Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model

Birhanu Gebisa Muleta

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model

Birhanu Gebisa Muleta

Advisor: Mohd Wazih Ahmed (Assistant Professor)

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

Approval Page of M.Sc. Thesis

I, the advisors of the thesis entitled "**Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model**" and developed by Birhanu Gebisa, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Mohd Wazih Ahmed (Assistant Professor) _____ _____

Major Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by Birhanu Gebisa have read and evaluated the thesis entitled "**Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering (CSE).

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Finally, approval and acceptance of the thesis are contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____

Department Head Signature

_____ _____

School Dean Signature

_____ _____

Office of Postgraduate Studies, Dean Signature

Declaration

I hereby declare that this Master Thesis entitled ” **Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model**” is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Birhanu Gebisa		
Name of the Student	Signature	Date

Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Fake Review Detection for Online Electronics Marketing using Hybrid Deep Neural Network Model**" prepared under my guidance by Birhanu Gebisa submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering (CSE). Therefore, I recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Mohd Wazih Ahmed (Assistant Professor) _____ _____

Major Advisor Signature Date

ACKNOWLEDGEMENTS

First of all, I want to thank the Almighty God for making things possible and helping me in all aspects of my life. This study would not have been possible without his grace and blessings. Also, I would like to express my gratitude to the Virgin Mary, the Mother of God.

I would like to express my gratitude to Dr. Mohid Wazih Ahmed, my advisor, for his guidance, his knowledge, and his willingness to discuss the subject in depth.

I would like to thank the Computer science and engineering staff, Dr. Mesfin Abebe, Genet Shankoo (PhD candidate), Endris Ahmed (PhD candidate), SIG supervisor, and PG coordinator who deserve special thanks for their insightful comments, and helpful assistance with my thesis.

I thank my parents for their constant support and encouragement throughout my years of study and thesis. Without them, this achievement would not have been possible. I appreciate it.

I would also like to thank the entire Data Science Special Interest Group (DS Sig) and friends who shared their knowledge with me, as well as everyone else who helped with this research.

Table of Contents Pages

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ACRONYMS

A

ACNN: Attention Convolution Neural Network

AI: Artificial Intelligence

AMT: Amazon Mechanical Turk

API: Application Programming Interface

B

Bi-LSTM: Bidirectional Long Short Term Memory

BOW: Bag of Words

C

CBOW: Continuous Bag of Words

CNN: Convolution Neural Network

CS: Cuckoo Search

CV: Cross Validation

D

DFFNN: Deep Feed-Forward Neural Network

Doc2vec: Document to Vector

DRI: Deceptive Review Identification

DT: Decision Tree

E

EL: Ensemble Learning

G

GB: Gigabyte

GPU: Graphics Processing Unit

GRU: Gated Recurrent Unit

GUI: Graphical User Interface

H

HSL: Hierarchical Supervised Learning

I

ID: Identity Document

IDE: Integrated Development Environment

IDF: Inverse Document Frequency

K

KNN: k-nearest Neighbors

L

LR: Linear Regression

LSTM: Long Short-Term Memory

M

MLP: Multilayer Perceptron

MTurk: Mechanical Turk

N

NB: Naive Bayes

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

P

POS: Part-of-Speech

R

RAM: Random Access Memory

RCNN: Recurrent Convolutional Neural Network

RD: Reviewer Deviation

Relu: Rectified Linear Unit

RF: Random Forest

RNN: Recurrent Neural Network

S

SVM: Support Vector Machines

R

RD: Reviewer deviation

T

TF: Term Frequency

TF-IDF: Term Frequency Inverse Document Frequency

TPU: Tensor Processing Units

U

UML: Unified Modeling Language

URL: Uniform Resource Locator

US: United States

W

Word2vec: Word to Vector

X

XGBT: Extreme Gradient Boosting

Abstract

Internet has transformed the domestic market into an online business, and customer feedback has made a difference in determining a company's e-commerce revenue. The majority of consumers read reviews given by previous reviewers before making any purchase decision on products or services. Trustworthiness of online reviews is critical for organizations; it has an indirect or direct impact on industry profitability. Because of this impact, several companies use spammers to post fake reviews and also influence customer decisions. Fake review identification is a challenging and time-consuming process that requires a lot of resources in the area of electronics consumers.. And most of the prior research was focused on detecting reviews in the domains of hotels, health, mobile apps, and restaurants. However, existing fake review detection studies that use machine learning and deep learning techniques are ineffective due to issues such as low detection performance, unbalanced datasets, and small attributes of detection review used to build models. And, it ignores the context of words in a review and focuses on extracting traditional linguistic units from reviews. The attribute of review used in earlier studies was insufficient to identify fake review content, and the focus of review authentication was on the reviews' individual word meanings. This study proposed a fake review detection using a hybrid deep neural network model that combines CNN and BILSTM with word embedding. For experiments, this study used publicly accessible labeled datasets from Amazon electronic consumers. The study was carried out using sentiment analysis as the feature used to calculate the polarity and subjectivity of the reviews. The proposed hybrid CNN-BILSTM model extracts features from the input review contents using CNN layers and sequence learning using BILSTM layers. The study used the functional Keras API, which concatenated review input types such as behavioral and contextual features together. An effective feature extraction technique and more attributes were used to increase detection performance. In this study, we compared different deep neural network-based algorithms to find out which is the most important for fake review detector model development, such as CNN, BILSTM, MLP, CNN-MLP, CNN-GRU, and CNN-LSTM. Finally, based on our experiment, we found that the CNN-BiLSTM model performed the best for the proposed fake review detection, with an accuracy of 96.70%, which is 3.40% higher than the previous study and an f1-score of 98%.

Keywords: ***Fake review, deep learning, fake review detection, sentiment analysis, hybrid model, and CNN-BILSTM***

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In today's world, the internet is becoming very important for human beings in their day-to-day activities. One of these activities is online shopping, which is a business environment where consumers buy products from suppliers and the company sells its products through electronic means (Chen & Xie, 2008). The growth of internet and efficiency of technology has allowed the e-commerce business market to expand repeatedly. And also, misleading false information about products is rapidly spreading and becoming common on the internet. In 2021, the worldwide e-commerce market's online retail sales reached $28.5 trillion US dollars, according to a report published by the United Nations Conference on Trade and Development (UNCTAD, 2022).

People’s opinions on online business are gradually more used by companies and individuals for making choices of products, purchasing decisions, production design, and online marketing of products (Bhuvaneshwari et al., 2021). The growing popularity of internet reviews also encourages the fake review writing industry, which involves hiring human authors to generate evaluations that are misleading to sway readers' perceptions. Customers who are used to reading reviews before making a final purchase decision may have issues because those reviews may be influenced by people who are not consumers.

According to an online review survey conducted in 2018 (Otar, 2018), negative online reviews influenced 94% of customers to change their product choice, 73% of consumers trusted the positive reviews, and 50% of negative reviews raised concerns about the quality of the product. The report of the World Economic Forum on the review in 2021 (Jonathan Marciano, 2021) showed that an online review influenced global e-commerce by $3.8 trillion US dollars, and the fake review also had a direct impact of $252 billion on the global market. Online product reviews on the internet are crucially increasing, forming a new type of word-of-mouth recommendation (Chen & Xie, 2008) that has a significant impact on today's business. More than 70% of online consumers have believed that reviews of purchased products and about 85% of consumers indicate that online consumer reviews are a critical factor in determining their purchasing decision (Z. Liu et al., 2020).

Online reviews are recommendations from reviewers in digital form and greatly influence consumer purchasing decisions. It is observed that customers who spend more on e-commerce websites tend to write their sentiments about the services provided by the online platform, as their attachment to the platform is high. These feelings and attachments are realized in the form of reviews and product ratings. The customer review represents satisfaction or dissatisfaction with the quality of products and is used to provide descriptions of products for consumers. However, a review opportunity for an e-commerce business has its own challenges, which as influencing people's opinions and conveying opinions is more than simply stating facts (Martínez Otero, 2021). Fake reviews are fraudulent reviews given to mislead consumers in their buying decisions and are frequently given by individuals with little knowledge or no genuine experience reviewer with the products or services being evaluated (D. Zhang et al., 2016).

A fake review is one that was created by someone who had no experience with the products being reviewed to manipulate the ratings of the product in favor of (or against) (Lee, 2016). Detecting fake reviews is a major problem in the current internet world, and it's difficult to differentiate between product service reviews that reflect customer experiences and opinions (Threats, n.d.). There are different names for fake reviews that are often used in this research, which are defined as spam reviews, deceptive opinions, and reviewers of these reviews are also called spammers (Mohawesh, Tran, et al., 2021a). Fake review detection has received a lot of attention from various sources, including service providers, customers, product vendors, and the research community (Anusha P, 2020).

Most e-commerce companies allow reviews to be posted after a verified purchase of products, and fake reviewers can write reviews online for free with a high potential for business profit. However, some companies do not allow the posting of reviews. Buyers, on the other hand, read product reviews as they need them as references when making a purchase decision because they cannot physically examine tangible goods in the online shopping marketplace. According to the study (W. Zhang et al., 2018), humans can't distinguish between fake and genuine reviews to a reasonable degree. As a result, the review has a significant effect on the business company.

The existing studies have suggested ML techniques for fake review detection. Most of the current research focuses on behavioral features using machine learning and extracts traditional linguistic

meanings such as n-grams, POS tagging, and rule-based ratings to classify reviews as fake or genuine. The previous method tried to solve the problem of identifying fake reviews, but there are still gaps, and fake reviews are currently a hot research topic. Due to the limited feature representation and data modifications by the reviewer, fake review detection is a difficult task. (Mohawesh, Xu, et al., 2021a). The Recurrent Neural Networks[1] and Convolutional Neural Networks[2] are the main deep learning architectures for text classification. The hybrid neural network methods have the ability to improve generalization and great potential in text classification to solve the fake review detection problem with datasets (Moyo & Sibanda, 2015).

The fundamental purpose of employing hybrid deep neural network models to handle the concern of fake review verification approaches is to ensure the integrity of model features for the review procedures. The BILSTM techniques are bidirectional and capable of acquiring the sequence outline more efficiently, whereas the CNN techniques are capable of filtering the input feature extraction and extracting valuable features. CNN has the capacity to learn the scene features of the review, which are identified using the ability to manage spatial relations. BILSTM is used to extract temporal, contextual, and sequential features from review text. It identifies long-term relationships between the review entities and important features are captured (W. Zhang et al., 2018).

