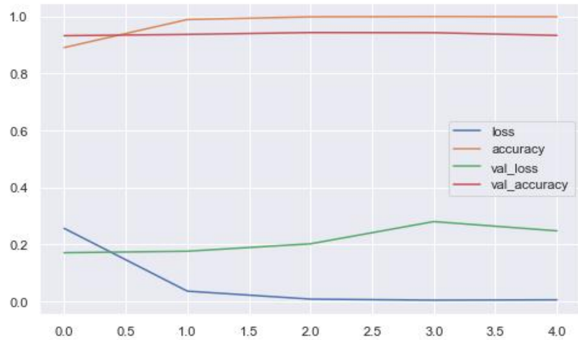


Fig. 4-2 accuracy, Val-accuracy and validation-loss results for LSTM algorithm

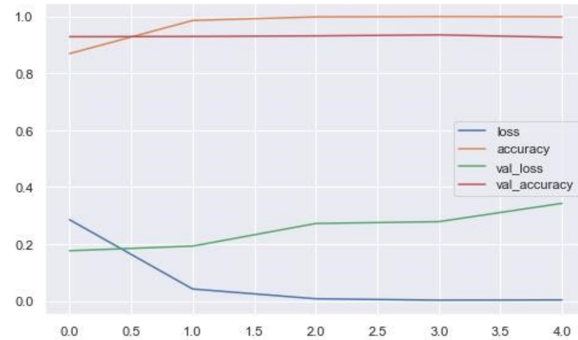


Fig. 4-3 accuracy, Val-accuracy and validation-loss results for CNN algorithm

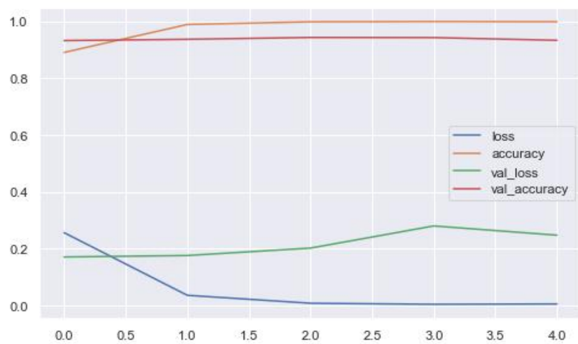


Fig. 4-4 accuracy, Val-accuracy and validation-loss results for GRU algorithm

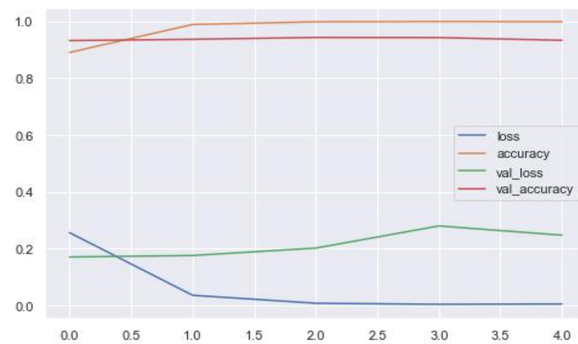


Fig. 4-5 accuracy, Val-accuracy and validation-loss results for BILSTM algorithm

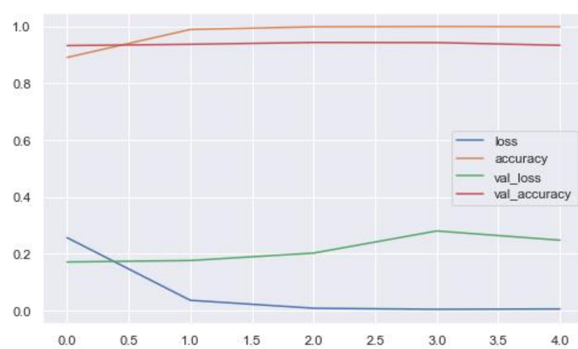


Fig. 4-6 accuracy, Val-accuracy and validation-loss results for CNN-BILSTM algorithm

4.4 Discussion

Study discussions were made concerning observations made by the researcher during the data capturing, recording, training and model estimation processes. As such, these instances were compared and contrasted with existing studies on clickbait detection in English and Kurdish languages. The observed findings are discussed in the next section as follows:

4.4.1 Cultural Nuances, Linguistic Patterns, And Language-Specific Characteristics Influencing Clickbait Detection

Following the study's second objective, cultural nuances, linguistic patterns, and language-specific characteristics that influence clickbait detection in English and Kurdish languages were identified. It is through the machine learning and model estimation process that these nuances, patterns and characteristics were identified. Additionally, existing knowledge on clickbait detection issues and challenges was applied to aid in determining the languages nuances, patterns and characteristics influencing clickbait detection in English and Kurdish languages.

Commencing with cultural nuances, their influence on clickbait detection is observed through contextual relevance and sensitivity and taboos. It is vital to note that for both English and Kurdish headlines to appeal to specific cultural groups idiomatic expressions, symbols, and cultural references may be used in such instances. In support of this notion, (Palau-Sampio, 2016) contends that clickbait should have contextual relevance. Furthermore, (Lischka and Garz, 2021) believe that understanding the cultural context and whether the content aligns with the expectations and values of the target audience is necessary for detecting clickbait. Such is instrumental, especially if they are to effectively achieve higher click rates. Adding further are sensitivity and taboos. With reference to both English and Kurdish cultures, it is vital to note that both cultures have varying taboos and sensitivities. This can be reinforced by Brady, (Crockett and Van Bavel's, 2020) established ideas denoting that Clickbait headlines that violate or exploit people's cultural norms are highly effective in eliciting attention and engagement. Hence, it can be inferred from these examinations that it is important to understand how certain topics might impact different cultural groups in order to detect clickbait.

Concerning linguistic patterns' influence on clickbait detection, three influences in the form of sensational language, wordplay and puns and localized keywords were identified in both English and Kurdish clickbait detection processes. The initial foundation upon which such examinations will be conducted is derived from (Biyani, Tsioutsiouliklis and Blackmer's, 2016) study suggestions. That is, they opined that Clickbait headlines tend to use sensational or exaggerated language to capture people's attention (Biyani, Tsioutsiouliklis and Blackmer, 2016). However, it is critical to note that such linguistic patterns can vary across cultures and languages. Hence, detecting clickbait requires identifying linguistic features such as emotional triggers, superlatives, and hyperbole that are specific to Kurdish and English languages. This also includes Localized keywords, Wordplay and puns as clickbait often incorporate localized phrases or keywords and employ clever puns or wordplay to pique curiosity. Hence, understanding the specific phrases or keywords used in clickbait and recognizing linguistic patterns related to wordplay can help in detecting clickbait.

Concerning language-specific characteristics, the study upholds that sentence structure, grammar and syntax, and tone and style influence the construction of clickbait. Such established are highly sidelined in previous examinations such as machine learning models (Ahmad et al., 2020), distributed learning (Altheneyan and Alhadlaq, 2023), and multivariate time series (Bianchi et al., 2019). However, similarities were observed in (Bronakowski, Al-Khassaweneh and Al Bataineh,

2023) contemporary examination entitled, "*Automatic detection of clickbait headlines using semantic analysis and machine learning techniques*". Amid such discoveries, the current study upholds that English and Kurdish languages have different sentence structures. These structures tend to influence how clickbait headlines are constructed and detected. Therefore, the ability to analyze such language-specific sentence patterns plays an important role in identifying typical English and Kurdish clickbait structures and their deviations. Along similar lines, specific syntactic or grammatical language features can be exploited by clickbait headlines to create suspense or intrigue. Therefore, it can be easier to spot clickbait if you are aware of language-specific syntactic constructions and grammar rules. Different languages have distinct styles and tones and this affects how clickbait is delivered and detected. Therefore, a language's formal or informal registers, for example, can be used to help discern between authentic content and clickbait. Therefore, understanding the cultural norms, linguistic patterns, and language-specific traits of the target audience is crucial to spotting clickbait across linguistic and cultural boundaries. By including these elements in their algorithms, machine learning models trained on varied datasets that cover a range of languages and cultural contexts might increase clickbait detection precision.

4.4.2 Improvements to Clickbait Detection in Terms of Strengths and Limitations of Each Algorithm

The fifth research objective was to suggest improvements to clickbait detection in both English and Kurdish language, in terms of the strengths and limitations of each algorithm. In light of the provided strengths and weaknesses of the LSTM, solutions were developed to improve the LSTM's performance when detecting clickbait in English and Kurdish languages.

4.3.3 Deep Learning Algorithm Performance and Effective Approach for Clickbait Detection

The third research objective was to develop and implement LSTM, BILSTM, CNN, GRU, and hybrid CNN BILSTM models for clickbait detection in English and Kurdish languages. It is upon the satisfaction of this objective that the study proceeded to achieve the fourth research objective. That is, to train and evaluate the performance of each deep learning algorithm using a large dataset of clickbait and non-clickbait headlines in English and Kurdish languages. Consequently, this will help in identifying the most effective approach for clickbait detection in each language. This is based on the precision, recall, F1 score, and computational efficiency of each algorithm.

Foremost, following the current study's establishments, the BILSTM algorithm ranks the most effective in detecting clickbait in English with an accuracy rate of 99.23%, 95.33% precision, 94.33% recall rate and an F1 score of 95%. Contributing to the algorithm's accuracy are single-layer, double-layer and triple Layer BILSTM accuracy rates of 99%, 99.1% and 99.6%. This implies that the BILSTM algorithm can be said to be highly accurate in correctly classifying clickbait instances.

Both model results are higher than (Dimpas, Po and Sabellano's, 2017) established accuracy rate the of 98% as well as the CNN (accuracy=99.03%), LSTM (accuracy=99%) and Hybrid CNN BILSTM (accuracy=98.97%) algorithms as shown in Figure 4-7. Therefore, it can be inferred that the BILSTM models perform extremely well in identifying English clickbait.

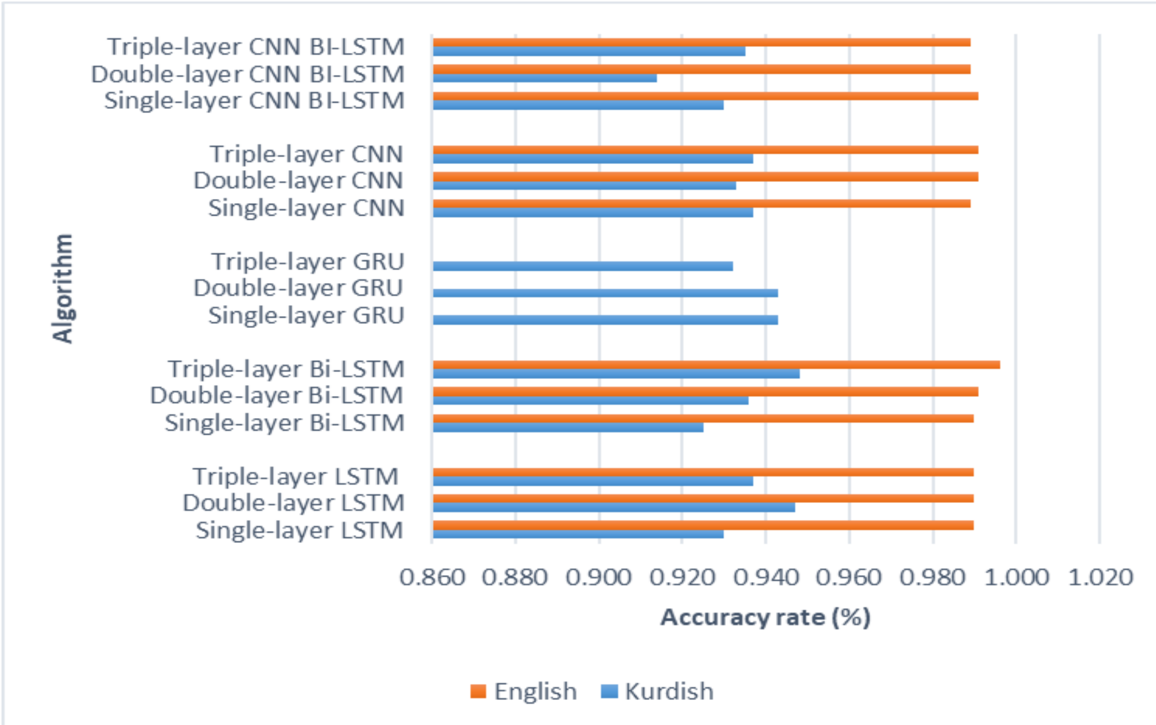


Fig. 4.7 Models Accuracy

Secondly, the detection performance results demonstrated that the GRU algorithm performs better than other algorithms at detecting clickbait in the Kurdish language. It has an accuracy rate of 93.93%, 93.13% precision, 95.17% recall rate and 94.13% F1 score. With a precision of 95.33%, the BILSTM model correctly identified clickbait instances in the English language than the GRU, which had a precision rate of precision, 95.17%. However, the GRU has a higher recall rate of 95.17%, which exceeds the BILSTM's recall rate of 94.33%. This implies the GRU is highly effective in capturing all relevant positive instances in Kurdish compared to the BILSTM captured English clickbait results. The BILSTM (F1 score=95%) model also establishes a good balance between precision and recall than the GRU model (F1 score=94.13%).