We use behavioral, and textual features of review content to determine the class. This study uses a sentimental polarity to categorize the review polarity into positive, negative, and neutral sentiments. Subjectivity is used to identify the reviewer's opinion about the product or discuss other issues not related to products. The helpful rating attribute is also used to categorize the usefulness of a review as either positive or negative. Classifying reviews based on whether they are positive or negative in terms of emotion rating and helpful features of the reviews. It predicts whether the reviews would be fake or genuine based on the terms used in the text, the rating given to the review, the helpful rating, and textual features.

To solve the challenge of detecting fake reviews and improving the performance of the current model, we built a hybrid neural network method by incorporating CNN and BILSTM deep learning

[1] https://machinelearningmastery.com/crash-course-recurrent-neural-networks-deep-learning/
[2] https://machinelearningmastery.com/crash-course-convolutional-neural-networks/

techniques. This study proposed a new architecture for the problem based on the Keras functional API python library. To ensure the model's performance, evaluate the model using evaluation metrics and stratified k-fold cross-validation. Also, compare the proposed model with CNN, BILSTM, CNN-MLP, CNN-GRU, and CNN-LSTM deep learning techniques, as well as baseline work. Finally, based on the features of the review and the stakeholder reviewer, this research addresses the difficulty of detecting fake reviews on electronic products.

1.2. The motivation of the Study

Reviews have gained great prominence around the world, and different online business sectors that can attract the attention of online reviews. And the impact of the review also increased globally, with both negative and positive effects on the e-commerce company's performance (Otar, 2018). The popularity of online reviews has increased, which encourages unwanted competition between product sellers and service providers to gain more customers by fabricating fake reviews (Paul & Nikolaev, 2021). On the other hand, the volume of review data is enormous, requiring the use of a modeling technique capable of handling such a large amount of data.

Deep learning approaches are more suitable to deal with these challenges. When given a lot of high-quality data to work with, deep learning techniques perform best, and as the amount of data accessible rises, so does their effectiveness in solving the problem. The number of customers using the internet to purchase services or products is growing with the number of product providers, and fake reviews are also growing in the same way.

1.3. Statement of the Problem

People can currently purchase products according to their needs, but they consider market trends as well as inspect product reviews. Customers frequently buy items online from various websites, where they read reviews to find out whether the item is suitable for their intended use and information about the product's pricing, durability, and service quality. Customers who have made purchases can post reviews of their experiences online. For a product, there would be thousands of comments, making it difficult to identify which reviews are genuine or fake. Difficult to detect fake reviews on the internet market for new customers, they rely on online reviews as a source of information when making a purchasing decision.

Some businesses work to produce genuine reviews for themselves, while others work to damage the reputations of their rivals by creating negative online reviews. It is difficult to detect whether product reviews on e-commerce sites are genuine or not due to the dynamic nature of the reviews and the review features that have changed over time (Mohawesh, Xu, et al., 2021a). Fake review detection is a complex and time-consuming task that requires a significant amount of resources, time, and attention to determine whether a review's contents are real or not. A fake reviewer often uses generic words to comment on products since they are not about technicalities, but rather they write a good word that describes the product or a negative word that attacks the products of companies. For customers, identifying a fake review is a difficult task because most fake reviewers do not have official methods for reviews to easily identify online reviews.

Due to the limited feature representations and the dynamic content modifications of reviews by spammers, identifying the review content is difficult from domain to domain. There is no control mechanism in place for a customer, who may post anything about products on e-commerce platforms, and a substantial number of false evaluations can be produced with ease. From a semantic point of view and due to the scarcity of features, it is a challenge to learn the fundamental laws of data and to extract non-local semantic information from a sentence and represent it from the perspective of global discourse structures in a text (Tim, Alan, 2021).

The challenge of detecting fake reviews is distinguishing reviews, which reading a large amount of review content from different stakeholders, it is quite difficult to recognize fake reviews (Kumar, 2020). This is concerning because the difference between reviews cannot be noticed by the naked eye. Similarly, previous research on detecting fake reviews focuses on machine learning techniques to evaluate methodologies for fact verification and detecting fake reviews to obtain more accurate findings. To solve the problem above, conduct an investigation using hybrid neural network techniques with appropriate feature and performance evaluation methods.

1.4. Research Questions

In this study, we design three research questions to address.

RQ1: How does sentiment analysis improve the performance of the fake review detection model?

RQ2: Does combining the textual and behavioral features have an impact on improving the fake review detection performance?

RQ3: Which deep learning algorithms improve the detection performance of fake reviews?

1.5. Objective of the Study

1.5.1. General Objectives

The main goal of this study is to build a model for fake review detection in e-commerce in the case of electronic consumers using a hybrid deep neural network model.

1.5.2. Specific Objectives:

The specific objectives of this research are to achieve the general objective of carrying out the following tasks:

- Explore datasets and appropriate preprocessing procedures for fake review detection.
- To develop the deep neural network model by analyzing the sentiment of the review feature.
- Examine how several suitable deep learning algorithms can be used to reduce fake reviews.
- Handle the multiple review feature using the functional API of Keras to build the hybrid model.
- Design and demonstrate an appropriate hybrid neural network model for fake review detection.

1.6. Scope and Limitations

1.6.1. Scope of the Study

The scope of this research focuses on designing a hybrid deep neural network model that detects fake reviews from the reviews given by reviewers of products. We were able to focus the investigation on the review and reviewer features because the datasets we utilized did not contain product verification and business network attributes. Based on the previous research, we implemented six deep learning hybrid models, such as CNN, MLP, GRU, LSTM, and BILSTM, together with glove word embedding, and evaluated them using stratified k-fold cross-validation and metrics evaluation.

1.6.2. Limitation of the Study

This study has some limitations, such as the investigation being limited only to customer reviews to detect fake electronics consumer reviews in online business marketing. This research cannot detect emojis and spelling errors to detect fake reviews. The study is based on publicly available datasets due to a lack of labeled datasets. We were also unable to create models for the detection of fake reviews using non-textual data due to a lack of labeled data. And the study does not include group spammers and non-reviews, such as advertisements.

1.7. Significance of the Study

The significance of this study can assist in minimizing and identifying disinformation about products with some improvement and full model integration. It can help in better decision-making for fake review detection. However, it does not say that the work can be independently filtered out of any genuine feedback from the various consumer reviews. It improves the identification of fake reviews in online businesses' marketing of electronic retailers. Additionally, this study can be a great asset for service providers to control unwanted computation between companies that produce similar products. For those who are interested in fake review detection and deep learning algorithms, the study findings can serve as a baseline for further analysis and investigation.

1.8. Organization of the Thesis

This study is organized into six different chapters, which are listed below:

Chapter One: The first chapter describes the introduction, which provides the basic concept of the investigation. Discuss the study motivation, problem statement, general and particular objectives, scope and limitations of the fake review detection research, and finally the study's importance.

Chapter Two: This chapter provides a comprehensive review of the research on fake review identification as well as supporting ideas. Evaluate the existing system and critically examine related works of study to identify previous research, contributions, limitations, and gaps in knowledge, as well as related works that further the understanding of the studied topic.

Chapter Three: This chapter describes the approaches and techniques employed in this study to design the design's research methodology. This chapter also covers data collection, analysis, preparation methods, and metrics used for performance evaluation.

Chapter Four: It contains the overall design of the model, proposed architecture, data preprocessing phase, word embedding, the hybrid model learning approach, and the evaluation process we used to determine the effectiveness of our solution.

Chapter Five: The experimentation and implementation of the approach are explained in this chapter. This includes defining the working implementation environment, describing the dataset, implementation data preprocessing, model construction, and implementation of performance evaluation.

Chapter Six: This chapter describes the results and discussion. Explain the outcome of the proposed solution to prior research findings. The results of data collection, data preprocessing, model evaluation, and the study's results are described. The Discussion and Outcomes chapter additionally explains the effectiveness of the proposed model of this conducted research.

Chapter Seven: Finally, in the end, we summarize the overall research findings, draw conclusions, and make recommendations for new researchers in the area of fake review identification. Finally, give a complete investigation in this chapter and provide some beneficial insight into the recommendations and valuable suggestions.

CHAPTER TWO

2. LITERATURE REVIEWS

2.1. Introduction

To gain a better understanding of review detection, the study discusses and reviews a selected set of recent relevant books, journals, and articles in this body of literature. Examines and compares existing work on fake review detection to identify attempts, contributions, limitations, gaps, and some related works for improving review detection model performance further. The objectives are to identify sub-problems previously attempted in the field of fake review detection, methods used, important contributions from different researchers, comparative study of designed models and algorithms, improvements, and gaps in various research papers.

2.1.1. Fake Review Overview

Fake reviews are posted to deceive consumers into making a purchase decision on products by the fake reviewer and are often written by inexperienced reviewers (Martínez Otero, 2021). The purpose of fake reviewers is to mislead the new or regular customers of the company with false reviews that intend to harm or promote the target product instead of giving their own opinion. We can use the concept of fake reviews to distinguish them from other suspicious business activities, such as influencer marketing, sponsored content, and selective targeting. People who recommend the product in the online business are people who have a large follower on e-commerce platforms and are well known for their online activities.

Most e-commerce products' purchase decisions are influenced significantly by online reviews, and the conversion rates increase as soon as products begin to receive reviews (Team, 2017). Fake reviews, on the other hand, pose as anonymous consumers who have purchased the products and express their comments about them in an unbiased manner. Unlike fake reviews, the promotional message in sponsored content is embedded in non-commercial content like a report, opinion piece, social media post, or blog post. Influencer marketing and sponsored content must disclose their commercial origins to be legal. They become illegal business practices with these features if they don't have common fake reviews.

Fake reviews may be against the law in terms of consumer protection, but they are incredibly attractive for companies to create more customers and businesses. Customers may leave a review

of a product provided by a company based on a particularly positive or negative experience with it. Most online reviews are really helpful for customers in making decisions, relying on these recommendations is very dangerous for both the seller and the buyer. Companies that pay for fake reviews want to hide their spots as much as possible by making the reviews look as genuine as possible (Brechon, 2014). Before making a new purchase online, most people would try to inspect each user's review. However, reviews may be fabricated for some illegal forms of profit, so any decision based on internet reviews should be taken with caution (He et al., 2020).