When compared with other related models, the study's BILSTM achieves a higher accuracy rate of 99.23% in detecting clickbait in English. This is higher than the Kurdish model's accuracy rate of 93.6% and (Dimpas, Po and Sabellano's, 2017) accuracy rate of 98%. Meanwhile, the current study's LSTM model exhibits a high accuracy rate of 99%, which is higher than the 94% established by (Chakraborty et al., 2017) and 98% by (Gamage et al., 2021). However, the same model performs relatively lower than these studies when used in detecting clickbait in Kurdish with an accuracy rate of 94%. This entails that the LSTM model should be used in detecting clickbait in English than in Kurdish as it performs extremely well in identifying English clickbait.

Table 4-11 The model accuracy in this study is compared with the results obtained by other authors With different dataset

		Observed accuracy values						
		Current study		***(Chakraborty et al, 2017)***	***(Ahmad et al., 2016)***	***(Gamage et al.'s, 2021)***	***(Dimpas, Po and Sabellano, 2017)***	***(Zheng et al.'s, 2018)***
		English	***Kurdish***					
No.	***Model***							
1	*LSTM*	*99.00%*	*93.8%*	*94%*		*98%*		
2	*BILSTM*	*99.23%*	*93.6%*				*98%*	
3	*GRU*	-	*93.93%*					
4	*CNN*	*99.03%*	*93.57%*					*78.18%*
5	*Hybrid CNN BILSTM*	*98.97%*	*92.6%*					
6	*RCNN + GRU*			*97%*				
7	*SVM*			*97%*	*93%*			
8	BERT				*98%*			
9	PNN				*92%*			
10	NBC				*92%*			
11	LR				*97%*			

The study's algorithm results in detecting clickbait in English are higher than other established algorithm results such as (Chakraborty et al.'s, 2017) RCNN + GRU and SVM accuracy results of 97% each. Similar inferences can also be drawn when other models such as (Ahmad et al.'s, 2016) BERT (accuracy rate=98%), PNN (accuracy rate=92%), PNN (accuracy rate=92%), NBC (accuracy rate=92%) and LR (accuracy rate=97%) results. The algorithms performed better in detecting clickbait in Kurdish when compared to (Zheng et al.'s, 2018) CNN results that exhibited an accuracy rate of 78.18% as shown in Table 4-16. Further insight into how the current study's results compare with similar work is provided in Table 4-17.

Table 4-12 Comparison of the study's results with similar work

Author	Approach	Feature selection techniques	Models	Used Dataset	Results
(Chakraborty, Paranjape and Kakarla, 2017)	The suggested method combines metric learning and deep learning techniques, coupled with Case-Based Reasoning.	TF-IDF, n-gram, 300 dimensional Word2vec.	CBR + CNN	The author used (Chakraborty, Paranjape and Kakarla, 2016) dataset, which has 32,000 headlines (clickbait and non-clickbait).	Using TF-IDF, Word2vec and n-gram count, the suggested method obtained 0.994, 0.95, and 0.90 average area under the ROC curve.
(Agrawal, 2016)	To detect clickbait, a convolution neural network-based approach is developed.	Click-Word2vec, Click-scratch.	CNN	Created their own corpus from social media platforms.	Click-scratch has an 89% accuracy rate with a 0.87 ROC-AUC score; Click-Word2vec has a 90% accuracy rate with a 0.90 ROC-AUC score.
(Zheng et al.'s, 2018)	Create a browser plugin to identify clickbait headlines automatically.	Sentence Structure, Clickbait Language, Word patterns and n-gram features.	SVM, Decision Tree, Random Forest	Collected 30,000 English headlines (clickbait and non-clickbait) from different websites.	SVM achieved a 93% accuracy rate with 0.95 precision, 0.90 recall, 0.93 F1-score, and 0.97 ROC-AUC values; Decision Tree achieved a 90% accuracy rate with 0.91 precision, 0.89 recall, 0.90 F1-score, and 0.90 ROC-AUC values; and Random Forest achieved a 92% accuracy rate with 0.94 precision, 0.91 recall, 0.92 F1-score, and 0.97 ROC-AUC values using all extracted features.
(Azad et al, 2021)	Used machine learning classification algorithms.	TF-IDF.	Naïve Bayes, SVM, Decision Tree, Random Forest, Logistic Regression	10,000 headlines comprising 5,000 clickbait and 5,000 non-clickbait.	Naïve Bayes-88.58% accuracy rate, with 0.88 precision,0.88 recall,0.88 F1-score.SVM-88.71% accuracy rate, with 0.88 precision, 0.88 recall,0.88 F1-score. Decision Tree-80.44% accuracy rate, with 0.80 precision, 0.80 recall,0.80 F1-s. RandomForest-86.34% accuracy rate, with 0.86

					precision, 0.86 recall,0.86 F1-s. LogisticRegression-87.58% accuracy rate, with 0.87 precision, 0.87 recall,0.87 F1-s
The current study's proposed approach	The proposed approach uses deep learning algorithms.	Keras Tokenizer text to sequence, TF-IDF.	LSTM, BILSTM, CNN, CNN BILSTM	10,000 news headlines were collected in the Kurdish language 5,000 clickbait (fake), and 5,000 non-clickbait(real).	Our proposed system's accuracy is LSTM-93.65% accuracy rate, with 0.96 precision,0.93 recall, and 0.94 F1-score. BILSTM 94.75% accuracy rate, with 0.92 precision, 0.94 recall, 0.93 F1-score. CNN 93.65% accuracy rate, with 0.93 precision,0.93 recall,0.93 F1-score. CNN BILSTM 93.45% accuracy rate, with 0.92 precision,0.94 recall, 0.93 F1-score.

4.4 Summary of Findings

At this stage, the study has successfully answered all the proposed research inquiries. Consequently, it has been reached that understanding the language-specific traits, linguistic patterns, and cultural norms of the target audience is crucial to spotting clickbait across English and Kurdish linguistic and cultural boundaries. The model configuration and implementation involved a comprehensive data preprocessing step, including text cleaning, tokenization, and labeling. Word embeddings were employed, utilizing custom -trained Word embeddings for English and custom-trained embeddings for Kurdish. The BILSTM algorithm for English exhibited a high accuracy rate of 99.23%, with benefits such as language-agnostic architecture and effective handling of long-term dependencies. Challenges included overfitting and hyperparameter tuning. For the BILSTM algorithm, a hidden layer with 100 units and a sigmoid activation function were used.

The GRU algorithm, preferred for Kurdish, achieved a 93.93% accuracy rate, showcasing benefits like language-agnostic architecture and computational efficiency. Challenges included obtaining sufficient labeled data and addressing interpretability issues, lack of parallelization, vanishing/exploding gradients, and limited long-term dependency modeling problems. For the GRU algorithm, a hidden layer with 100 units and a sigmoid activation function were utilized. The study provides valuable insights into the intricacies of clickbait detection across diverse languages and highlights the importance of tailoring models to language-specific characteristics.

Chapter Five

CONCLUSION AND FUTURE WORKS

5. CONCLUSION AND FUTURE WORD

5.1 Conclusion

The study's main emphasis was to evaluate the performance Clickbait Detection Through Deep Learning and Language-specific Analysis in English and Kurdish. we have investigated the detection of clickbait headlines in both English and Kurdish languages using various deep learning algorithms, including LSTM, BILSTM, CNN, GRU and CNN BILSTM. The primary objective of this research was to develop effective and accurate clickbait detection models capable of handling the nuances and linguistic differences in both languages.

Throughout the study, we collected and curated a substantial dataset of clickbait and non-clickbait headlines from diverse sources in English and Kurdish languages. This dataset served as the foundation for training, validating, and testing our models. We preprocessed the data by tokenizing, converting to lowercase, and removing stop words and special characters to improve the performance of the algorithms.

Other notable conclusions drawn from this chapter are highlighted as follows:

- The BILSTM algorithm ranked the best in detecting clickbait in English (accuracy=99.23%, precision=95.33%, recall=94.33% and F1 score=95%). The CNN algorithm was ranked second (accuracy=99.03%, precision=94.33%, recall=96% and F1 score=94.33%). Third was the LSTM algorithm (accuracy=99%, precision=94.33%, recall=95.33% and F1 score=95%) and last was the hybrid CNN BILSTM (accuracy=98.96%, precision=94%, recall=95.67% and F1 score=94.67%).

- By selecting the BILSTM algorithm in detecting clickbait in English benefits such as language-agnostic architecture, generalization ability, language modelling, handling long-term dependencies and capturing contextual information are conceivable. However, in dealing with its overfitting and hyperparameter tuning, the need for sufficient labelled data, difficulty in interpretability, limited parallelization and computational complexity problems is instrumental if better performance results are to be achieved.
- Concerning the detection of clickbait in Kurdish, the GRU algorithm performs better than other algorithms. It has an accuracy rate of 93.93%, 93.13% precision, 95.17% recall rate and 94.13% F1 score. LSTM ranks second (accuracy=93.8%, precision=93.5%, recall=94.13% and F1 score=93.77%). The BILSTM ranked third with an accuracy of 99.23%, 95.33% precision, 94.33% recall rate and an F1 score of 95%. CNN fourth with an accuracy of 93.57%, 92.77% precision, 93.97% recall rate and an F1 score of 93.37%. Lastly, the hybrid CNN BILSTM ranked fifth with an accuracy of 92.3%, 90.87% precision, 95.03% recall rate and an F1 score of 92.97%.
- By selecting the GRU algorithm in detecting clickbait in Kurdish, benefits such as language-agnostic architecture, handling variable-length inputs, generalization ability, computational efficiency, and capturing sequential information are conceivable. However, dealing with its need for sufficient labelled data, difficulty in interpretability, lack of parallelization, vanishing/exploding gradients, and limited long-term dependency modelling problems is instrumental if better performance results are to be achieved.
- Concerning clickbait detection performance metrics, accuracy, precision, recall and F1 scores vary significantly between each algorithm's single, double and triple layers.

The study contributes to the field of natural language processing by investigating the detection of clickbait in Kurdish and English. This study addresses the need for effective methods for detecting clickbait content automatically, to facilitate user experiences and prevent the spread of false information.

5.2 Future Works

The present study focused on detecting clickbait headlines using deep learning algorithms, specifically exploring LSTM, BILSTM, GRU, and CNN, CNN-BILSTM architectures. While the findings provide valuable insights into the performance of these models in the context of Kurdish and English languages, several areas warrant further investigation to enhance the scope and applicability of clickbait detection approaches. In this section, we outline potential future directions and feature work for advancing clickbait detection research:

1- **Applying GAN or TGAN for Text Generation**: Consider applying Generative Adversarial Networks (GAN) or Text GAN for text generation. Train a GAN model on the clickbait headlines to generate synthetic clickbait-like headlines. Combining real and synthetic clickbait headlines during training may improve the model's ability to identify subtle patterns and distinguish clickbait from legitimate content effectively.

2- **Applying Transfer Learning to Clickbait Detection**: Utilize transfer learning from large-scale language models like GPT-3 or BERT pretrained on vast corpora. Fine-tune these models on the clickbait detection task with a small amount of labeled data. Transfer learning from such models can significantly boost performance, especially when dealing with limited labeled data.

3- **Addressing Sample Bias and Generalizability**: As noted during the study, the dataset's limited representation of clickbait headlines in Kurdish and English may introduce sample bias, affecting the generalizability of the models' performance. Future research should focus on collecting a more diverse and balanced dataset, encompassing clickbait headlines from multiple sources, languages, genres, and domains. Techniques such as data augmentation and cross-lingual transfer learning can contribute to a richer dataset and improve the models' ability to detect clickbait across different linguistic contexts.

4- **Explainable AI and Human Evaluation**: Considering the importance of understanding model decisions, incorporating explainable AI techniques will provide valuable insights into the linguistic patterns and features contributing to clickbait classification. Moreover, conducting user studies and obtaining feedback from human users will allow for a comprehensive evaluation of the models' alignment with human judgments and user expectations.

5- **Real-world Deployment Considerations**: To ensure the practical applicability of clickbait detection approaches, it is essential to address challenges related to real-world deployment. Evaluating the computational efficiency and scalability of the models will help identify feasible solutions for integrating clickbait detection into online platforms and content management systems.

REFERENCES

REFERENCES

AHMAD, I., ALQARNI, M.A., ALMAZROI, A.A. AND TARIQ, A., 2020. Experimental Evaluation of Clickbait Detection Using Machine Learning Models. *Intelligent Automation & Soft Computing*, *26*(6).

AL-SAREM, M., SAEED, F., AL-MEKHLAFI, Z.G., MOHAMMED, B.A., HADWAN, M., AL-HADHRAMI, T., ALSHAMMARI, M.T., ALRESHIDI, A. AND ALSHAMMARI, T.S., 2021. An Improved Multiple Features and Machine Learning-Based Approach for Detecting Clickbait News on Social Networks. *Applied Sciences*, *11*(20), p.9487.

ALTHENEYAN, A. AND ALHADLAQ, A., 2023. Big Data ML-Based Fake News Detection using Distributed Learning. *IEEE Access*, *11*, pp.29447-29463.

ATILA, Ü. AND SABAZ, F., 2022. Turkish lip-reading using BILSTM and deep learning models. *Engineering Science and Technology, an International Journal*, *35*, p.101206.

AVAZOV, K., ABDUSALOMOV, A., MUKHIDDINOV, M., BARATOV, N., MAKHMUDOV, F. AND CHO, Y.I., 2022. An improvement for the automatic classification method for ultrasound images used on CNN. *International Journal of Wavelets, Multiresolution and Information Processing*, *20*(02), p.2150054.

BARWICK, R. 2022. *Advertisers spent $115 million on clickbait sites, report finds*, viewed 20 November 2022 <https://www.marketingbrew.com/stories/2022/07/18/advertisers-spent-usd115-million-on-clickbait-sites-report-finds>.

BIANCHI, F.M., LIVI, L., MIKALSEN, K.Ø., KAMPFFMEYER, M. AND JENSSEN, R., 2019. Learning representations of multivariate time series with missing data. *Pattern Recognition*, *96*, p.106973.

BIEBERICH, E., 2002. Recurrent fractal neural networks: a strategy for the exchange of local and global information processing in the brain. *Biosystems*, *66*(3), pp.145-164.

BIYANI, P., TSIOUTSIOULIKLIS, K. AND BLACKMER, J., 2016, February. " 8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

BRONAKOWSKI, M., AL-KHASSAWENEH, M. AND AL BATAINEH, A., 2023. Automatic Detection of Clickbait Headlines Using Semantic Analysis and Machine Learning Techniques. *Applied Sciences*, *13*(4), p.2456.

BULLOCK, J., CUESTA-LÁZARO, C. AND QUERA-BOFARULL, A., 2019, March. XNet: a convolutional neural network (CNN) implementation for medical x-ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging* (Vol. 10953, pp. 453-463). SPIE.

CHAKRABORTY, A., PARANJAPE, B., KAKARLA, S. AND GANGULY, N., 2016, August. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 9-16). IEEE.

CHEN, J., ZHANG, G., XU, S. AND YU, H., 2019. A blind CNN denoising model for random-valued impulse noise. *IEEE Access*, *7*, pp.124647-124661.

CHEN, Y., CONROY, N.J. AND RUBIN, V.L., 2015, November. Misleading online content: recognizing clickbait as" false news". In *Proceedings of the 2015 ACM on workshop on multimodal deception detection* (pp. 15-19).

CHOOBI, M.S., HAGHPANAHI, M. AND SEDIGHI, M., 2012. Prediction of welding-induced angular distortions in thin butt-welded plates using artificial neural networks. *Computational Materials Science*, *62*, pp.152-159.

CODDINGTON, M., 2019. Chapter Four. Clickbait, Analytics, And Gut Feelings: How Aggregators Understand Their Audiences. In *Aggregating the News* (pp. 129-157). Columbia University Press.

DEAN, B. 2022. *Facebook Demographic Statistics: How Many People Use Facebook in 2022?,* viewed 20 November 2022, <https://backlinko.com/facebook-users>.

References

DEPERLIOGLU, O., KOSE, U., GUPTA, D., KHANNA, A., GIAMPAOLO, F. AND FORTINO, G., 2022. Explainable framework for Glaucoma diagnosis by image processing and convolutional neural network synergy: analysis with doctor evaluation. *Future Generation Computer Systems*, *129*, pp.152-169.

DIMPAS, P.K., PO, R.V. AND SABELLANO, M.J., 2017, December. Filipino and english clickbait detection using a long short term memory recurrent neural network. In *2017 International Conference on Asian Language Processing (IALP)* (pp. 276-280). IEEE.

DOMINGO, M., GARCÍA-MARTÍNEZ, M., HELLE, A., CASACUBERTA, F. AND HERRANZ, M., 2023, February. How much does tokenization affect neural machine translation?. In *Computational Linguistics and Intelligent Text Processing: 20th International Conference, CICLing 2019, La Rochelle, France, April 7–13, 2019, Revised Selected Papers, Part I* (pp. 545-554). Cham: Springer Nature Switzerland.

ELYASHAR, A., BENDAHAN, J. AND PUZIS, R., 2022, June. Detecting clickbait in online social media: You won't believe how we did it. In *Cyber Security, Cryptology, and Machine Learning: 6th International Symposium, CSCML 2022, Be'er Sheva, Israel, June 30–July 1, 2022, Proceedings* (pp. 377-387). Cham: Springer International Publishing.

ELYASHAR, A., BENDAHAN, J. AND PUZIS, R., 2022. Detecting clickbait in online social media: You won't believe how we did it. In *International Symposium on Cyber Security, Cryptology, and Machine Learning* (pp. 377-387). Springer, Cham.

FAKHRUZZAMAN, M.N., SA'IDAH ZAHROTUL JANNAH, R.A. AND NINGRUM, I.F., 2023. Flagging clickbait in Indonesian online news websites using fine-tuned transformers. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(3), pp.2921-2930.

FENG, H., FU, X., SUN, H., WANG, H. AND ZHANG, Y., 2020, July. Efficient Vulnerability Detection based on abstract syntax tree and Deep Learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 722-727). IEEE.

References

FU, J., LIANG, L., ZHOU, X. AND ZHENG, J., 2017, July. A convolutional neural network for clickbait detection. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 6-10). IEEE.

GAMAGE, B., LABIB, A., JOOMUN, A., LIM, C.H. AND WONG, K., 2021, June. Baitradar: A multi-model clickbait detection algorithm using deep learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2665-2669). IEEE.

GAN, W., SUN, Y. AND SUN, Y., 2022. Knowledge structure enhanced graph representation learning model for attentive knowledge tracing. *International Journal of Intelligent Systems*, *37*(3), pp.2012-2045.

GEÇKIL, A., MÜNGEN, A.A., GÜNDOGAN, E. AND KAYA, M., 2018, August. A clickbait detection method on news sites. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 932-937). IEEE.

GENÇ, Ş., 2021. *Turkish clickbait detection in social media via machine learning algorithms* (Master's thesis, Middle East Technical University).

GHOSH, S., 2021. Is it a click bait? Let's predict using Machine Learning. *arXiv preprint arXiv:2106.07348*.

GROSSBERG, S., 2013. Recurrent neural networks. *Scholarpedia*, *8*(2), p.1888.

HASSOUN, A., BEACOCK, I., CONSOLVO, S., GOLDBERG, B., KELLEY, P.G. AND RUSSELL, D.M., 2023, April. Practicing Information Sensibility: How Gen Z Engages with Online Information. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1-17).

INDURTHI, V. AND OOTA, S.R., 2017. Clickbait detection using word embeddings. *arXiv preprint arXiv:1710.02861*.

INDURTHI, V., SYED, B., GUPTA, M. AND VARMA, V., 2020, December. Predicting clickbait strength in online social media. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 4835-4846).

References

JAIN, M., MOWAR, P., GOEL, R. AND VISHWAKARMA, D.K., 2021, March. Clickbait in Social Media: Detection and Analysis of the Bait. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)* (pp. 1-6). IEEE.

JANG, B., KIM, M., HARERIMANA, G., KANG, S.U. AND KIM, J.W., 2020. BILSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism. *Applied Sciences*, *10*(17), p.5841.

KHAN, J.Y., KHONDAKER, M.T.I., AFROZ, S., UDDIN, G. AND IQBAL, A., 2021. A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, *4*, p.100032.

KHATER, S.R., AL-SAHLEE, O. H., DAOUD, D. M. AND EL-SEOUD, M.S.A., 2018. Clickbait detection. In *Proceedings of the 7th international conference on software and information engineering* (pp. 111-115).

KLAIRITH, P. AND TANACHUTIWAT, S., 2018, July. Thai clickbait detection algorithms using natural language processing with machine learning techniques. In *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)* (pp. 1-4). IEEE.

KUMAR, V., KHATTAR, D., GAIROLA, S., KUMAR LAL, Y. AND VARMA, V., 2018, June. Identifying clickbait: A multi-strategy approach using neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 1225-1228).

LI, X., ZHOU, J., XIANG, H. AND CAO, J., 2022. Attention Grabbing through Forward Reference: An ERP Study on Clickbait and Top News Stories. *International Journal of Human–Computer Interaction*, pp.1-16.

LOEWENSTEIN, G., 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, *116*(1), p.75.

LONG, M., WANG, J., CAO, Y., SUN, J. AND PHILIP, S.Y., 2016. Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, *28*(8), pp.2027-2040.

References

MARTINS, G.B., PAPA, J.P. AND ADELI, H., 2020. Deep learning techniques for recommender systems based on collaborative filtering. *Expert Systems*, *37*(6), p.e12647.

MEKRUKSAVANICH, S. AND JITPATTANAKUL, A., 2021. Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data. *Electronics*, *10*(14), p.1685.

MOON, S., LEE, G., CHI, S. AND OH, H., 2021. Automated construction specification review with named entity recognition using natural language processing. *Journal of Construction Engineering and Management*, *147*(1), p.04020147.

MOSCADELLI, A., ALBORA, G., BIAMONTE, M.A., GIORGETTI, D., INNOCENZIO, M., PAOLI, S., LORINI, C., BONANNI, P. AND BONACCORSI, G., 2020. Fake news and Covid-19 in Italy: results of a quantitative observational study. *International journal of environmental research and public health*, *17*(16), p.5850.

MOU, L., GHAMISI, P. AND ZHU, X.X., 2017. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), pp.3639-3655.

MUKHERJEE, P., DUTTA, S. AND DE BRUYN, A., 2022. Did clickbait crack the code on virality?. *Journal of the Academy of Marketing Science*, *50*(3), 482-502.

NADIA, B.U. AND ISWANTO, I.A., 2021. Indonesian Clickbait Detection Using Improved Backpropagation Neural Network. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 252-256). IEEE.

NAEEM, B., KHAN, A., BEG, M.O. AND MUJTABA, H., 2020. A deep learning framework for clickbait detection on social area network using natural language cues. *Journal of Computational Social Science*, *3*(1), pp.231-243.

NGAN, N.N.T. 2020. *Clickbait anatomy: Identifying clickbait with machine learning* (Doctoral dissertation, Vrije Universiteit Amsterdam).

References

PARANJAPE, B., 2015. *Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media Study of Click-Bait Articles* (Doctoral dissertation, Indian Institute of Technology, Kharagpur).

POTTHAST, M., KÖPSEL, S., STEIN, B. AND HAGEN, M., 2016, March. Clickbait detection. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38* (pp. 810-817). Springer International Publishing.

PRENTICE, S., 2022. The Fear of Missing Out. In *The Future of Workplace Fear: How Human Reflex Stands in the Way of Digital Transformation* (pp. 183-198). Berkeley, CA: Apress.

PUJAHARI, A. AND SISODIA, D.S., 2021. Clickbait detection using multiple categorisation techniques. *Journal of Information Science*, *47*(1), pp.118-128.

QIN, Y., SONG, D., CHEN, H., CHENG, W., JIANG, G. AND COTTRELL, G., 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*.

RAZAQUE, A., ALOTAIBI, B., ALOTAIBI, M., HUSSAIN, S., ALOTAIBI, A. AND JOTSOV, V., 2022. Clickbait detection using deep recurrent neural network. *Applied Sciences*, *12*(1), p.504.

ROLNIK, G., CAGÉ, J., GANS, J., GOODMAN, E., KNIGHT, B., PRAT, A. AND SCHIFFRIN, A., 2019. *Protecting journalism in the age of digital platforms* (Doctoral dissertation, George J. Stigler Center for the Study of the Economy and the State; The University of Chicago Booth School of Business).

SANTOS, A.A., ROCHA, F.A., REIS, A.J.D.R. AND GUIMARÃES, F.G., 2020. Automatic system for visual detection of dirt buildup on conveyor belts using convolutional neural networks. *Sensors*, *20*(20), p.5762.

SETLUR, A.R., 2018. Semi-supervised confidence network aided gated attention based recurrent neural network for clickbait detection. *arXiv preprint arXiv:1811.01355*.

References

SHANG, L., ZHANG, D.Y., WANG, M., LAI, S. AND WANG, D., 2019. Towards reliable online clickbait video detection: A content-agnostic approach. *Knowledge-Based Systems*, *182*, p.104851.

SHENG, Q., ZHANG, X., CAO, J. AND ZHONG, L., 2021, October. Integrating pattern-and fact-based fake news detection via model preference learning. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1640-1650).

SHEZAN, M.R.H., ZAWAD, M.N., SHAHED, Y.A., AND RIPON, S., 2023. Bangla Fake News Detection Using Hybrid Deep Learning Models. In *Applied Informatics for Industry 4.0* (pp. 46-60). Chapman and Hall/CRC.

SHI, Q., KATUWAL, R., SUGANTHAN, P.N. AND TANVEER, M., 2021. Random vector functional link neural network based ensemble deep learning. *Pattern Recognition*, *117*, p.107978.

SHU, K., SLIVA, A., WANG, S., TANG, J. AND LIU, H., 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, *19*(1), pp.22-36.

SIREGAR, B., HABIBIE, I. AND NABABAN, E.B., 2021, May. Identification of Indonesian clickbait news headlines with long short-term memory recurrent neural network algorithm. In *Journal of Physics: Conference Series* (Vol. 1882, No. 1, p. 012129). IOP Publishing.

SULTANA, F., SUFIAN, A. AND DUTTA, P., 2020. A review of object detection models based on convolutional neural network. *Intelligent computing: image processing-based applications*, pp.1-16.