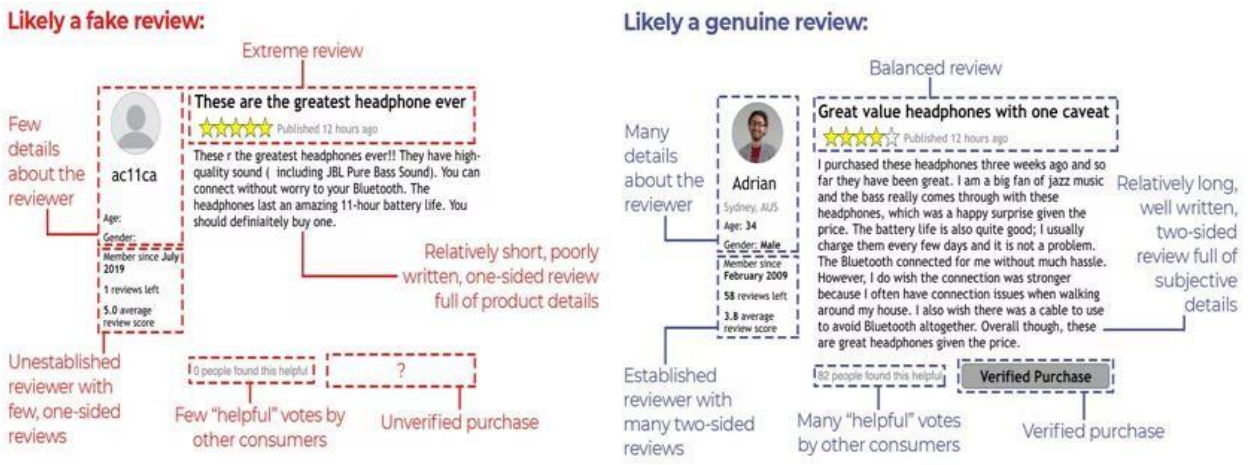

Figure 2-1: Example of fake review and genuine review[3]

Fake review brokers try to make money by misleading unaware customers and giving themselves an unfair advantage over their rivals, which is bad for selling partners. Fake review identification has no common method to classify reviews as fake or genuine, but there are some features used to say whether a review is fake or genuine, such as extreme review, poor writing, voted by other reviewers, contextual meaning, review rating, the similarity of the contents, generic tone, and review length. Fake reviews use the rating systems available on digital platforms to improve or decrease a product's market position to increase or decrease sales. They also help with products that have few technical features, because potential buyers trust the opinions of other users more than products that stand out for their technological characteristics. There are two main categories of fake spammers, such as individual and group reviewers (Mukherjee et al., 2012).

[3] https://theconversation.com/how-to-spot-a-fake-review-youre-probably-worse-at-it-than-you-realise-121043

Figure 2-2: Fake review example with overlapping review contents adopted (Lau et al., 2011)

A. Individual Fake Reviewers

An individual spammer is a reviewer who does not work with anyone to promote a product to mislead customers or demote a competitor's sales company products. A single person writes negative or positive reviews for the products (Cao et al., 2021). He/she just creates false content using one user ID account that may be using a well-known account.

B. Group Fake Reviewers

A group of reviewers is a collection of spammers, each with their user id, who may work together unknowingly or knowingly to promote or degrade certain e-commerce products. It leads to the propagation of fake content reviews in real-time, which may be used to attack the company's products using negative reviews or write positive reviews to build and promote services using false reviews (Cao et al., 2021). Most scammers are run by professional reviewers or companies that write fake reviews and can be deliberately unscrupulous. In an e-commerce business, there are typically two types of group spammers who post fake reviews. The first category of spammers could be a single person who creates many accounts or user ids and writes multiple fake reviews to promote or downgrade products. The second type of spammer is a group of people who collaborate to help or hurt a target entity's reputation (Hussain et al., 2021). The group's spammers may or may not be aware of each other's activities.

2.1.2. Domain of Fake Reviews

Various review domains are available in the modern world to accept the most relevant comments via online mechanisms. The creation of fake positive customer reviews for products they want to promote and fake negative user reviews for products they wish to criticize is a strategy used by marketers, advertisers, and other competitors to attract customers. An actor writing a fake review may set up a user profile using information from a marketing campaign and then submit a review while posing as the person they claim to be. This is a misapplication of the review system, which generally only allows genuine users to leave reviews rather than sponsored shills. A real user may provide a fictitious review of a product or service they have not used. The most common domains of online reviews, such as restaurants, hotels, healthcare, and medical services, insurance services, clothing stores, beauty salons, grocery stores, and auto services (Ott et al., 2013) (J. Li et al., 2014) (H. Li et al., 2014) (Rayana & Akoglu, 2015) (Mohawesh, Xu, et al., 2021b) (Anusha P, 2020).

2.1.3. Review Dataset labeling methods

Different datasets were used in previous research to get good performance in fake review detection. There are four basic construction methods to collect data from the source(Mohawesh, Xu, et al., 2021a).

Human methods: construct datasets using rules that labeled the data by creating the rules to classify data as fake or genuine reviews. This method needs a lot of human power and time to label the review datasets, plus they may have inaccurate labels because the decision is given by a vote from the group.

Amazon Mechanical Turk (AMT) - This data collection method collects review datasets by different educated anomalies using web scraping methods. In the end, it pays for that collected data for anomalies.

Filtering algorithm methods: This algorithm is one of the best filtering algorithms and is used to label reviews as fake or guanine, but it's not open source for the public to access the algorithm. The algorithm uses the behavioral and lexical features of data to categorize reviews into classes of reviews.

Rule-based methods: a rule to annotate the dataset, so to say fake or genuine depends on the attributes of the review dataset. This method is less accurate but less expensive, and it only takes a short time to determine whether data is fake or real based on the rules.

2.2. Fake Review Detection

Fake review detection research primarily focuses on strategies for keeping e-commerce review sites secure enough to prevent the propagation of fake reviews that could mislead customers. When consumers read a lot of reviews, it's hard to tell the difference between fake and accurate reviews. The major challenge with fake review detection is identifying the review from a large number of reviews given by different customers or false users as fake or genuine. The majority of existing research has focused on textual features, and some studies also focus on behavioral attributes, while some techniques have considered social and temporal factors (Mohawesh, Xu, et al., 2021a).

The fake review detection research based on reviewers can be divided into three categories: individual spammer detection, group spammer detection, and cross-domain spammer detection (Barbado et al., 2019). The aim is also to create a technique that is capable of accurately identifying fake reviews while integrating all relevant information about the opinions and users. The neural network models take the reviews and the reviewer's features as input values and produce a label for each review that specifies whether it is genuine or fake as an output. Finding a fake review detection analyzes the features and sentiments attributes with the characteristics of reviews by considering similarities between the two reviews (Mohawesh, Xu, et al., 2021a) (Anusha P, 2020) (Paul & Nikolaev, 2021). Currently, the fake review detection techniques have three basic approaches depending on the feature used to identify the review (Jindal & Liu, 2007). Let us discuss the centric approaches to fake review detection one by one.

Review Centric Approach: an approach that identifies review content as a genuine or fake review based on the content of the review written by the reviewer. Multiple attributes are used to differentiate between reviews, including review content consistency, use of all capital words, number usage, brand name, the similarity of items and reviews, and repeated use of both positive and negative phrases in the review. This approach contains different attributes such as verified review, the length of review such as the number of words contain, the incentive review, sample review, and sentiment analysis of text or rating.

Reviewer Centric Approach: these are methods that depend on reviewer behavior. This approach takes into account user information as well as all of their evaluations. Additional features to consider in this method include account age, profile picture, URL link, logical address, number of reviews published by one reviewer, daily status description, and account age.

Product-Centric Approach: is mainly focuses on product-related information. According to this strategy, features include the total number of reviews posted, the product's sales rank, the actual ratings, the average review length, the price of the products, the total number of brand repeat customers, and the total number of brand admirers.

2.3. Application of Fake Review Detection

Various fake review detection methods are used to classify a reviewer's opinion to challenge the acceptance of the product both in the negative and positive impacts. To overcome the fake review problem, multiple machine learning techniques such as Naive Bayes, SVM, RF, Decision Tree, Ensemble learning, and deep learning such as CNN, simple RNN, and LSTM have been used to identify fake and genuine reviews in different application areas such as Hotel, Healthcare, Clothing Store, Grocery Store, Auto Service, Hotel, Retailers, Insurance Service, and Car Dealerships (Barbado et al., 2019)(Gutierrez-Espinoza et al., 2020)(Sedighi et al., 2019)(Mohawesh, Xu, et al., 2021a). The previous study's methods analyze social media comments, capture semantics, and extract the feature that is relevant to the data which are then assigned into two categories genuine and fake reviews.

2.3.1. Machine learning for fake review detection

The various studies conducted on the identification of fake reviews using machine learning and deep learning to identify the reviews based on textual and behavioral characteristics (G. ; S. R. Patil, 2021)(Et. al., 2021)(Barbado et al., 2019)(Gutierrez-Espinoza et al., 2020). A learning model of consumer online reviews based on the framework of model and trustworthiness was produced by adjusting the ratio of real to false reviews, which established the relationship between fake reviews and customer legitimacy.

Machine learning approaches can recognize the difference between fake and real review content by detecting hidden knowledge in the text that the human eye cannot detect. Because spammers attempt to explain a product using a contextual feature and sentimental analysis when they write fake reviews (Sun et al., 2016). Fake review detection can be viewed as a classification problem with two categories, previous studies on the review using machine learning models such as supervised semi-supervised, and unsupervised, can be used to classify reviews as fake and genuine (Jindal & Liu, 2007)(Mohawesh, Xu, et al., 2021a).

Supervised machine learning:[4] algorithm uses labeled training data to learn how to forecast results for unlabeled data. The algorithms can predict future events by applying previous knowledge to new data using labeled examples. The model can then give targets for any additional input after it has been trained. The model can be accurately updated by comparing the results with the expected results and using the learning method to identify errors. Using a labeled dataset, a supervised learning model is used to predict whether a review content is fake or guanine (Cardoso et al., 2018a)(L. Li et al., 201 7)(Khurshid et al., 2017)(Ott et al., 2011)

Semi-supervised machine learning[5]: are combines both supervised and unsupervised learning techniques. Semi-supervised techniques are used to add unlabeled data to the training set when the amount of labeled data needed to develop an optimal solution is insufficient, but there are not enough resources to gather more labeled data. Due to the difficulty in obtaining labeled reviews, fake review detection can use semi-supervised techniques. Machine learning can detect fake reviews using unlabeled data (Hai et al., 2016)(Deng et al., 2018).