TANG, Y., HUANG, Y., WU, Z., MENG, H., XU, M. AND CAI, L., 2016, March. Question detection from acoustic features using recurrent neural network with gated recurrent unit. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6125-6129). IEEE.

THIEL, K., 2018. *Avoiding Clickbait*. Cavendish Square Publishing, LLC.

References

THOMAS, P., 2017. Clickbait identification using neural networks. *arXiv preprint arXiv:1710.08721*.

TIWARI, V., PANDEY, C., DWIVEDI, A. AND YADAV, V., 2020, December. Image classification using deep neural network. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (pp. 730-733). IEEE.

WANG, L., TANG, D., LIU, C., NIE, Q., WANG, Z. AND ZHANG, L., 2022. An Augmented Reality-Assisted Prognostics and Health Management System Based on Deep Learning for IoT-Enabled Manufacturing. *Sensors*, *22*(17), p.6472.

WANG, X., XU, J., SHI, W. AND LIU, J., 2019, October. OGRU: An optimized gated recurrent unit neural network. In *Journal of Physics: Conference Series* (Vol. 1325, No. 1, p. 012089). IOP Publishing.

VAN DER LINDEN, S., ROOZENBEEK, J. AND COMPTON, J., 2020. Inoculating against fake news about COVID-19. *Frontiers in psychology*, *11*, p.566790.

VARSHNEY, D. AND VISHWAKARMA, D.K., 2021. A unified approach for detection of Clickbait videos on YouTube using cognitive evidences. *Applied Intelligence*, *51*(7), pp.4214-4235.

VORAKITPHAN, V., LEU, F.Y. AND FAN, Y.C., 2019. Clickbait detection based on word embedding models. In *Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 12th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018)* (pp. 557-564). Springer International Publishing.

WONGSAP, N. et al. (2018) ‘Thai clickbait headline news classification and its characteristic’, 2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES) [Preprint]. doi:10.1109/icesit-icictes.2018.8442064.

YADAV, S.S. AND JADHAV, S.M., 2019. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big data*, *6*(1), 1-18.

YUAN, X., QI, S., WANG, Y. AND XIA, H., 2020. A dynamic CNN for nonlinear dynamic feature learning in soft sensor modeling of industrial process data. *Control Engineering Practice*, *104*, 104614.

ZAREMBA, W., Sutskever, I., Vinyals, O., 2015. Recurrent Neural Network Regularization.

ZANNETTOU, S., CHATZIS, S., PAPADAMOU, K. AND SIRIVIANOS, M., 2018, May. The good, the bad and the bait: Detecting and characterizing clickbait on youtube. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 63-69). IEEE.

ZANNETTOU, S., SIRIVIANOS, M., BLACKBURN, J. AND KOURTELLIS, N., 2019. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. *Journal of Data and Information Quality (JDIQ)*, *11*(3), pp.1-37.

ZHENG, H.T., CHEN, J.Y., YAO, X., SANGAIAH, A.K., JIANG, Y. AND ZHAO, C.Z., 2018. Clickbait convolutional neural network. *Symmetry*, *10*(5), p.138.

ZHENG, H.T., YAO, X., JIANG, Y., XIA, S.T. AND XIAO, X., 2017, July. Boost clickbait detection based on user behavior analysis. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data* (pp. 73-80). Springer, Cham.

ZHENG, J., XU, C., ZHANG, Z. AND LI, X., 2017, March. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In *2017 51st Annual conference on information sciences and systems (CISS)* (pp. 1-6). IEEE.

ZHOU, M., XU, W., ZHANG, W. AND JIANG, Q., 2022. Leverage knowledge graph and GCN for fine-grained-level clickbait detection. *World Wide Web*, *25*(3), pp.1243-1258.

ZHOU, X., JAIN, A., PHOHA, V.V. AND ZAFARANI, R., 2020. Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, *1*(2), pp.1-25.

ZHOU, Y., 2017. Clickbait detection in tweets using self-attentive network. *arXiv preprint arXiv:1710.05*

1 Layer LSTM

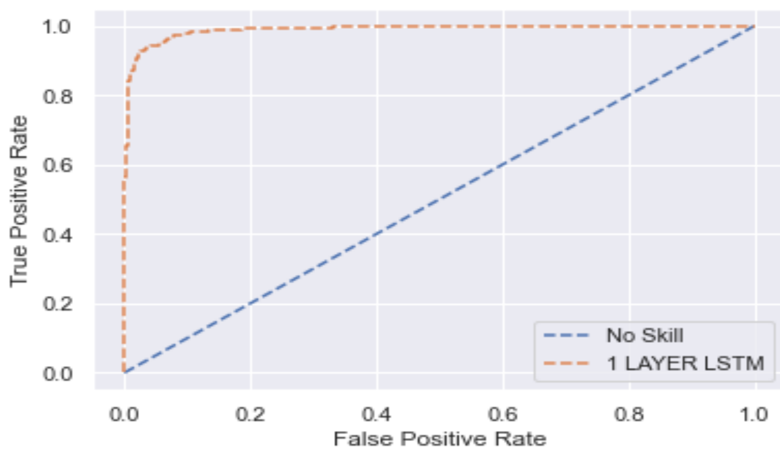
```
n_lstm=100
embeding_dim =128
drop_lstm =0.2
model1 = Sequential()
model1.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model1.add(LSTM(100, dropout=drop_lstm))
model1.add(Dense(1, activation='sigmoid'))
model1.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 13, 128)           2837632

 lstm (LSTM)                 (None, 100)               91600

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 2,929,333
Trainable params: 2,929,333
Non-trainable params: 0
_________________________________________________________________
```

```
model1.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model1.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 28s - loss: 0.2543 - accuracy: 0.8884 - val_loss: 0.1295 - val_accuracy: 0.9515 - 28s/epoch - 28ms/step
Epoch 2/5
1000/1000 - 27s - loss: 0.0389 - accuracy: 0.9871 - val_loss: 0.1356 - val_accuracy: 0.9485 - 27s/epoch - 27ms/step
Epoch 3/5
1000/1000 - 27s - loss: 0.0102 - accuracy: 0.9976 - val_loss: 0.1721 - val_accuracy: 0.9510 - 27s/epoch - 27ms/step
Epoch 4/5
1000/1000 - 27s - loss: 0.0083 - accuracy: 0.9981 - val_loss: 0.1979 - val_accuracy: 0.9440 - 27s/epoch - 27ms/step
Epoch 5/5
1000/1000 - 27s - loss: 0.0029 - accuracy: 0.9996 - val_loss: 0.2146 - val_accuracy: 0.9480 - 27s/epoch - 27ms/step
```

```
auc score 0.9896984952262662
```




```
yhat_classes1 = (model1.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes1))
```

```
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      1007
           1       0.95      0.94      0.95       993

    accuracy                           0.95      2000
   macro avg       0.95      0.95      0.95      2000
weighted avg       0.95      0.95      0.95      2000
```

Appendix

2 Layer Lstm

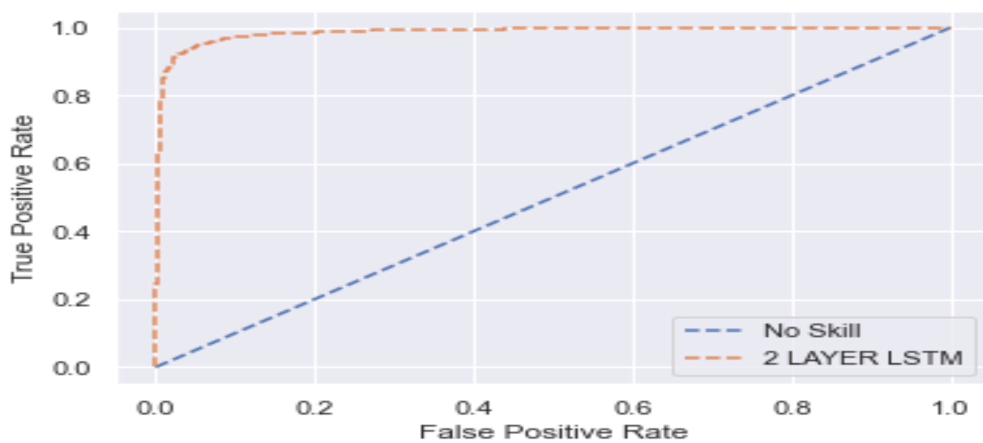
```
n_lstm=100
embeding_dim =128
drop_lstm =0.2
model2 = Sequential()
model2.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model2.add(LSTM(100, dropout=drop_lstm,return_sequences=True))
model2.add(LSTM(100, dropout=drop_lstm))
model2.add(Dense(1, activation='sigmoid'))
model2.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 13, 128)           2837632

 lstm_1 (LSTM)               (None, 13, 100)           91600

 lstm_2 (LSTM)               (None, 100)               80400

 dense_1 (Dense)             (None, 1)                 101

=================================================================
Total params: 3,009,733
Trainable params: 3,009,733
Non-trainable params: 0
_________________________________________________________________
```

```
model2.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model2.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 32s - loss: 0.2541 - accuracy: 0.8909 - val_loss: 0.1475 - val_accuracy: 0.9465 - 32s/epoch - 32ms/step
Epoch 2/5
1000/1000 - 29s - loss: 0.0381 - accuracy: 0.9865 - val_loss: 0.1356 - val_accuracy: 0.9470 - 29s/epoch - 29ms/step
Epoch 3/5
1000/1000 - 29s - loss: 0.0100 - accuracy: 0.9971 - val_loss: 0.1622 - val_accuracy: 0.9405 - 29s/epoch - 29ms/step
Epoch 4/5
1000/1000 - 28s - loss: 0.0051 - accuracy: 0.9990 - val_loss: 0.2107 - val_accuracy: 0.9465 - 28s/epoch - 28ms/step
Epoch 5/5
1000/1000 - 29s - loss: 0.0068 - accuracy: 0.9985 - val_loss: 0.2204 - val_accuracy: 0.9460 - 29s/epoch - 29ms/step
```

```
auc score 0.9861823229338238
```




```
yhat_classes2 = (model2.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes2))
```

```
              precision    recall  f1-score   support

           0       0.93      0.96      0.95      1007
           1       0.96      0.93      0.94       993

    accuracy                           0.95      2000
   macro avg       0.95      0.95      0.95      2000
weighted avg       0.95      0.95      0.95      2000
```

Appendix

3 Layer Lstm

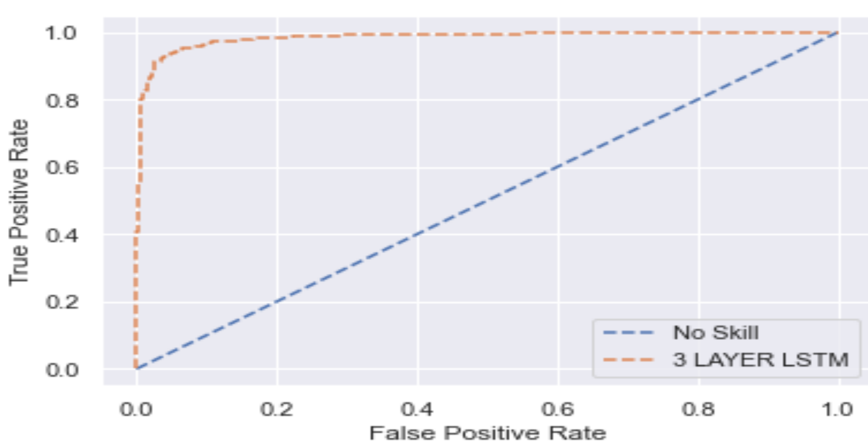
```
n_lstm=100
embeding_dim =128
drop_lstm =0.2
model3 = Sequential()
model3.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model3.add(LSTM(100, dropout=drop_lstm,return_sequences=True))
model3.add(LSTM(100, dropout=drop_lstm,return_sequences=True))
model3.add(LSTM(100, dropout=drop_lstm))
model3.add(Dense(1, activation='sigmoid'))
model3.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 13, 128)           2837632

 lstm_3 (LSTM)               (None, 13, 100)           91600

 lstm_4 (LSTM)               (None, 13, 100)           80400

 lstm_5 (LSTM)               (None, 100)               80400

 dense_2 (Dense)             (None, 1)                 101

=================================================================
Total params: 3,090,133
Trainable params: 3,090,133
Non-trainable params: 0
_________________________________________________________________
```

```
model3.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model3.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 33s - loss: 0.2564 - accuracy: 0.8879 - val_loss: 0.1387 - val_accuracy: 0.9495 - 33s/epoch - 33ms/step
Epoch 2/5
1000/1000 - 30s - loss: 0.0387 - accuracy: 0.9865 - val_loss: 0.1519 - val_accuracy: 0.9395 - 30s/epoch - 30ms/step
Epoch 3/5
1000/1000 - 30s - loss: 0.0098 - accuracy: 0.9979 - val_loss: 0.2620 - val_accuracy: 0.9445 - 30s/epoch - 30ms/step
Epoch 4/5
1000/1000 - 30s - loss: 0.0081 - accuracy: 0.9983 - val_loss: 0.2376 - val_accuracy: 0.9345 - 30s/epoch - 30ms/step
Epoch 5/5
1000/1000 - 32s - loss: 0.0061 - accuracy: 0.9984 - val_loss: 0.2544 - val_accuracy: 0.9425 - 32s/epoch - 32ms/step
```

```
auc score 0.9842372276241536
```




```
yhat_classes3 = (model3.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes3))
```

```
              precision    recall  f1-score   support

           0       0.95      0.94      0.94      1007
           1       0.94      0.94      0.94       993

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000
```

Appendix

LSTM

1 Layer Bi-LSTM ¶

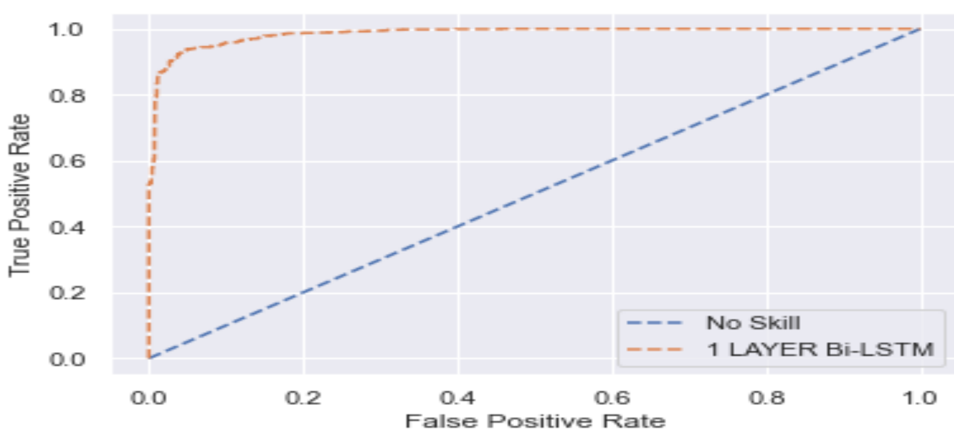
```
4]: n_lstm=100
embeding_dim =128
drop_lstm =0.2
model1 = Sequential()
model1.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model1.add(Bidirectional(LSTM(100, dropout=drop_lstm)))
model1.add(Dense(1, activation='sigmoid'))
model1.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 13, 128)           2837632

 bidirectional (Bidirectiona  (None, 200)              183200
 l)

 dense (Dense)               (None, 1)                 201

=================================================================
Total params: 3,021,033
Trainable params: 3,021,033
Non-trainable params: 0
_________________________________________________________________
```

```
5]: model1.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model1.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 33s - loss: 0.2430 - accuracy: 0.8899 - val_loss: 0.1545 - val_accuracy: 0.9415 - 33s/epoch - 33ms/step
Epoch 2/5
1000/1000 - 30s - loss: 0.0345 - accuracy: 0.9894 - val_loss: 0.1728 - val_accuracy: 0.9395 - 30s/epoch - 30ms/step
Epoch 3/5
1000/1000 - 30s - loss: 0.0093 - accuracy: 0.9974 - val_loss: 0.2420 - val_accuracy: 0.9225 - 30s/epoch - 30ms/step
Epoch 4/5
1000/1000 - 30s - loss: 0.0112 - accuracy: 0.9967 - val_loss: 0.2125 - val_accuracy: 0.9385 - 30s/epoch - 30ms/step
Epoch 5/5
1000/1000 - 30s - loss: 0.0082 - accuracy: 0.9979 - val_loss: 0.2550 - val_accuracy: 0.9440 - 30s/epoch - 30ms/step
```

```
auc score 0.9855932050727771
```




```
yhat_classes1 = (model1.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes1))
```

```
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       972
           1       0.95      0.94      0.95      1028

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000
```

Appendix

2 Layer Bi - Lstm

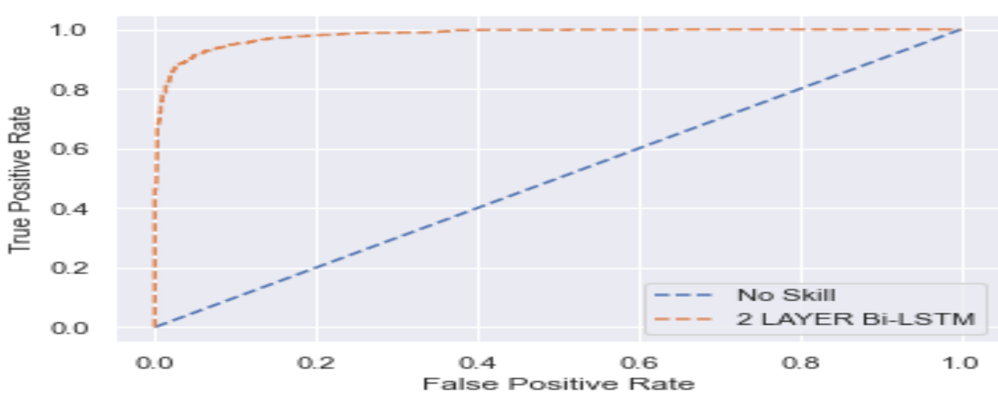
```
n_lstm=100
embeding_dim =128
drop_lstm =0.2
model2 = Sequential()
model2.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model2.add(Bidirectional(LSTM(100, dropout=drop_lstm,return_sequences=True)))
model2.add(Bidirectional(LSTM(100, dropout=drop_lstm)))
model2.add(Dense(1, activation='sigmoid'))
model2.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 13, 128)           2837632

 bidirectional_1 (Bidirectio  (None, 13, 200)          183200
 nal)

 bidirectional_2 (Bidirectio  (None, 200)              240800
 nal)

 dense_1 (Dense)             (None, 1)                 201

=================================================================
Total params: 3,261,833
Trainable params: 3,261,833
Non-trainable params: 0
_________________________________________________________________
```

```
model2.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model2.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 37s - loss: 0.2354 - accuracy: 0.9000 - val_loss: 0.1467 - val_accuracy: 0.9445 - 37s/epoch - 37ms/step
Epoch 2/5
1000/1000 - 33s - loss: 0.0320 - accuracy: 0.9893 - val_loss: 0.2115 - val_accuracy: 0.9315 - 33s/epoch - 33ms/step
Epoch 3/5
1000/1000 - 33s - loss: 0.0134 - accuracy: 0.9960 - val_loss: 0.1742 - val_accuracy: 0.9290 - 33s/epoch - 33ms/step
Epoch 4/5
1000/1000 - 33s - loss: 0.0085 - accuracy: 0.9977 - val_loss: 0.2747 - val_accuracy: 0.9300 - 33s/epoch - 33ms/step
Epoch 5/5
1000/1000 - 33s - loss: 0.0098 - accuracy: 0.9971 - val_loss: 0.2786 - val_accuracy: 0.9295 - 33s/epoch - 33ms/step
```

```
auc score 0.982245080142832
```




```
yhat_classes2 = (model2.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes2))
```

```
              precision    recall  f1-score   support

           0       0.93      0.92      0.93       972
           1       0.93      0.93      0.93      1028

    accuracy                           0.93      2000
   macro avg       0.93      0.93      0.93      2000
weighted avg       0.93      0.93      0.93      2000
```

Appendix

3 Layer Bi- Lstm

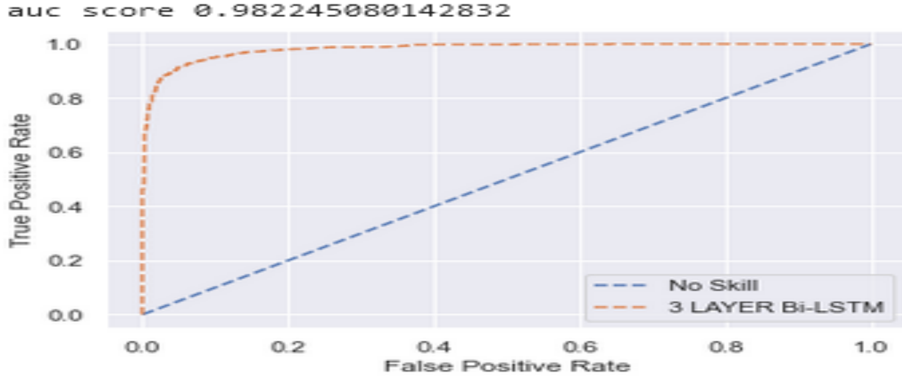
```
n_lstm=100
embeding_dim =128
drop_lstm =0.2
model3 = Sequential()
model3.add(Embedding(vocab_size, embeding_dim, input_length=max_len))
model3.add(Bidirectional(LSTM(100, dropout=drop_lstm,return_sequences=True)))
model3.add(Bidirectional(LSTM(100, dropout=drop_lstm,return_sequences=True)))
model3.add(Bidirectional(LSTM(100, dropout=drop_lstm)))
model3.add(Dense(1, activation='sigmoid'))
model3.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 13, 128)           2837632

 bidirectional_3 (Bidirectio  (None, 13, 200)          183200
 nal)

 bidirectional_4 (Bidirectio  (None, 13, 200)          240800
 nal)

 bidirectional_5 (Bidirectio  (None, 200)              240800
 nal)

 dense_2 (Dense)             (None, 1)                 201

=================================================================
Total params: 3,502,633
Trainable params: 3,502,633
Non-trainable params: 0
_________________________________________________________________
```

```
model3.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model3.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 43s - loss: 0.2395 - accuracy: 0.8988 - val_loss: 0.1648 - val_accuracy: 0.9435 - 43s/epoch - 43ms/step
Epoch 2/5
1000/1000 - 36s - loss: 0.0341 - accuracy: 0.9893 - val_loss: 0.1707 - val_accuracy: 0.9370 - 36s/epoch - 36ms/step
Epoch 3/5
1000/1000 - 36s - loss: 0.0137 - accuracy: 0.9971 - val_loss: 0.3613 - val_accuracy: 0.9310 - 36s/epoch - 36ms/step
Epoch 4/5
1000/1000 - 36s - loss: 0.0110 - accuracy: 0.9977 - val_loss: 0.3512 - val_accuracy: 0.9470 - 36s/epoch - 36ms/step
Epoch 5/5
1000/1000 - 36s - loss: 0.0106 - accuracy: 0.9974 - val_loss: 0.3353 - val_accuracy: 0.9390 - 36s/epoch - 36ms/step
```

```
auc score 0.982245080142832
```




```
yhat_classes3 = (model3.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes3))
```

```
              precision    recall  f1-score   support

           0       0.92      0.95      0.94       972
           1       0.96      0.92      0.94      1028

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000
```

Appendix

1 CNN LAYER

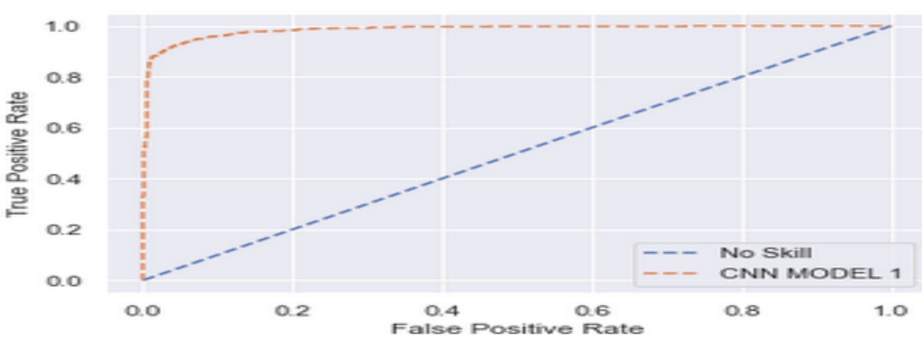
```
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D,Flatten,Dense
model1 = Sequential()
model1.add(Embedding(vocab_size, 100, input_length=max_len))
model1.add(Conv1D(filters=32, kernel_size=8, activation='relu'))
model1.add(MaxPooling1D(pool_size=2))
model1.add(Flatten())
model1.add(Dense(10, activation='relu'))
model1.add(Dense(1, activation='sigmoid'))
print(model1.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 13, 100)           2216900

 conv1d (Conv1D)             (None, 6, 32)             25632

 max_pooling1d (MaxPooling1D  (None, 3, 32)            0
 )

 flatten (Flatten)           (None, 96)                0

 dense (Dense)               (None, 10)                970

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,243,513
Trainable params: 2,243,513
Non-trainable params: 0
_________________________________________________________________
None
```

```
model1.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model1.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 22s - loss: 0.2723 - accuracy: 0.8819 - val_loss: 0.1571 - val_accuracy: 0.9390 - 22s/epoch - 22ms/step
Epoch 2/5
1000/1000 - 21s - loss: 0.0332 - accuracy: 0.9887 - val_loss: 0.1728 - val_accuracy: 0.9350 - 21s/epoch - 21ms/step
Epoch 3/5
1000/1000 - 21s - loss: 0.0069 - accuracy: 0.9986 - val_loss: 0.2815 - val_accuracy: 0.9215 - 21s/epoch - 21ms/step
Epoch 4/5
1000/1000 - 21s - loss: 0.0042 - accuracy: 0.9994 - val_loss: 0.2080 - val_accuracy: 0.9380 - 21s/epoch - 21ms/step
Epoch 5/5
1000/1000 - 21s - loss: 0.0031 - accuracy: 0.9996 - val_loss: 0.2240 - val_accuracy: 0.9390 - 21s/epoch - 21ms/step
```

auc score 0.9853474853474853




```
yhat_classes1 = (model1.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes1))
```

```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94       999
           1       0.94      0.94      0.94      1001

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000
```