Unsupervised machine learning[6]: this is a kind of algorithm that learns patterns from unlabeled datasets. It's a machine learning technique that allows you to find all kinds of unknown patterns in data without having to supervise the model. Algorithms are utilized in situations when the training data is neither labeled nor classed. Instead of choosing the best result, the system analyzes the data and employs data to identify hidden patterns from the training dataset (Noekhah et al., 2020). The fact that the model catches these product-related review attributes is beneficial. Previously there are different techniques used for fake review detection such as Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB), K-nearest neighbor (KNN), Logistic Regression (LR), and Random Forest (RF).

2.3.2. Deep learning for fake review detection

There are various study are done on fake review detection around the different domains. Many scholars use linguistic and behavioral information to identify false positives using in-depthlearning algorithms that have excelled in other fields. Deep learning methodologies can address issues in feature engineering that cannot be addressed by more conventional machine learning techniques. The neural network is utilized for feature extraction, which decreases the time,

[4] https://ai-ml-analytics.com/machine-learning-algorithm/
[5] https://ai-ml-analytics.com/tag/supervised-learning/
[6] https://ai-ml-analytics.com/tag/unsupervised-learning/

production costs, and demand for technical skills that artificial feature creation requires. Most text classification research now focuses on building and optimizing neural networks with datasets from hotels and restaurants (Hajek et al., 2020)(Cui et al., 2018)(Abdolrasol et al.,2021)(Bhuvaneshwari et al., 2021).

A. Convolutional Neural Network

[78]CNN is a deep learning technique of feed-forward neural network used to detect and classify text. Train a model that is adequate for identification and classification by using forward propagation and backpropagation to identify a collection of the most appropriate information transferring connections from input to output in a large parameter set. The CNN model extracts local characteristics from the input matrix by changing the convolution kernel and then combines the local data in a higher layer to produce statistical information. A convolutional neural network is a mathematical construction that includes layers such as convolutional, pooling, and fully connected.

[9]

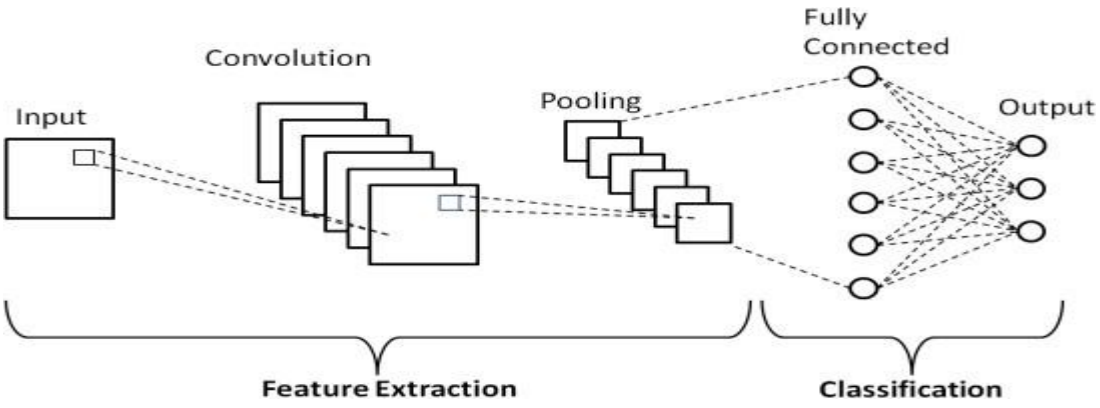

Figure 2-3: Architectures of Convolution Neural Network

The architecture does not demand professional human feature extraction and tumor segmentation by hand (G. ; S. R. Patil, 2021). To recognize misleading spam comments, trains document representation using convolutional neural networks. The word vector was used as an input for train and testing in the proposed model (X. Wang et al., 2017)(Zhao et al., 2018)(L. You et al., 2020)(A. Patil & Rane, 2021).

[7] https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697

B. Recurrent Neural Network

Recurrent neural networks are constructed from several connected neural networks. They are commonly used for natural language processing in sentimental analysis and text mining. It is a link between a recurrent node in a sequence and networks with loops that assists in information persistence. In text classification, RNNs process words in a phrase recurrently and sequentially, mapping a complex and low-dimensional representation of the data into a low-dimensional vector (Nasir et al., 2021). RNNs have the potential to enhance complexity and evaluate word by word, maintaining the context of texts as one of its properties. The input vector is fed into the RNN cells in the next hidden layer. Each input word has the same weight in the RNN, and the parameters are shared between different parts. RNN can only run for a limited number of steps because of problems with vanishing gradients. As a result, new models such as Long Short-Term Memory (LSTM)[10], Gated Recurrent Unit (GRU)[11], Bidirectional LSTM[12], Stacked LSTM[13], and LSTM with attention method have been developed to overcome the limitations of RNN. Bidirectional LSTMs train two LSTMs rather than one in the input pattern when all of the appropriate actions of the training data are available (Dhamani et al., 2019)(W. Liu et al., 2020). To get around the RNN limitation, researchers used a bidirectional LSTM model to learn the document-level representation of reviews and detect fake reviews using a combination of features (Dong et al., 2018)(Jain et al., 2019)(C. C. Wang et al., 2018).

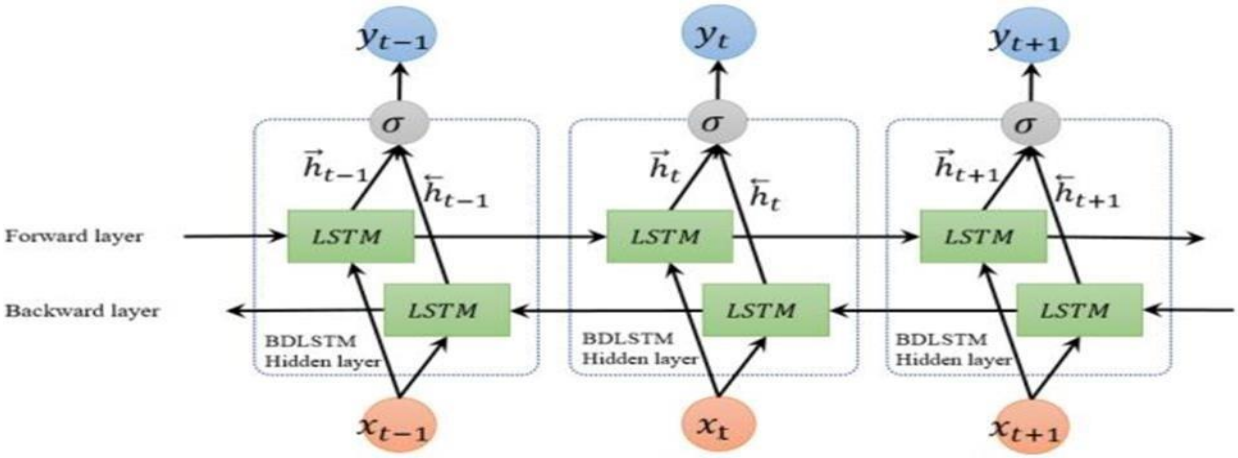

Figure 2-4: Bidirectional recurrent neural networks architecture adopted from (Cui et al., 2018)

[10] https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory
[11] https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be
[12] https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/
[13] https://machinelearningmastery.com/stacked-long-short-term-memory-networks/

2.4. Text Representation

Data preprocessing in computer applications is one of the main things for extracting knowledge, analyzing data, and visualizing it. A technique that decreases the number of features in datasets to more manageable groups of processes while also reducing the quantity of redundant data. It is a method of appropriately picking or combining variables into features to get good performance and avoid the curse of dimensionality in NLP text classification challenges. Most research publications apply text classification algorithms to the specific challenge of fake review detection. However, other research employs specific review detection features to address the problem (Nigam et al., 2017). There are various ways for detecting fake reviews of text representation that are currently used such as.

2.4.1. Term Frequency-Inverse Document Frequency (TF-IDF)

A weighted strategy for calculating a value for word-to-document relationships is widely used in machine learning text classification (Ramos, 2003). Higher log TF-IDF weights with words are deemed more representative and are retained, whereas lower weights with words are considered less representative and are deleted. Measure TF-IDF[14] results using the following formula:

$$\mathbf{tf}(t, d) = \frac{f_d(t)}{\max_{w \in d} f_d(w)}$$

$$\mathbf{idf}(t, D) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

$$\mathbf{tfidf}(t, d, D) = \mathbf{tf}(t, d) \cdot \mathbf{idf}(t, D)$$

$$\mathbf{tfidf}'(t, d, D) = \frac{\mathbf{idf}(t, D)}{|D|} + \mathbf{tfidf}(t, d, D)$$

$f_d(t) :=$ frequency of term t in document d

$D :=$ corpus of documents

TF = the number of occurrences of a term in a document divided by the total number of words in the document.

IDF = log (the number of docs) divided by (number of docs in which the word appears)

A well-chosen feature set of words can help to accelerate information retrieval while maintaining the efficiency of the algorithm. For many publications, however, the threshold values or TF-IDF criteria for identifying significant phrases are solely based on experience. Various study use TF-IDF to extract features to get the characteristics of review content fake or genuine review (Cardoso et al., 2018b)(Khurshid et al., 2018)(Mohawesh, Tran, et al., 2021b).

[14] https://www.r-bloggers.com/2014/02/the-tf-idf-statistic-for-keyword-extraction/

2.4.2. N-grams

The N-gram is used to determine the next word in a text after looking at the main arguments for generating characteristics and algorithms for trained machine learning models (Cardoso et al., 2018b). All potential word or character combinations that could be n-gram together are known as n-grams. Compared to BOW, it is better suited to enhancing classifier performance because it partially takes into account each word's context (B. Li et al., 2017).

2.4.3. Bag-of-words (BOW)

Commonly used for text representation a way to extract attributes from given data for use in machine learning algorithms to classify text by extracting features (V M & Kumar R, 2019). The method is simple, adaptable, and used to extract attributes from data sets in different ways. These methods have both known forms, such as a dictionary of known terms and an estimate of the existence of known terms in a data set. The technique was only concerned with whether or not recognized terms were found in the document, not with where they were found in the document.