Appendix

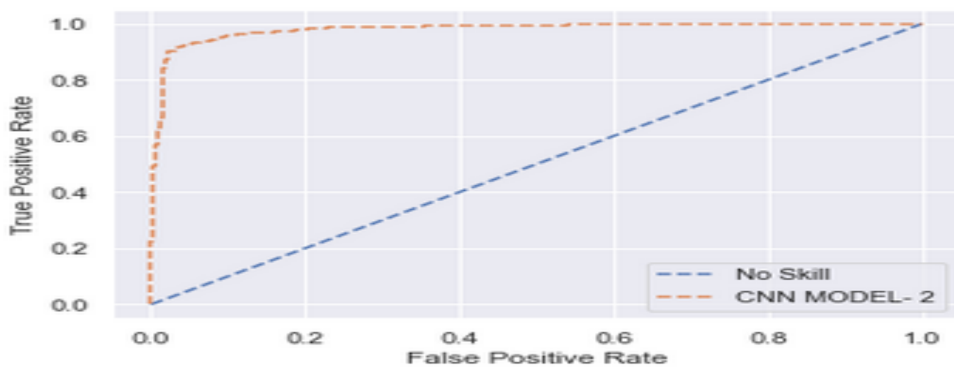
```
model2.add(Flatten())
model2.add(Dense(10, activation='relu'))
model2.add(Dense(1, activation='sigmoid'))
print(model2.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 13, 100)           2216900

 conv1d_1 (Conv1D)           (None, 13, 32)            25632

 max_pooling1d_1 (MaxPooling  (None, 6, 32)            0
 1D)

 conv1d_2 (Conv1D)           (None, 6, 64)             16448

 max_pooling1d_2 (MaxPooling  (None, 3, 64)            0
 1D)

 flatten_1 (Flatten)         (None, 192)               0

 dense_2 (Dense)             (None, 10)                1930

 dense_3 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,260,921
Trainable params: 2,260,921
Non-trainable params: 0
_________________________________________________________________
None
```

```
model2.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model2.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 22s - loss: 0.2474 - accuracy: 0.8920 - val_loss: 0.1643 - val_accuracy: 0.9355 - 22s/epoch - 22ms/step
Epoch 2/5
1000/1000 - 22s - loss: 0.0330 - accuracy: 0.9898 - val_loss: 0.1830 - val_accuracy: 0.9395 - 22s/epoch - 22ms/step
Epoch 3/5
1000/1000 - 21s - loss: 0.0119 - accuracy: 0.9962 - val_loss: 0.3673 - val_accuracy: 0.9070 - 21s/epoch - 21ms/step
Epoch 4/5
1000/1000 - 21s - loss: 0.0061 - accuracy: 0.9986 - val_loss: 0.2429 - val_accuracy: 0.9375 - 21s/epoch - 21ms/step
Epoch 5/5
1000/1000 - 22s - loss: 0.0059 - accuracy: 0.9987 - val_loss: 0.2360 - val_accuracy: 0.9245 - 22s/epoch - 22ms/step
```

auc score 0.9804839804839804




```
yhat_classes2 = (model2.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes2))
```

```
              precision    recall  f1-score   support

           0       0.96      0.89      0.92       999
           1       0.90      0.96      0.93      1001

    accuracy                           0.92      2000
   macro avg       0.93      0.92      0.92      2000
weighted avg       0.93      0.92      0.92      2000
```

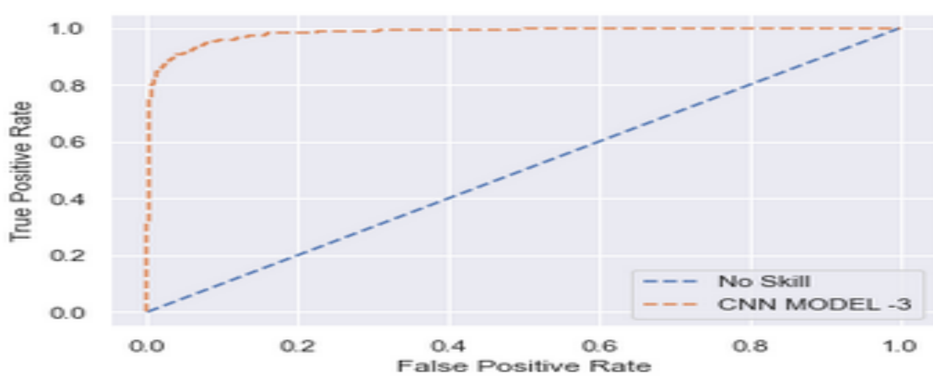
Appendix

```
 embedding_2 (Embedding)     (None, 13, 100)           2216900

 conv1d_3 (Conv1D)           (None, 13, 32)            25632

 max_pooling1d_3 (MaxPooling  (None, 6, 32)            0
 1D)

 conv1d_4 (Conv1D)           (None, 6, 64)             16448

 max_pooling1d_4 (MaxPooling  (None, 3, 64)            0
 1D)

 conv1d_5 (Conv1D)           (None, 3, 128)            65664

 max_pooling1d_5 (MaxPooling  (None, 1, 128)           0
 1D)

 flatten_2 (Flatten)         (None, 128)               0

 dense_4 (Dense)             (None, 10)                1290

 dense_5 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,325,945
Trainable params: 2,325,945
Non-trainable params: 0
_________________________________________________________________
None
```

```
model3.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model3.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 22s - loss: 0.2700 - accuracy: 0.8794 - val_loss: 0.1625 - val_accuracy: 0.9330 - 22s/epoch - 22ms/step
Epoch 2/5
1000/1000 - 22s - loss: 0.0352 - accuracy: 0.9887 - val_loss: 0.1605 - val_accuracy: 0.9380 - 22s/epoch - 22ms/step
Epoch 3/5
1000/1000 - 21s - loss: 0.0082 - accuracy: 0.9975 - val_loss: 0.2412 - val_accuracy: 0.9430 - 21s/epoch - 21ms/step
Epoch 4/5
1000/1000 - 21s - loss: 0.0069 - accuracy: 0.9981 - val_loss: 0.3245 - val_accuracy: 0.9315 - 21s/epoch - 21ms/step
Epoch 5/5
1000/1000 - 21s - loss: 0.0079 - accuracy: 0.9983 - val_loss: 0.3450 - val_accuracy: 0.9320 - 21s/epoch - 21ms/step
```

auc score 0.9837139837139837




```
yhat_classes3 = (model3.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes3))
```

```
              precision    recall  f1-score   support

           0       0.91      0.96      0.93       999
           1       0.95      0.91      0.93      1001

    accuracy                           0.93      2000
   macro avg       0.93      0.93      0.93      2000
weighted avg       0.93      0.93      0.93      2000
```

Appendix

Bi - Lstm Cnn Layer

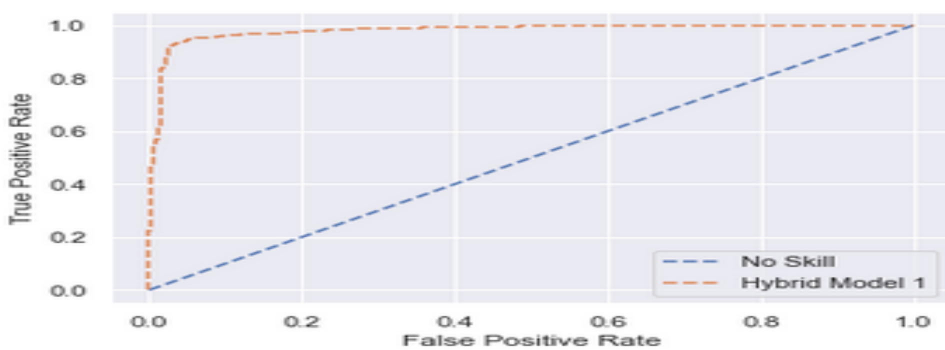
```
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D,Flatten,Dense
model1 = Sequential()
model1.add(Embedding(vocab_size, 100, input_length=max_len))
model1.add(Conv1D(filters=32, kernel_size=8, activation='relu'))
model1.add(MaxPooling1D(pool_size=2))
model1.add(Bidirectional(LSTM(100, dropout=0.2)))
model1.add(Flatten())
model1.add(Dense(10, activation='relu'))
model1.add(Dense(1, activation='sigmoid'))
print(model1.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 13, 100)           2216900

 conv1d (Conv1D)             (None, 6, 32)             25632

 max_pooling1d (MaxPooling1D  (None, 3, 32)            0
 )

 bidirectional (Bidirectiona  (None, 200)              106400
 l)

 flatten (Flatten)           (None, 200)               0

 dense (Dense)               (None, 10)                2010

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,350,953
Trainable params: 2,350,953
Non-trainable params: 0
_________________________________________________________________
None
```

```
model1.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model1.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 24s - loss: 0.2534 - accuracy: 0.8916 - val_loss: 0.1720 - val_accuracy: 0.9365 - 24s/epoch - 24ms/step
Epoch 2/5
1000/1000 - 22s - loss: 0.0308 - accuracy: 0.9908 - val_loss: 0.1972 - val_accuracy: 0.9370 - 22s/epoch - 22ms/step
Epoch 3/5
1000/1000 - 22s - loss: 0.0081 - accuracy: 0.9984 - val_loss: 0.2877 - val_accuracy: 0.9330 - 22s/epoch - 22ms/step
Epoch 4/5
1000/1000 - 22s - loss: 0.0082 - accuracy: 0.9979 - val_loss: 0.2250 - val_accuracy: 0.9455 - 22s/epoch - 22ms/step
Epoch 5/5
1000/1000 - 22s - loss: 0.0056 - accuracy: 0.9989 - val_loss: 0.2400 - val_accuracy: 0.9480 - 22s/epoch - 22ms/step
```

auc score 0.98024




```
yhat_classes1 = (model1.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes1))
```

```
              precision    recall  f1-score   support

           0       0.94      0.96      0.95      1000
           1       0.96      0.94      0.95      1000

    accuracy                           0.95      2000
   macro avg       0.95      0.95      0.95      2000
weighted avg       0.95      0.95      0.95      2000
```

Appendix

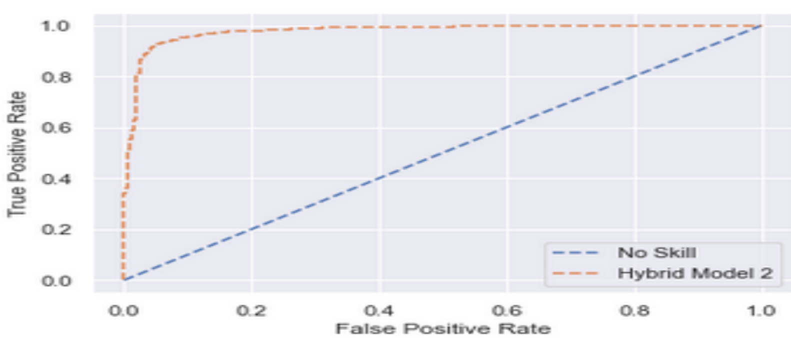
```
model2.add(Bidirectional(LSTM(100, dropout=0.2,return_sequences=True)))
model2.add(Bidirectional(LSTM(100, dropout=0.2)))
model2.add(Flatten())
model2.add(Dense(10, activation='relu'))
model2.add(Dense(1, activation='sigmoid'))
print(model2.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 13, 100)           2216900

 conv1d_1 (Conv1D)           (None, 13, 32)            25632

 max_pooling1d_1 (MaxPooling  (None, 6, 32)            0
 1D)

 conv1d_2 (Conv1D)           (None, 6, 64)             16448

 max_pooling1d_2 (MaxPooling  (None, 3, 64)            0
 1D)

 bidirectional_1 (Bidirectio  (None, 3, 200)           132000
 nal)

 bidirectional_2 (Bidirectio  (None, 200)              240800
 nal)

 flatten_1 (Flatten)         (None, 200)               0

 dense_2 (Dense)             (None, 10)                2010

 dense_3 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,633,801
Trainable params: 2,633,801
Non-trainable params: 0
_________________________________________________________________
None
```

```
model2.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model2.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 29s - loss: 0.2629 - accuracy: 0.8804 - val_loss: 0.1691 - val_accuracy: 0.9400 - 29s/epoch - 29ms/step
Epoch 2/5
1000/1000 - 29s - loss: 0.0317 - accuracy: 0.9908 - val_loss: 0.3091 - val_accuracy: 0.9270 - 29s/epoch - 29ms/step
Epoch 3/5
1000/1000 - 27s - loss: 0.0144 - accuracy: 0.9961 - val_loss: 0.2888 - val_accuracy: 0.9295 - 27s/epoch - 27ms/step
Epoch 4/5
1000/1000 - 24s - loss: 0.0072 - accuracy: 0.9985 - val_loss: 0.3447 - val_accuracy: 0.9370 - 24s/epoch - 24ms/step
Epoch 5/5
1000/1000 - 24s - loss: 0.0076 - accuracy: 0.9989 - val_loss: 0.2876 - val_accuracy: 0.9325 - 24s/epoch - 24ms/step
```

```
auc score 0.9779380000000001
```




```
yhat_classes2 = (model2.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes2))
```

```
              precision    recall  f1-score   support

           0       0.95      0.91      0.93      1000
           1       0.91      0.95      0.93      1000

    accuracy                           0.93      2000
   macro avg       0.93      0.93      0.93      2000
weighted avg       0.93      0.93      0.93      2000
```

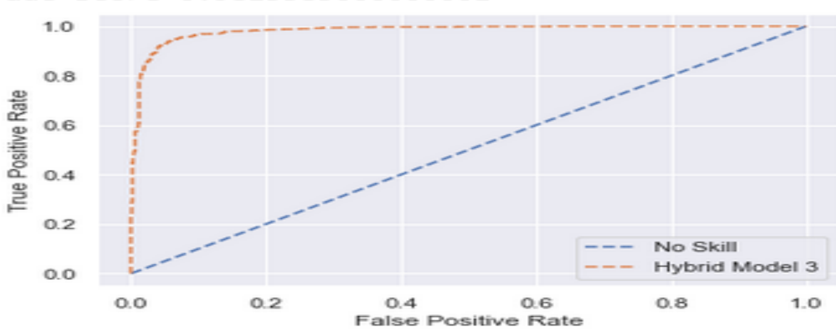
Appendix

```
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 13, 100)           2216900

 conv1d_3 (Conv1D)           (None, 13, 32)            25632

 max_pooling1d_3 (MaxPooling  (None, 6, 32)            0
 1D)

 conv1d_4 (Conv1D)           (None, 6, 64)             16448

 max_pooling1d_4 (MaxPooling  (None, 3, 64)            0
 1D)

 conv1d_5 (Conv1D)           (None, 3, 64)             32832

 max_pooling1d_5 (MaxPooling  (None, 1, 64)            0
 1D)

 bidirectional_3 (Bidirectio  (None, 1, 200)           132000
 nal)

 bidirectional_4 (Bidirectio  (None, 1, 200)           240800
 nal)

 bidirectional_5 (Bidirectio  (None, 200)              240800
 nal)

 flatten_2 (Flatten)         (None, 200)               0

 dense_4 (Dense)             (None, 10)                2010

 dense_5 (Dense)             (None, 1)                 11

=================================================================
Total params: 2,907,433
Trainable params: 2,907,433
Non-trainable params: 0
_________________________________________________________________
None
```

```
model3.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics=['accuracy'])
num_epochs = 5
history = model3.fit(train_pad_seq,Y_train,batch_size=8, epochs=num_epochs,validation_data=(test_pad_seq, Y_test), verbose=2)
```

```
Epoch 1/5
1000/1000 - 31s - loss: 0.2614 - accuracy: 0.8815 - val_loss: 0.1551 - val_accuracy: 0.9400 - 31s/epoch - 31ms/step
Epoch 2/5
1000/1000 - 26s - loss: 0.0314 - accuracy: 0.9900 - val_loss: 0.1817 - val_accuracy: 0.9440 - 26s/epoch - 26ms/step
Epoch 3/5
1000/1000 - 26s - loss: 0.0123 - accuracy: 0.9976 - val_loss: 0.1516 - val_accuracy: 0.9470 - 26s/epoch - 26ms/step
Epoch 4/5
1000/1000 - 25s - loss: 0.0079 - accuracy: 0.9983 - val_loss: 0.2611 - val_accuracy: 0.9360 - 25s/epoch - 25ms/step
Epoch 5/5
1000/1000 - 24s - loss: 0.0090 - accuracy: 0.9980 - val_loss: 0.3349 - val_accuracy: 0.9290 - 24s/epoch - 24ms/step
```

```
auc score 0.9825585000000001
```




```
yhat_classes3 = (model3.predict(X_test) > 0.5).astype("int32")
from sklearn.metrics import classification_report
print(classification_report(y_test, yhat_classes3))
```

```
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      1000
           1       0.96      0.90      0.93      1000

    accuracy                           0.93      2000
   macro avg       0.93      0.93      0.93      2000
weighted avg       0.93      0.93      0.93      2000
```

Appendix

The LSTM algorithm's strengths, limitations and solutions

No.	Strengths	Limitations	Solution
1	**Feature extraction:** By eliminating the requirement for laborious feature engineering, LSTMs automatically retrieve pertinent features from the input data. This has the distinct benefit of capturing both explicit and implicit linguistic patterns indicative of clickbait, which is helpful in clickbait detection.	**Overfitting:** LSTM algorithms can overfit models, especially when the model is too complex or few data observations are available.	Regularize the LSTM model by using strategies like dropout, which disables a section of the LSTM units at random during training to avoid over-dependence on particular characteristics.
2	**Learning contextual representations:** Contextual representations of words and phrases within a sequence can be learned by LSTMs. This skill aids in capturing the subtle semantic meaning and contextual cues that are crucial for clickbait detection. LSTMs can better distinguish between clickbait and non-clickbait content by understanding the situations where words are used.	**Imbalanced datasets:** There is always a mismatch between the number of clickbait and non-clickbait data, which can affect the LSTM's performance.	To resolve class imbalance, use data balancing approaches like undersampling the majority class, oversampling the minority class or combining both. In this study, class imbalance was not a concern as our dataset is already balanced
3	**Handling variable-length input:** The length of clickbait headlines can change and LSTM networks can handle cycles of various sizes. Because of their adaptability, LSTMs can handle clickbait material with a range of several words and can manage various clickbait headline structures.	**Out-of-vocabulary words:** It can be challenging to catch certain word embeddings that are absent from the trained clickbait headline data set. Their significance and context may not be captured by the LSTM.	To handle terms outside lexicons, use methods like sub word tokenization or character-level tokenization.
4	**Capturing sequential patterns:** Long-range dependencies and sequential patterns in text data are captured by LSTMs. Subtle linguistic clues and patterns that are dispersed across several phrases or words are frequently used in clickbait headlines. Since they can effectively simulate these dependencies, LSTMs are useful for identifying clickbait.	**Language complexities:** The linguistic structures and traits of the English and Kurdish languages are different from one another. These linguistic intricacies may not be captured by the LSTM.	When training the LSTM model, take into account linguistic elements that are language-specific, like semantic, syntax and morphology principles. Use pre-trained language embeddings or models to capture certain language details.
5	-	**Unavailability of data:** To learn and generalize the data in English and Kurdish, the LSTM may need many labelled training data.	The collection and annotation of a sizable, high-quality dataset for clickbait recognition in English and Kurdish.

Source: Compiled by the Researcher based on empirical deductions.

Appendix

The BILSTM algorithm's strengths, limitations and solutions

No.	Strengths	Limitations	Solution
1	**Language-agnostic architecture:** The BILSTM algorithm can be used to perform clickbait detection tasks in several languages because it is language-independent. BILSTM can successfully identify nuances and language-specific features that identify clickbait headlines by training on language-specific clickbait detection datasets in English and Kurdish.	**Overfitting and hyperparameter tuning:** BILSTM models can overfit, particularly when working with scant amounts of labelled data.	To prevent overfitting, appropriate regularization methods like dropout or L2 regularization should be used. Finding the right BILSTM model configuration also requires tweaking the hyperparameters, including regularization parameters, batch size, learning rate, and LSTM units.
2	**Generalization ability:** BILSTM has demonstrated strong generalization capabilities, enabling it to efficiently learn from little labelled data. It may identify underlying trends and extrapolate them to hypothetical clickbait instances. When clickbait detection datasets are tiny or there are insufficient resources for detailed labelling, this strength can be quite helpful.	**Need for sufficient labelled data:** For efficient training and generalization, BILSTM needs a sizable amount of labelled data, just like any machine learning model. It might be difficult to gather a comprehensive, interesting, and representative clickbait detection dataset that is unique to English and Kurdish.	To ensure adequate coverage of clickbait traits, efforts should be undertaken to curate or enhance the dataset.
3	**Capturing contextual information:** By taking into account both past and future words in a sequence, BILSTM can efficiently capture contextual information in clickbait headlines. The model can comprehend the relationships between words in a headline and produce more accurate predictions thanks to the use of forward and backward LSTM layers. These layers detect dependencies in both directions.	**Difficulty in interpretability:** Similar to other deep learning models, BILSTM models can be difficult to comprehend. With BILSTM alone, it can be challenging to comprehend which particular words or phrases contribute to the clickbait detection choice.	To learn more about the process of making choices for the model, additional analysis approaches may be used. These can include gradient-based attribution methods and attention mechanisms.
4	**Language modelling:** BILSTM models learn complex linguistic representations when trained on extensive language modelling challenges. Clickbait detection can be done using general language patterns that the model has learned. Clickbait can be identified with this pre-training.	**Limited parallelization:** BILSTM's sequential design makes it difficult to utilize parallel computing resources to their full potential. Due to the bidirectional nature of the model, parallelism is limited. The training period and inference speed may be slower than models that can be parallelized, like CNNs.	Investigate semi-supervised learning techniques that mix a smaller labelled dataset with a larger unlabeled dataset. Adding unlabeled data to the model can enhance its performance. Furthermore, active learning techniques can be used to choose the most instructive samples for manual labelling, minimizing labelling work while enhancing the model's performance.
5	**Handling long-term dependencies:** The exploding/vanishing gradient problem can be reduced and long-term dependencies in sequential data can be efficiently modelled using LSTM, a version of the RNN. This is especially helpful for detecting clickbait since it enables the model to catch semantic dependencies and relationships between phrases or words that are dispersed throughout a headline.	**Computational complexity:** When compared to more straightforward models like conventional machine learning algorithms or logistic regression, BILSTM models are computationally more expensive.	

Source: Compiled by the Researcher based on empirical deductions.

Appendix

The GRU algorithm’s strengths, limitations and solutions

No.	Strengths	Limitations	Solution
1	**Language-agnostic architecture:** The GRU algorithm can be used to perform clickbait detection tasks in several languages because it is language-independent. GRU can successfully capture language-specific patterns and subtleties that differentiate clickbait headlines by training it on datasets for clickbait identification in English and Kurdish.	**Need for sufficient labelled data:** For efficient training and generalization, GRU needs a lot of labelled data, just like any machine learning model. It might be difficult to gather a comprehensive, interesting, and representative clickbait detection dataset unique to English and Kurdish. To ensure adequate coverage of clickbait traits, efforts should be undertaken to curate or enhance the dataset.	Explore more advanced architectures designed to address the limitations of traditional RNNs. For example, you can investigate the effectiveness of using the GRU with additional mechanisms like attention or residual connections.
2	**Handling variable-length inputs:** GRU handles variable-length inputs, such as clickbait headlines with varying word counts. It can process headlines of different lengths without additional preprocessing steps like padding or truncation. This flexibility makes GRU suitable for clickbait detection in languages where headlines vary in length, such as English and Kurdish.	**Difficulty in interpretability:** Similar to other deep learning models, interpreting the decision-making process of GRU can be challenging. Understanding which specific words or phrases contribute to the clickbait detection decision can be difficult with GRU alone.	Use interpretation methods like layer-wise relevance propagation (LRP) or attention mechanisms to obtain an understanding of the GRU model's decision-making process.
3	**Generalization ability:** GRU has demonstrated strong generalization capabilities, enabling it to efficiently learn from scant-labelled data. It may identify underlying trends and extrapolate them to hypothetical clickbait instances. When clickbait detection datasets are limited or resources are insufficient for detailed labelling, this strength can be quite helpful.	**Lack of parallelization:** RNNs, including GRU, are sequential by design, which restricts how much parallel processing they can do. To fully utilize parallel computing resources, each timestep in the sequence depends on the previous timestep. Compared to parallelizable models like CNNs, this can slow down the training and inference processes.	To make the most of parallel computing resources, consider adopting methods like mini-batching and improving model implementation. Training and inference procedures can be sped up, lessening the effect of the GRU algorithm's sequential nature. This is done by processing numerous sequences concurrently, either on a single GPU or across multiple GPUs.
4	**Computational efficiency:** The GRU has a simpler architecture and fewer parameters than its sibling, the LSTM. Because of this, GRU is more quickly trainable and computationally efficient, especially for large-scale clickbait detection tasks. It uses comparatively fewer computational resources to process sequential data.	**Vanishing/exploding gradients:** RNNs like GRU may experience exploding or vanishing gradients during training. These problems may make it more difficult for the model to recognize and learn from meaningful patterns in the data. These issues can be reduced by employing methods like gradient clipping or more sophisticated designs (such as LSTM).	To keep the gradients from getting too big or disappearing entirely during training, use gradient clipping techniques. Gradient clipping reduces the gradients' magnitude and aids in maintaining training stability.

Source: Compiled by the Researcher based on empirical deductions.