2.4.4. Word embedding

One of the most popular and crucial techniques for extracting text data features is word embedding. In NLP, it is common practice to use pre-trained word vector representations, also known as word embedding, for several downstream tasks. Although more information needs to be learned, large amounts of embedding can lead to more accurate connections between words. The goal of Word Embedding is to improve the ability of networks to learn from text data and converts words into numerical representations. Today, natural language processing has been made possible by including word embedding into a neural network model. The embedding layer, which maintains a lookup table to convert words represented by numeric indexes to their dense vector representations, frequently controls the deep learning component of the TensorFlow and Kera's frameworks (Popov et al., 2018).

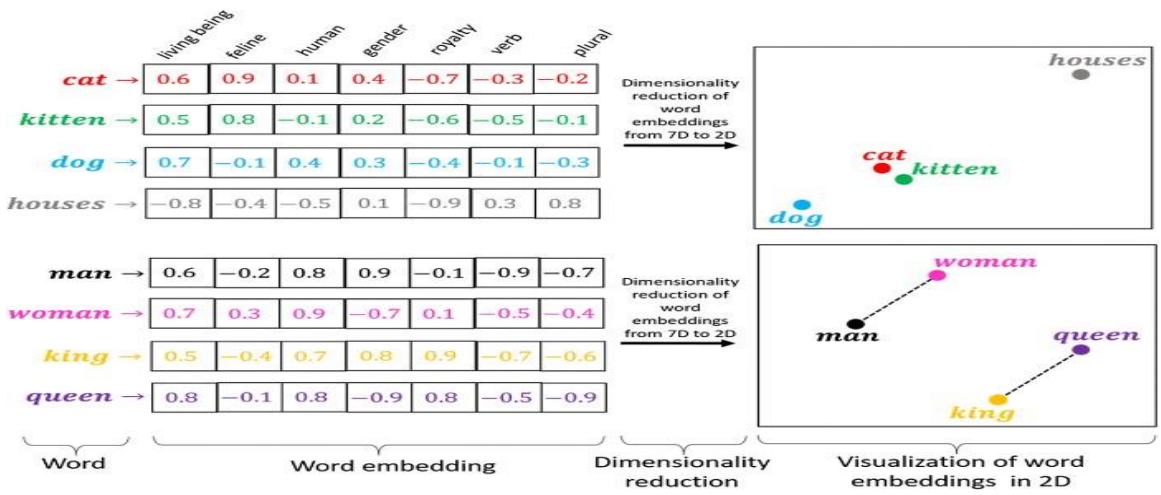

Figure 2-5: Word Embedding diagram adopted from (Gautam, 2020).

We require word embedding for converting text into numbers, to represent the text into vector representation of a word that encoded one-shot vector. It can detect the context of a word in a dataset, as well as syntactic and semantic similarity, and relationships with other terms. Word embedding differs from the traditional vector space method, in which each word can represent a fixed-length vector representing the vocabulary of the documents in the corpus (Zou et al., 2013)(Delbeto, 2018).

Word2vec is a technique for extracting word embedding features developed by Tomas Mikolov in 2013 at Google that employs a shallow neural network to train word embedding (Mikolov et al., 2013). It is commonly used to train word embedding from raw text. The notion of word2vec evolved from the concept of distributed representation of words; it employs a shallow neural network to train word embedding and predict the relationship between each word and its context words to words occurring in similar circumstances (L. Li et al., 2015)(X. Wang et al., 2017)(Zhao et al., 2018)(Z. You et al., 2018). Word2vec employs two methods: the Continuous Bag of Words (CBOW) model, which predicts the current word from a window of surrounding context words, with the order of the context words having no bearing on prediction. Second, the Skip Gram works on adjacent context words, which are weighted more heavily than context words further away.

Fast Text is an open-source, lightweight library that uses sentences as a bag of words and it takes a longer time to train a classification of text developed by Facebook. One of the word2vec model extensions represents each word as an n-gram of characters instead of learning vectors for words

directly. The challenges of anticipating the meaning of phrases using skip-gram can be examined with a categorization assignment. As a result, the rapid text model takes into consideration the true meaning of words (Joulin et al., 2017)(Almeida & Xexéo, 2019).

Glove is called global vector works by understanding the built-on global matrix factorization and local context window that learns to embed words based on word frequency through the matrix of co-occurrence and weight loss. Its main purpose is to create word vectors that capture meaning in vector space while utilizing global count statistics rather than just local data (Almeida & Xexéo, 2019).

2.5. Related Works of the Study

2.5.1. Related Works

The majority of researchers have used a variety of strategies to detect fake online reviews. These methods have resulted in significant improvements in the detection of fake and genuine reviews. The approaches are divided into two categories: content-based and user-behavior studies. Various researchers have conducted studies on the detection of fake reviews (Mohawesh, Xu, et al., 2021a). We choose a paper that is relevant to our research to visualize the gap and performance of the current method for detecting fake reviews.

(Barbado et al., 2019) Authors have investigated the detection of fake reviews on online consumer electronics retailers' websites. The article proposed a feature framework for detecting fake reviews that have been tested in the consumer electronics domain. This paper uses web scraping to collect Yelp business center datasets from the four most populated cities in the United States of America, and it investigates a machine learning model for detecting fraudulent reviews in the consumer electronics industry. They introduce a fake feature framework (F3) for organizing the extraction and characterization of features in fake detection. The article uses a model that incorporates review-centric and user-centric features. The feature framework model is oriented to the analysis of social sites constituted by the future extraction of fake reviews. And the findings of the experiment are evaluated by 10-fold cross-validation.

(W. Zhang et al., 2018) They use a recursive convolutional vector approach to represent each word in a six-component review. The first and second components are two numerical vectors of words derived from the training of both deceptive and truthful evaluations. The third and fourth steps of the study were to train the RCNN techniques model with the context vector of the content text and

word vectors from the nearest phrase as genuine and fake reviews from left-neighbor contextual factors are generated. The fifth and sixth components are accurate neighbor honest and misleading context vectors of correct words. Furthermore, they use max-pooling and Rectified Linear Unit filters to convert recurring convolutional vectors of words in a revision to a revision vector by selecting the most positive feature components in the recurring convolutional vectors of words in the revision.

(Bhuvaneshwari et al., 2021) In this study, they used a novel deep learning model based on self-attention to learn how to identify document level, performance, and the fake reviews model. The objective is to classify based on syntactic contextual information and they used datasets such as MTurk, and TripAdvisor datasets in the online restaurant. They use the TripAdvisor datasets for different splits such as training, validation, and testing that demonstrate the representation of the model. The study compared the performance of CNNB, ACNN, CNN, HSL, and CS using precision F1-score, and recall evaluation methods. They use different results that the optimizer for all variants with a learning rate of 0.0001 and a decay of 10^{-6} for the RMSprop.

(Gutierrez-Espinoza et al., 2020) This study used restaurant datasets and ensemble learning-based algorithms to identify fake and genuine reviews. They look at the performance of ensemble-based approaches and compare them to that of traditional machine learning algorithms. Ensemble learning-based approaches are incorporated into traditional Support Vector Machines and extreme gradient-boosting machines, and their performance is compared to that of traditional algorithms. The comparison of the methods also uses machine learning approaches and deep learning for fake review detection to get good results. The research solution shows the MLP techniques with Doc2Vec embedding matrix model is more performance accuracy in case of single classifiers documents. The ensemble of classifiers outperforms the underlying classifier, whether it isRandom Forest and XGBT with Decision Tree, or the bagging ensembles with their correspondingSVMs or MLPs.

(Hajek et al., 2020) In this study, the DFFNN model is evaluated for the effect of different emotion, word, and sentence representations on the performance of fake review detection. They used four benchmark datasets such as the hotel, doctor, restaurant, and Amazon datasets. The study used the integrated framework of Skip-Gram, n-gram, and emotion models. The model did not fully use the

reviewer- and product-based features to increase the accuracy of fake review detection to classify as fake or genuine.

(J. Wang et al., 2020) this study develops a model by rolling collaborative training and multiple feature fusion. The researcher use the yelp shopping website to review the training and testing datasets. They used unlabeled data in the study to increase ranking performance and fake review accuracy. The model detects a fake review by combining multi-feature fusion, rolling collaborative training, and labeled manually. To get good results, they used consistency of the sentiment and examined the results. The text representation model is used to draw out the aspects of the review, which are then combined with the external features of the text to increase the classification model. The study was put as a reference accuracy increased by 5.3 percent as compared to the prior deep learning model. The top classifiers are the support vector machine and random forest.

(Sun et al., 2016) In this study, they used datasets from the amazon review dataset to develop the fake review detection model. By integrating a product features review with a product word creation, the researcher uses a convolutional neural network model. The experimental data sets were divided into a ratio of 80:10:10 sections into training, validation, and test sets, and then tokenized using NLTK and the sentences were separated. The models improve performance primarily by combining composition information with a product attribute. They combined the CNN model with the SVM classification model to reduce high variance and overfitting while capturing product-related features and classifying the reviews.

2.5.2. Summary of the Related Work

Table 1: A summary of Related works on fake review detection

Title of Papers	Dataset (Data Collection)	Algorithm	Performance Evaluation in (Accuracy)	Features Extraction	Research Gaps of Related Papers
(Barbado et al., 2019)	Yelp Datasets	RF, Ada Boost	82%	TF-IDF, Word2vec	• Few attribute that specific to the product's feature they used. • The performance and evaluation for model are low.
(W. Zhang et al., 2018)	AMT, Deceptive Datasets	RCNN and word contexts	80.8%	Skip-gram	• They use only the contextual features and ignored behavioral features from the model. • Only focus on the semantics and statically structure.
(Bhuvaneshwari et al., 2021)	Yelp Zip, MTurk, TripAdvisor Datasets	CNN, and Bi-LSTM	86%	Word2vec	• There is no rating deviation for feature selection. • There is used the textual feature are used to identify the review class.
(Gutierrez-Espinoza et al., 2020)	Restaurant Dataset	EL, MLP, DT, RF, SVMs, XGBT.	77.3%	Doc2Vec	• Limited datasets to train and test machines. • Less accuracy and only analyzing static texts.