Appendix

The CNN algorithm’s strengths, limitations and solutions

No.	Strengths	Limitations	Solution
1	**Feature hierarchies:** Multiple layers are frequently featured in CNN architectures, enabling them to learn hierarchical representations of features. Deeper layers capture more complicated data, such as word or phrase pairings. CNNs can comprehend the compositional structure of clickbait headlines and identify higher-level patterns thanks to this hierarchical feature learning.	**Language-specific challenges:** Despite the fact that CNNs' architecture is language-neutral, they could still have trouble detecting clickbait in some languages. Kurdish is one example of a language that may have particular cultural nuances, idiomatic expressions or linguistic characteristics that should be carefully taken into account while training models. It can be difficult to gather adequate and representative training data for languages other than English.	Collect and manage datasets that are language-specific for Kurdish clickbait identification, concentrating on headlines from pertinent sources. This guarantees that the model is exposed to linguistic nuances and characteristics that are language-specific.
2	**Scalability:** CNNs are effective at processing massive datasets because of their computational efficiency. With minimal computing overhead, they can perform clickbait detection jobs requiring substantial amounts of textual data, including English and Kurdish languages. Dealing with large-scale clickbait detection applications benefits from its scalability.	**Lack of interpretability:** Since CNNs are "black-box" models, it might be difficult to understand how they make decisions. It may be challenging to comprehend how various traits or patterns affect clickbait detection when using CNNs. More clarity in comprehending the logic behind clickbait detection may be provided by interpretable models like decision trees or rule-based algorithms.	To determine which characteristics or elements of the input contribute to the clickbait detection judgment, use model interpretation techniques like attention mechanisms or gradient-based attribution methods
3	**Translation invariance:** Because CNNs are fundamentally translation-invariant, they can recognize patterns wherever they appear in the input. This trait enables CNNs, regardless of their position, to collect significant language clues and patterns within clickbait headlines. It renders CNNs resistant to changes in the order of phrases or words, which helps clickbait detection across linguistic boundaries.	**Fixed-size input:** The length of the clickbait headline needs to be set or preprocessed to a certain length because CNNs often want fixed-size inputs. It may be necessary to use supplementary preprocessing procedures like truncation or padding when dealing with headlines with variable word counts. This fixed-size input restriction may hinder CNN's ability to accurately process clickbait headlines of various lengths.	Make clickbait headlines shorter by preprocessing them with padding and truncation techniques. While truncation shortens larger headlines to the required length, padding adds dummy tokens to shorter headlines to ensure consistent input size. CNN can handle variable-length inputs thanks to these preprocessing processes.
4	**Local feature extraction:** Applying convolutional filters to input data allows CNNs to extract local features. CNNs can recognize particular patterns, such as phrases or words typical of clickbait headlines, in the context of clickbait detection. CNNs can distinguish between clickbait and non-clickbait information efficiently by knowing various regional characteristics.	**Limited sequential information:** CNNs may have trouble catching long-range relationships and repeated patterns because they are focused on capturing local features. Due to CNNs' limited ability to handle sequential information, detecting clickbait requires comprehending the context and relationships between words in a headline. Sequential dependencies may be better captured with LSTMs or other RNN versions.	Combine CNNs with RNNs like GRU or LSTM to detect both close-proximity relationships and distant features. This hybrid architecture uses CNN's local feature extraction capabilities to efficiently capture sequential patterns in clickbait headlines.

Source: Compiled by the Researcher based on empirical deductions.

Appendix

The Hybrid CNN BILSTM algorithm's strengths, limitations and solutions

No.	Strengths	Limitations	Solution
1	**Robustness to input length:** The Hybrid CNN BILSTM can analyze clickbait headlines of various lengths accurately since the BILSTM component of the model can handle variable-length inputs. When working with headlines with different word counts, this flexibility is especially useful.	**Hyperparameter tuning:** Numerous hyperparameters in the Hybrid CNN BILSTM architecture require careful tuning. To get the best performance, parameters including regularization strategies, learning rates, LSTM hidden units, kernel sizes, and filter sizes should be tuned systematically.	Using methods such as random search and random search or, hyperparameter tuning. Evaluate various combinations of hyperparameters, such as regularization strategies, learning rates, LSTM hidden units, kernel sizes, and filter sizes.
2	**Long-term dependency modelling:** By processing the input sequence both forward and backwards, the Hybrid CNN BILSTM component may efficiently capture long-range relationships. This improves the model's comprehension of the clickbait content by allowing it to gather contextual information and relationships between phrases or words in a headline.	**Interpretability challenges:** Hybrid CNN BILSTM models' interpretability may be constrained, just like with other deep learning models. It may be difficult to comprehend the precise traits or trends that influence clickbait detection choices. Gaining knowledge about the model's decision-making process may call for the use of additional interpretability techniques and tools.	Apply model interpretation methods to understand the decision-making of the Hybrid CNN BILSTM. Approaches like gradient-based attribution methods might draw attention to key elements or areas of the input that affect the choice to identify clickbait.
3	**Hierarchical feature learning:** The Hybrid CNN BILSTM's CNN component is capable of learning hierarchical feature representations. It gradually captures higher-level features in deeper levels after capturing low-level features in the first few layers. The model is assisted in comprehending the compositional structure of clickbait headlines by this hierarchical learning of features, which enables it to identify intricate patterns.	**Need for large training data:** In order to learn well, hybrid CNN BILSTM models frequently need a lot of labelled training data. It can be difficult to gather a substantial and varied clickbait detection dataset for English and Kurdish. To guarantee the availability of a representative dataset for training the model, data collecting and annotation efforts should be made.	Use approaches for data augmentation to expand the clickbait detection dataset. This can involve methods like creating artificial clickbait examples, back-translation or random word substitution based on linguistic peculiarities.
4	**Language-specific adaptability:** The Hybrid CNN BILSTM may easily adjust to the peculiarities and qualities of several languages. The model may learn to recognize cultural nuances, colloquial idioms and language-specific patterns that are suggestive of clickbait by training on language-specific datasets in English and Kurdish. The Hybrid CNN BILSTM is flexible enough to be used for clickbait detection across many languages.	**Computational complexity:** More computing is required to run the Hybrid CNN BILSTM architecture than standalone CNN or BILSTM models. In comparison to simpler models, training and inference with the Hybrid CNN BILSTM may take more time and computer resources. To accommodate the rising complexity, there needs to be a sufficient computing system.	Accelerate the training and inference of the Hybrid CNN BILSTM model by using Graphics Processing Units (GPUs) or distributed computing frameworks.

Source: Compiled by the Researcher based on empirical deductions

IEC2023

9th International Engineering Conference

Acceptance and Invitation Letter

Subject: Acceptance Letter **Date:** February 1 , 2023

Dear Ibrahim Abdulkhaleq and Shahab Kareem

I am pleased to inform you that your paper entitled “**A Deep Learning Approach and Recurrent Neural Network Towards Clickbait Detection**” with Paper ID: **1570876397** has been accepted for **presentation** in the 9th International Engineering Conference (IEC2023) to be held in Erbil, Kurdistan Region-Iraq on February 15-16, 2023 organized by Tishk International University and Erbil Polytechnic University and technically sponsored by IEEE and IEEE Iraq Section.

The **Opening Ceremony** will take place at **Erbil International Hotel at 10:00 am on February 15, 2023**. You are supposed to register between 8:30 am and 9:30 am.

Please be aware that all **presentation sessions** will be held on **February 15-16, 2023 at Erbil International Hotel**. Presentations will take 15 minutes plus 10 minutes for questions and answers. You can get program flow, timetable and halls of presentations online from IEC2023 Conference website:

www.tiu.edu.iq/conf/iec

We are looking forward to seeing you on the Conference Days amongst us.

Dr. Abubakar M. Ashir
Chair of IEC2022 Conference

P.S.: Kindly be advised that this article acceptance letter is not an official document which can be considered for any academic promotion unless it is presented and discussed at the conference.

Location for Opening Ceremony: Erbil International Hotel
Date: February 15, 2023
Time: 10:00 am
Place: Erbil International Hotel / 30-meter Street, Erbil, Kurdistan Region-Iraq, Tel. +964 750 2600600

DATE: 27-07-2023
Manuscript ID: UTMJ.2023.1280234

LETTER OF ACCEPTANCE

Dear Author's:

Ibrahim Abdulkhaleq
Dept. Information System Engineering
Technical Engineering College, Erbil Polytechnic University, Iraq
Ibrahim.abdulkhaliq@epu.edu.iq
Shahab Wahhab Kareem
Dept. Information System Engineering,
Technical Engineering College,Erbil Polytechnic University, Iraq
shahab.kareem@epu.edu.iq

It is a great pleasure to inform that, after the peer review, your article titled " **Machine Learning Techniques For Detecting Fake News in the Kurdish language**", has been accepted to publish in the journal of Utilitas Mathematica.

Comments from the Editorial Board:
Article plagiarism is below 10%.
Article perfectly fits in the scope.
Article is well written.
No more corrections are needed.

Hence, the article is accepted to be published in the upcoming Regular issue 2023. Utilitas Mathematica is a scientific journal. For any enquires/complaints, please email to editor.utilitasmathematica@gmail.com.

Note: The Journal index in the Scopus Database Category Quartile, Scopus Q4.

Olivia Wilson

Journal Editorial Office
UTILITAS MATHEMATICA
ISSN: 0315-3681
http://utilitasmathematica.com/index.php/Index/index

پوختە

ئامانجی توێژینەوەکە هەڵسەنگاندنی ئەدای تۆڕی دەماری قووڵە بۆ دۆزینەوەی کلیک بەیت بە زمانەکانی ئینگلیزی و کوردی. بۆ دیاریکردنی کلیکبەیت بە زمانی کوردی، ١٠ هەزار داتای بابەتی هەواڵی کوردی لە پلاتفۆرمە جۆراوجۆرەکانی هەواڵی کوردییەوە کۆکرانەوە. ئێمە کۆمەڵە داتایەکمان بەکارهێنا کە لە ٣٢ هەزار سەردێڕی ئینگلیزی پێکهاتبوو، کە لەلایەن چاکرابۆرتییەوە سەرپەرشتی کرابوو. لە ڕێگەی ئەم جۆرە ڕێبازانە، ئەدای بیرگەی ، بیرەوەری کورتخایەنی درێژخایەنی دوو ئاراستەیی(LSTM) کورتخایەنی درێژخایەن ، (GRU) ، یەکەی دووبارەبوونەوەی دەروازەدار(CNN) ، تۆڕی دەماری پێچاوپێچ(BILSTM) و تیکەڵەیەک لە تۆڕی دەماری پێچاوپێچ لەگەڵ درێژی دوو ئاراستەیی تەکنیکەکانی بۆ هەردوو زمانی ئینگلیزی و (CNN BILSTM) دیاریکردنی کلیکبەیت بیرگەی کورتخایەن کوردی هەڵسەنگێندرا. دۆزینەوەکان دەریانخست کە تێگەیشتن لە تایبەتمەندییە تایبەتەکانی زمان، شێوازە زمانەوانییەکان و نۆرمەکانی کولتووری ئامانجدارەکان زۆر گرنگە بۆ دەستنیشانکردنی کلیکبەیت لە سنووری زمانەوانی و کولتووری ئینگلیزی و کوردیدا. ئەلگۆریتمەکەی بیرگەی کورتخایەنی درێژخایەنی دوو ئاراستەیی وەک کاریگەرترین ڕیزبەندی هەیە لە دیاریکردنی کلیکبەیت بە زمانی ئینگلیزی، بە ڕێژەی وردبینی 99.23%، ئەلگۆریتمەکەی .95% F1 95.33% وردبینی، 94.33% ڕێژەی بیرهێنانەوە، و نمرەی یەکەی دووبارەبووەوەی دەروازەدار لە ئەلگۆریتمەکانی تر باشتر کاردەکات لە دۆزینەوەی کلیکبەیت بە زمانی کوردی، بە ڕێژەی وردبینی 93.93%، 93.13% وردبینی، 95.17% توێژینەوەکە بەکارهێنانی تۆڕی دەماری .F1 ڕێژەی بیرهێنانەوە، و 94.13% نمرەی دووبارەبووەوە و شێوازەکانی فێربوونی قووڵ لە چوارچێوەی دیاریکردنی کلیکبەیتدا درێژدەکاتەوە، تواناکانیان لە شیکردنەوەی داتا دەقیەکان بە تایبەتمەندییە مانادارە نوانسییەکان نیشان دەدات. بە نیشاندانی ئەوەی کە ئەم تەکنیکانە کاریگەرن لە گرتنی ئاماژەی زمانی ورد

بۆ ئەرکەکانی پۆلێنکردن، ئەمە دەتوانێت بەشداربێت لە بواری فراوانتری پرۆسێسی زمانی سروشتی.

حکومەتی هەرێمی کوردستان – عێراق
سەرۆکایەتی ئەنجومەنی وەزیران
وەزارەتی خوێندنی باڵا و توێژینەوەی زانستی
زانکۆی پۆلیتەکنیکی هەولێر
بەشی ئەندازیاری سیستەمی زانیاری

بەرزکردنەوەی دۆزینەوەی کلیکبەیت لە ڕێگەی فێربوونی قووڵ و زیرەکی دەستکرد تایبەت بە زمانی ئینگلیزی و کوردی

تیزەکە

پێشکەشی ئەنجومەنی کۆلێژی تەکنیکی ئەندازیاری کراوە لە زانکۆی پۆلیتەکنیکی هەولێر وەک بەشێک لە پێداویستیەکانی بەدەست هێنانی پلەی ماستەر لە ئەندازیاری سیستەمی زانیاری

لەلایەن

ئیبراهیم شەماڵ عەبدولخالیق

بەکالۆریۆس لە ئەندازیاری سیستەمی زانیاری

بەسەرپەرشتیاری

پرۆفیسۆری یاریدەدەر دکتۆر شەهاب وەهاب کەریم

هەولێر-کوردستان

ڕێبەندان ٢٠٢٣