(Hajek et al., 2020)	Hotel, Restaurant, Doctor Datasets.	DFFNN, CNN	82.80%	Skip-Gram	• The domain is also not specific to get more performance with datasets. • The focus of the contextual reviews ignores the behavioral features.
(J. Wang et al., 2020)	Yelp CHI Dataset	SVM and RF	81.89%	Doc2vec	• Not end-to-end detection properly to identify the fake reviews.
(Sun et al., 2016)	Amazon Dataset	CNN, SVM	77.2%	Traditional Feature Engineering	• The reviewer feature is limited to the contextual feature of reviews. • The similarity of the words in the reviews is not considered.
(Et. al., 2021)	Amazon	Decision Tree	65.4%	Traditional Feature Engineering	• The training datasets are small. • Few feature selection was used and ignored the behavioral attributes.
(C. C. Wang et al., 2018)	Mobli10.com	LSTM	89.4%	Dictionary	• Does not compare the performance with other neural networks. • They focus on contextual features and do not consider the metadata of reviews.

2.6. Gap of Research

The study of online reviews includes a wide range of areas, with the main goal of detecting fake reviews using various methodologies. Fake reviews are typically associated with email spam and are mostly utilized for marketing purposes. Through the manipulation of the page's content, online spam attempts to draw users by raising the page's search engine ranking. Spam reviews, on the other hand, are distinct in that they convey a negative assessment of a good or service, and it can be quite challenging to identify them manually. As a result, spam review detection cannot be accomplished with current web spam or email spam detection methods. There is an increasing number of customers who are concerned about being deceived or caught up in these fake reviews when shopping online, and identifying fraudulent reviews has become a very popular topic in the field of research. Furthermore, researchers claim that humans are incapable of distinguishing between deceptive and truthful reviews to a satisfactory degree.

Various researchers in the online review have conducted related works with promising results using deep learning and machine-learning techniques in various sectors such as hotels, restaurants, movies, and entertainment. The existing work has some gaps such as:

- The previous study focus on the textual feature and ignores the context of words in a sentence extracting the traditional linguistic units from texts, such as individual words and context-free grammar rules and classify them (Anusha P, 2020).
- There are a limited number of features, and the methodology used for detecting fake reviews has low-performance accuracy. According to the survey (Mohawesh, Xu, et al., 2021a) the previous studies focused on the content of the reviews, and the other researchers also worked on the behavioral feature instead of the review content.
- The performance of the previous studies for fake review detection in electronics consumers is quite low (Pendyala, 2019) due to the small number of datasets used for building the model.
- Most research works on unrepresentative real-world reviews that are generated using machine learning to build models and the review dataset covers a small portion of users (Z. You et al., 2018).
- The review dataset class is imbalanced, with more real data covering the genuine class than the fake class. The disproportional representation of each class in the dataset tends to

bias the model. Deep learning models work under the premise that class values are distributed evenly in the dataset (Buda, 2017).

The number of studies focusing on the analysis and study of fake review detection has increased, but there is still plenty of room for research in this area because there is no unified solution yet. Due to those difficulties, detecting fake reviews is an open task, and those issues still require improvements and ongoing research areas. This paper proposed fake review detection using hybrid neural network approaches in the case of e-commerce retailers, which introduces constraints.

CHAPTER THREE

3. RESEARCH METHODOLOGY AND PROPOSED MODEL

3.1. Introduction

This chapter describes techniques, methods, and approaches intended to achieve the goals of identifying and classifying fake reviews using hybrid deep neural networks. This includes how data preprocessing techniques and features engineering of fake review detection, development tools (software, hardware, and design), training model selection, and evaluation metrics are used. In addition, research procedures, practices, and systematic approaches are defined. The proposed hybrid neural network model uses product-related review features and provides more robust prediction results. The research design used in this study is scientific research and focuses primarily on experimental scientific research.

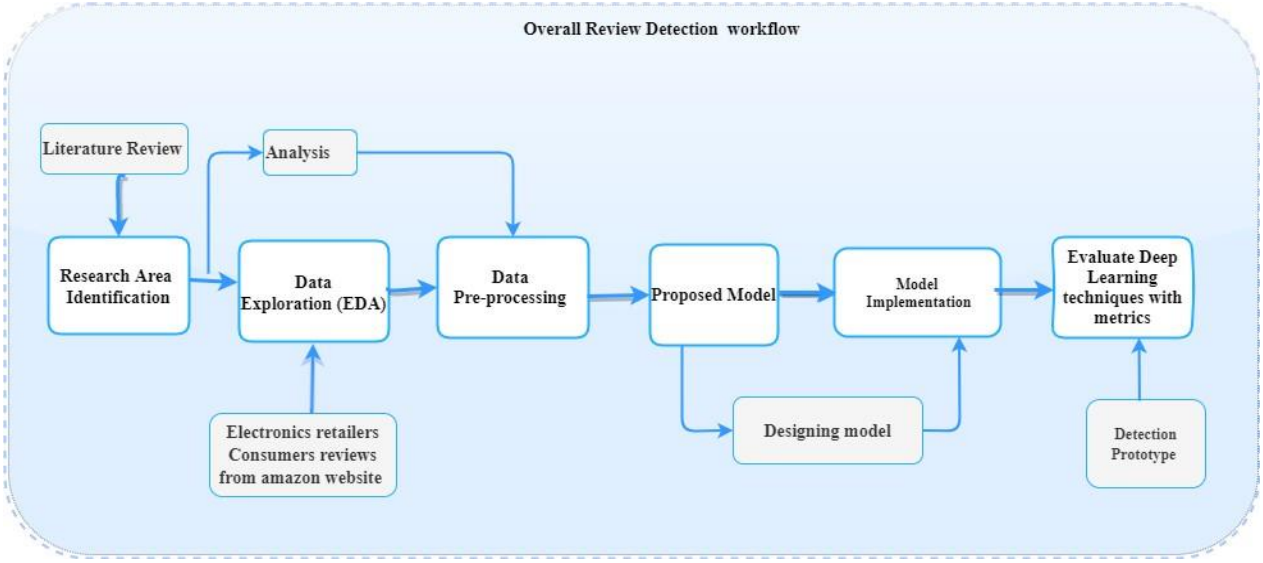

Figure 3-1: The research methodology workflow diagram

3.1.1. Dataset Description

The review product datasets are obtained from the reviewer contributions column of the profiles dataset. Our experimental dataset, which contained reviews from the Amazon e-commerce platform, was provided by the Amazon electronics review datasets. For experimentation, we use data from the 5-core subset, which includes all consumers and items with at least five reviews. The review product dataset is derived from the profiles dataset's reviewer contributions column. We used an Amazon electronics products review dataset and the experiment contains the ID of the

reviewer, product IDs, rating, review texts, review feedback, and review time. The descriptions and features of the e-commerce datasets are listed in APPENDIX B.

The full dataset is available at this link: http://jmcauley.ucsd.edu/data/amazon/ electronics reviews contain 1,689,188 reviews in JSON file format. The data is in a list of dictionaries and requires data manipulation to transform the data into a structured table. The transformation of a character into a single common representation does not affect its meaning. The columns in the original tab-separate value file were reduced to only the statement and label for the CSV file. There are various features in the review, extract 11 columns from a list of dictionaries as a result and show the sample on APPENDIX A. In addition, we used an Amazon electronics dataset collected by the hugging face[15] data repository, which is open to the online community.

The main challenges in the field of electronics are the lack of real datasets and labeled class datasets, which are important to detect the review understanding. The main reason for selecting this dataset:

- The datasets contain a large number of reviews with 5 scores and from verified amazon accounts within different attributes of datasets such as useful in achieving good performance.
- The dataset used to train and test our model is labeled using 13 different filtering frameworks to classify the reviews (Hussain et al., 2020).
- The main advantage of choosing these datasets is that they are grouped for review in the electronics domain.
- Additionally, in the field of electronic reviews, there are no well-labeled datasets containing reviews within metadata.
- There is not enough real application review dataset in electronics fields that is collected from the e-commerce platform to identify the experience of customers with the products.
- The data contains textual and behavioral features that help to build a model of fake review detection.

The data for this study were chosen from a comprehensive dataset based on the number of reviews for the products provided by reviewers. To reduce the dataset and utilize it effectively, we choose

[15] https://huggingface.co/datasets/amazon_us_reviews

products that have more than 10 reviews. To determine the ideal ratio for this model, weexperiment with the ratio's performance and review earlier work. Because there is no standard fordividing the ratio, different studies employ the d/t percentage. For building the model and testing performance, we divided the dataset into three sections as follows:

- **Training dataset:** used to train the model and hold a large part of the data
- **Validation dataset:** to validate the trained model that may not be seen with the training dataset.
- **Test dataset:** to test the model and check the performance of the model.

3.2. Dataset Preprocessing Techniques

Data Preprocessing is a method of converting data into a machine-readable format, and it is an important step in machine learning because the quality of the data directly affects the learning ability of the model (Goyal, 2021). We cannot work with raw data because it have contain missing values, noise, irrelevant data, and inconsistent data that cannot be directly used at an important stage in model building. Before using data sets with deep learning, the data quality must be validated, the data cleaned, and the input data must be appropriately formatted. That makes it suitable for our model and a look at some interesting text techniques. To train and test the proposed model, the datasets must be properly preprocessed to achieve good accuracy and efficiency in our model.

Data cleaning: We perform the cleaning of the review datasets, such as cleaning the whitespace, symbols, irrelevant characters, punctuations, URLs, and emoji's. Remove the elongation from the review contents and normalize the review text. We used data cleaning to improve the quality of our dataset, which improves the accuracy of our proposed method.

Handling missing values: Missing values in a dataset can be handled in a variety of ways, which include replacing missing values with their mean or removing the missing values if they are unimportant. To fill the average value of the observed data, the mean imputation technique is preferable to our preprocessing phase of the review dataset.

Tokenization: In this work, the text review was broken down into word-level tokens. There are various types of tokenization from which regular tokenization is used to extract the token from the

text by utilizing the regular expression. We use genism to tokenize the content of the review in the preprocessing and the textual content breaks the sentence by using space between words.

Stop words removal: Stop words are frequently removed from the text before training deep learning and machine learning models because they occur in abundance, providing little to no unique information that can be used for classification (Brownlee, 2017). The words that occur commonly across all the documents in the corpus of review contents are removed using the genism preprocessing library. Remove stop words to enhance the performance of our model because there are fewer and more meaningful tokens remaining.

Lemmatization: is the reduction of a review text that is tokenized into its word stem, which is removing the attached suffixes, and prefixes into the roots of words known as a lemma, and is different from stemming by taking into consideration the vocabulary. It's commonly employed in tagging systems, indexing, Web search results, and information retrieval. In this research, we use WordNet to lemmatize the tokenized content of the review by importing the WordNetLemmatizer from the NLTK library.

Handling imbalanced dataset- The review dataset had an imbalance class problem, which was resolved by changing the hyperparameter for the algorithm to balance. However, this may lead to the loss of essential information. We tried to test the oversampling and undersampling random methods to choose the best-balanced datasets. From the dataset and performance, we chose undersampling random methods. The synthesized data, and even the conclusions of an unbalanced classification, are unstable due to this randomness.

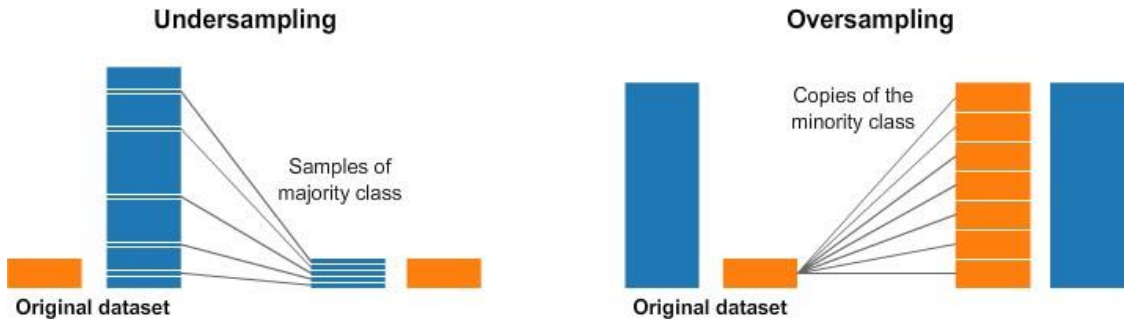

Figure 3-2: Dataset Balancing using Undersampleing and Oversampling[16]

[16] https://www.kdnuggets.com/2020/01/5-most-useful-techniques-handle-imbalanced-datasets.html

Feature Scaling: The technique of feature scaling allows us to normalize our data to important standards. There are several things we should do to prepare our review datasets for learning before training a deep neural network. By conducting some sort of feature scaling on the review dataset, we can normalize it and significantly enhance the performance of our hybrid deep neural network for the fake review detection system. Without normalizing the dataset deep learning techniques is not effective because the values range is high so it need feature scaling. We use standardization scaler methods to scale our models in this study because we are using the deep neural network algorithm, which requires feature scaling. By working on the linear transformation of data, the standardization scaler maintains an existing relationship among data.

$$Xstand = \frac{(X-\mu)}{(\sigma)}$$

Equation 3.1: Feature Scaling

Xstand is new transformed features, **X** is the normal value of a feature, **μ** is the mean value, and **σ** is the standard deviation value of the review dataset.

3.2.1. Features Extraction of Fake Review Detection

This section is going to describe the extraction technique required for reviews that are collected from amazon sources of data to identify the contents of reviews using feature extraction. The Fake review detection feature is the input variable that is used to determine the class of the review content as a genuine or fake opinion. According to research by Mukherjee et al., behavioral features are more advantageous than linguistic features for classifying fake reviews easily, and they enable us to enhance classification performance and achieve high accuracy (Mohawesh, Xu, et al., 2021a).

Contextual features, usually referred to as linguistic features, are derived from review-centered features and represent several perspectives on the review content. Textual features include semantic, grammatical, and lexical aspects of the review, which can assist in detecting bogus reviews. We use the review content feature to identify the review class using glove word embedding feature extraction.

Behavioral features describe the statistical importance of a user's review and behavior based on his previous and current reviews. The features of the text description can be extracted into a word vector using a variety of approaches. We use the characteristics of the fake reviews to identify the

review used for fake review extraction, such as contextual (textual) and behavioral features. This research uses basic and derivational features of the notation definition listed on APPENDIX D.

- **Positive feedback percentage (PR):** This is the number of positive reviews a reviewer has written in the feedback place.

$$R_{pos}(a) = \frac{N(\{r_a | rating_r \geq 4\})}{N_r(a)}$$ Equation 3.2: Positive feedback

- **The length of the review (RL):** Each review's length is determined by its word count after preprocessing the review contents.

$$R_{Len} = \begin{cases} 1, & len(r\alpha) < x \\ 0, & otherwise \end{cases}$$ Equation 3.3: Review length

- **Reviewer deviation (RD):** To calculate RD, we first estimated the absolute rating deviation of a review from other reviews of the same business based on their star ratings.

$$RevDev(r) = \left| rating - \frac{\sum rating_r(p)}{N_r(p)} \right|$$ Equation 3.4: Reviewer deviation

- **Positive-to-negative ratio:** It shows the ratio of a reviewer having more than or equal to 4 reviews per rating value to the reviews having less than or equal to 2 rating values.

$$R_{pn}(a) = \frac{N(\{r_a | rating_r \geq 4\})}{N(\{r_a | rating_r \leq 2\})}$$ Equation 3.5: Ratio of a review

- **Extreme Rating:** considering the rating attribute in the review dataset, the rating of the reviewers promotes and downgrades the company's business.

$$\boldsymbol{ER}(a) = \begin{cases} 1, & *(r\alpha) \ \epsilon \{1, 5\} \\ 0, & *(r\alpha) \ \epsilon \{2, 3, 4\} \end{cases}$$ Equation 3.6: Extreme Rating

- **Unique words**: The unique words in the review content that were utilized to characterize the customer's opinions were used to compute the class of the reviews.
- **Helpful feedback**: the reviewer votes for helpful reviews by ratings useful or not to

identify the review's reliability and truthfulness.

- **Polarity of the content**: We used the polarity of the text using the text blob pretrained NLP library. Text Blob is a natural language processing package written in Python. It uses NLTK because it is simple, easy to deploy, consumes fewer resources, provides dependency parsing, and can be utilized for small applications. Text Blob can be utilized for advanced textual data analysis and manipulation. When a sentence is entered into Text Blob, it produces two results: polarity and subjectivity. We used polarity as the output that falls between [-1, 1], where -1 denotes negative sentiment and +1 denotes positive sentiment.
- **Subjectivity:** refers to personal judgments and opinions that define the output, such as 0 and 1. Contextual features, usually referred to as linguistic features, are derived from review-centered features. Contextual characteristics represent several perspectives on the review content. Utilize the characteristics in the fake review detection area, they were summarized and analyzed.

- **Part of Speech (POS) tagging:** Part-of-speech tagging is the process of classifying words into their parts of speech and correctly identifying them. A POS is a grammatical classification consisting of verbs, adjectives, adverbs, nouns, and grammatical factors (Usop et al., 2017). Ambiguity is a fundamental issue that must be addressed as part of speech tagging. Identifying properly the tags from the language sentence is most of the time a difficult task for the users. Several ways of automating POS tagging have been adopted, including transformational-based, rule-based, and probabilistic approaches. We use part-of-speech tagging to calculate the occurrence and the frequency of each part of speech in the textual feature to determine the class of review content (Chiche & Yitagesu, 2022).

3.2.2. Word Embedding

We use pretrained word embedding for contextual feature review content created by Stanford for producing word embedding (Jeffrey, P; Richard, 2014). We conduct comparative study on the word embedding such as FastText, Word2vec and glove to get good performance on the fake review detection model. The word embedding was trained on massive datasets of million and billion words are gathers the whole word-to-word co-occurrence matrix of a corpus. It’s tokenizing

each word in a sequence and transforming it into vector space by capturing the semantic meaning of words in a review text stream.

As previously stated, we use the word embedding to construct vector representations of review content words and to produce good results in fake review detection. We focuses on word co-occurrences across the entire corpus and achieves better results by employing global matrix factorization and local context window approaches. The representations are trained using global word-to-word co-occurrence data that has been compiled from a corpus. Furthermore, the word embedding was gives more practical significance to word vectors by focusing on the interactions between word pairs rather than individual words, and assigns less weight to frequently occurring word pairs.

3.3. Model Selection

3.3.1. Hybrid Deep Neural Network

Deep learning models produce high-performance accuracy in text categorization and extract higher-level features from the input text. The convolutional neural network and bidirectional long-term memory techniques are used in the process of model selection. CNN was used to create a model of the review text because it is a powerful representational tool for complex semantic data that is difficult to define using conventional engineering methods. A fully connected layer maps the outcome after the convolution and pooling layers have extracted the features. Bidirectional recurrent neural networks are simply the combination of two independent RNNs. This structure enables networks to contain forward and backward information about the sequence at each time step. Bidirectional LSTMs are a type of LSTM that can increase prediction accuracy in sequence classification tasks.

The combination of CNN and BLSTM has been demonstrated to capture both local and sequential elements of input data in numerous categories. CNN is well-known for its capacity to extract feature as many elements as possible from review text. BiLSTM maintains the chronological order of words in a document. It can learn temporal and context aspects from text and capture long-term connections between text entities and essential features in NLP tasks, which are discovered using CNN's spatial relations ability. The hybrid neural network model can improve the generalization and has great potential in related text classification such as fake news detection. In several classifications, the combination of CNN and BILSTM has been shown to capture both local and

sequential aspects of input data. BILSTM can learn temporal and context aspects from text and capture long-term connections between text entities and essential features in NLP tasks, which are discovered using CNN's spatial relations ability. The hybrid neural network model can improve the generalization and has great potential in related text classification such as fake news detection.

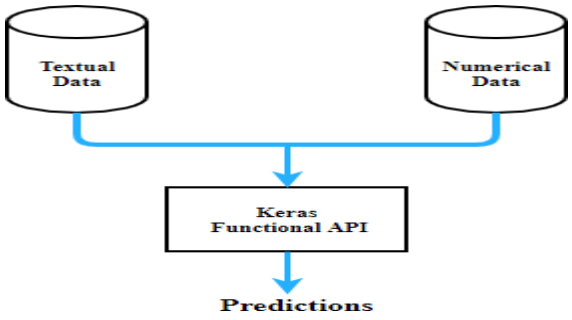

Figure 3-3: Keras Functional API for the concatenate

3.4. Development and Design Tools

To design the proposed research model, we use a variety of development tools provided that were employed in the thesis. These include model development tools, UML modeling tools, and other tools that are important for this research. We described this development tool with a short introduction.

3.4.1. Design Tools

The study used design tools to build, present, and understand the design concepts of models. The draw.io and XMind software design tools were used to design the proposed methods to make various diagrams and for designing this work. This design tool is a simple yet effective visual design tool for making professional-looking flowcharts of the study, proposed architecture diagrams, and network diagrams of the research.

Draw.io[17]: is a software that offers a lot of shapes with simple drag-and-drop capabilities tools. It also allows us to save the diagram on a local computer and diagramming software that allows us to make flowcharts and customized diagrams.

[17] https://drawio-app.com/

XMind[18]: is brainstorming and mind-mapping software that is used to capture ideas, clarify thinking, manage complex data, and improve team cooperation. It's used to schedule the diagrammatically, save the picture of the research, and visualize the diagram.

3.4.2. Hardware Development Tools

To implement this research, the following hardware tools are needed:

- **Hard Disk**: to store the research materials, documents, and datasets related to fake review detection from different sources to keep the files of the research.
- **RAM**: We used to work with the GPU to speed up the training and testing of our model process performance.

3.4.3. Software Development Framework Tools

For implementing the research by coding, different software and coding tools are used, and those are listed below with their descriptions.

Anaconda[19]: is a high-level, general-purpose distribution of the R and python programming languages for machine learning and data science. To implement the proposed approach, we used the anaconda application version with 64-bit support, which is used to install the latest version of python together with all of its numerous modules and IDEs.

Microsoft office package[20]: is a collection of office applications used to write this thesisdocument on word office, present the presentation using PowerPoint, and formulate calculation tables on excel. We used Microsoft Excel for data preparation tasks in cleaning, filtering, and sorting the gathered data from the sources. Microsoft PowerPoint is a presentation program, we used it for presentation.

Google Colab[21]: is a platform that allows developers to use google cloud to write and execute python code in the browser. It's ideal for deep learning and machine learning, as well as data analysis and education purposes. It has free access to GPU, TPU, and Zero Configuration, plus easy sharing of the project or code within a limited time per day. We use Colab to build our model implementation and visualize the data analysis with the function used to build the model.

[18] https://www.xmind.net/
[19] https://www.anaconda.com/
[20] https://www.microsoft.com/en-ww/microsoft-365/microsoft-office
[21] https://colab.research.google.com/?utm_source=scs-index

Google Drive[22]: We use it to store the thesis document, annotate datasets, and code the model for easy access in the Colab running environment.

Jupyter Notebook[23]: is the open source software most widely used and most convenient IDE for working with python among AI and deep learning researchers. We use the Jupyter notebook in this research to clean and transform review data, draw statistical diagrams to show the review dataset, and visualize the analysis of the result using different libraries. It's been integrated with anaconda navigator, and we've built our code on top of it.

Mendeley Desktop[24]: is a free platform application that acts as a scholarly, simplifies the workflow, and social network for the reference of related works that have been done before in the fake review, and makes it easy to cite the references of this study.

Tensor Flow[25]: Tensor Flow is a freely available software framework for high-performance numerical analysis and processing. Use this framework to easily access the library and functions used in building the model as we build our deep neural learning model of the fake review. We use the Colab Tensor Flow to work with the installed library.

Keras[26]: is a free open-source library that serves as an interface for Tensor Flow, offering a python interface for building neural networks. Installed Keras using Google Colab to import the library and functions useful for this study.

Python[27]: is the programming required to implement and demonstrate this thesis requires many of the drivers used to configure and package to install some devices on the computer. It is an excellent choice for software developers, researchers, and data analysts working with machine and deep learning models. The method we used to implement and demonstrate this thesis requires many of the drivers to be configured and packaged to install some devices on the computer.

3.5. Baseline work

To validate the fake review detection model's performance, it was compared to a previous method for detecting fake reviews in the electronics domain area. We chose two papers as the foundation for our study: the first is from the electronics domain (Barbado et al., 2019), which focuses solely

[22] https://drive.google.com/drive/my-drive
[23] https://jupyter.org/
[24] https://www.mendeley.com/
[25] https://www.tensorflow.org/
[26] https://keras.io/
[27] https://www.python.org/

on the electronics retailers' domain in online consumers. They created a framework for detecting fake reviews and achieved an F-score of 82 percent using the Ada Boost classifiers machine learning model. Second, papers from a variety of domains have strong performance and are connected to our models for our baseline study (Jeffrey, P; Richard, 2014). They proposed a hybrid model based on the Denoising Autoencoder and the Paragraph Vector Distributed Bag of Words models. The Gold Standard datasets include 1600 reviews from Orbitz, Priceline, TripAdvisor, Expedia, Hotels.com, and Yelp. They got 92.5 percent accuracy and 92.4 percent F1 measure, however, they don't take into account semantic and emotional factors.

3.6. Proposed Model Evaluation

Deep learning models are used in this study, and the model should be able to make accurate predictions to provide real value to an organization. A model developer divides data into training and test data to perform during training. The evaluation is necessary to thoroughly analyze the work result. The models were evaluated using the test set in the sample data. In this study, the trains, tests, and evaluates the performance of the proposed system to determine how well the model generalizes to previously unseen data. The goal of the model evaluation is to determine the model's generalization accuracy on unknown data.

3.6.1. Stratified K-Fold Cross-Validation

Cross-validation is a re-sampling process used to evaluate a model and check its ability to generalize to unseen data. The main processes in cross-validation include separating the data into folds and rotating the training and validation among them. For reorganizing review data, we employed Stratified K-Fold Cross-Validation, which ensures that each fold accurately represents the whole dataset (Raschka, 2018).

Why do we use Stratified K-Fold Cross Validation? As Stratified K-fold Cross-Validation randomly shuffles the dataset and divides it into folds, the chances of getting highly imbalanced folds are high, causing model training to be biased. It could be useful for our review datasets to have a balanced distribution of classes. We reorganize the review datasets, and we're using review datasets to ensure that each fold is representative of the entire dataset. It is one of the most effective methods for minimizing bias and variance.

3.6.2. Classification Metrics Methods

Model Evaluations are used to evaluate the performance and describe how a model works efficiently and evaluate relationships (Raschka, 2018). Different machine learning algorithms use different evaluation metrics to explain the performance of their work, and their evaluation criteria vary from one to another. The evaluation metrics used for this research are used to review the proposed system model to see if the desired goal has been achieved. In machine learning, the success rate and efficiency of the model were assessed using several measures.

- **Accuracy:** We used the accuracy evaluation to know how many observations the model classified correctly for the reviews. This is the degree to which a particular set of measures is close to or far from its real value.

$$\text{Accuracy} = \frac{TP+TN}{Total\ instsances\ (TP+FN+TN+FP)}$$ Equation 3.7: Accuracy Calculation

- **Precision:** is calculated by dividing the total number of true positive reviews by the total number of true positive reviews plus the number of false positive reviews captured by the proposed model from all anticipated positive cases of the review datasets.

$$\text{Precision} = \frac{TP}{TP+FP}$$ Equation 3.8: Precision Calculation

- **Recall:** the ratio of relevant review documents retrieved by a search to the total number of relevant review documents was accurately classified as positive out of all positives in the dataset.

$$\text{Recall} = \frac{TP}{TP+FN}$$ Equation 3.9: Recall Calculation

- **F1 score:** is the harmonic average of the recall review value and the precision review detection value. It takes into account the false negatives and positive review documents.

$$\text{F-Score} = 2*\left(\frac{Recall*precision}{Rescall+Precision}\right)$$ Equation 3.10: F1 Score Calculation

- **Confusion matrix:** is used to evaluate the classification task of review performance described and visualize the important predictive analytics. We use the confusion matrix report to easily show the report of the training accuracy and testing accuracy after the classification task and a summary of predicted outcomes. The total number of correct and incorrect predictions is added together and divided by class using count values.

Table 2: Confusion Matrix for classification of the reviews

Prediction Values of Reviews	**Actual Values of Reviews**	
	Positive Review	Negative Review
Positive class	TP	FP
Negative class	FN	TN

We have two classes in our case, so the confusion matrix is two by two matrix.

True Negative (TN): When the data point's actual class of review was false and the predicted class was false as well.

False Negative (FN): When the data point's actual class of review was true and the predicted class was false.

True Positive (TP): When the data point's actual class of review was true and the predicted class was true as well.

False Positive (FP): When the data point's actual class of review is false and this predicted class is true.

3.6.3. Sample prototype of Fake Review Detection

We created a user interface using a free online HTML5 framework and CSS file and used the HTTP server to access the Flask model. To create a responsive web application, we used the open-source Bootstrap framework to develop the user interface. The purpose of building a user interface is to use the model with new data entered by the model's user. To publish the sample prototype it need some integration with the e-commerce site.

CHAPTER FOUR

4. PROPOSED MODEL ARCHITECTURE FOR FAKE REVIEW DETECTION

4.1. Introduction

This chapter, discuss the proposed model architecture, tools needed for experimentation, diagrammatic representation, and components that are incorporated to make the proposed model for improving the current model. In addition to the architecture, the process and steps for model implementation used in this research are discussed. We define the models that were used in the study to address the research questions and describe a process for the problem raised in the preceding chapters, as well as a component of the model utilized in this research.

4.1.1. Proposed Fake Review Detection System Architecture

To classify the reviews, we used the proposed architecture of hybrid deep neural network techniques. This study's overall proposed architecture serves as a guideline for the project's work, trained using e-commerce datasets, and deep learning techniques for solving problems are an important activity in this study. After loading e-commerce datasets, the implementation process of the preprocessing step begins. It takes e-commerce review datasets as input and then preprocesses the review, including punctuation removal, cleaning, normalization, formatting, tokenization, and other basic preprocessing. The model is then trained using CNN, BILSTM, CNN-MLP, CNN-GRU, CNN-LSTM, and CNN-BILSTM deep learning algorithms.

To determine the most performing model, stratified k-fold CV and other metrics evaluations are used to build predictive models and their predictive performance. And the results are obtained using the model evaluation method we discussed, the detection model is evaluated and selected. Various performance evaluation metrics are also employed to obtain the most accurate model capable of providing the best future prediction on the test dataset. The best classification model was chosen based on the results of the performance evaluation or test accuracy. The deployment of the fake review detection model is the final step after all of these stages have been completed.

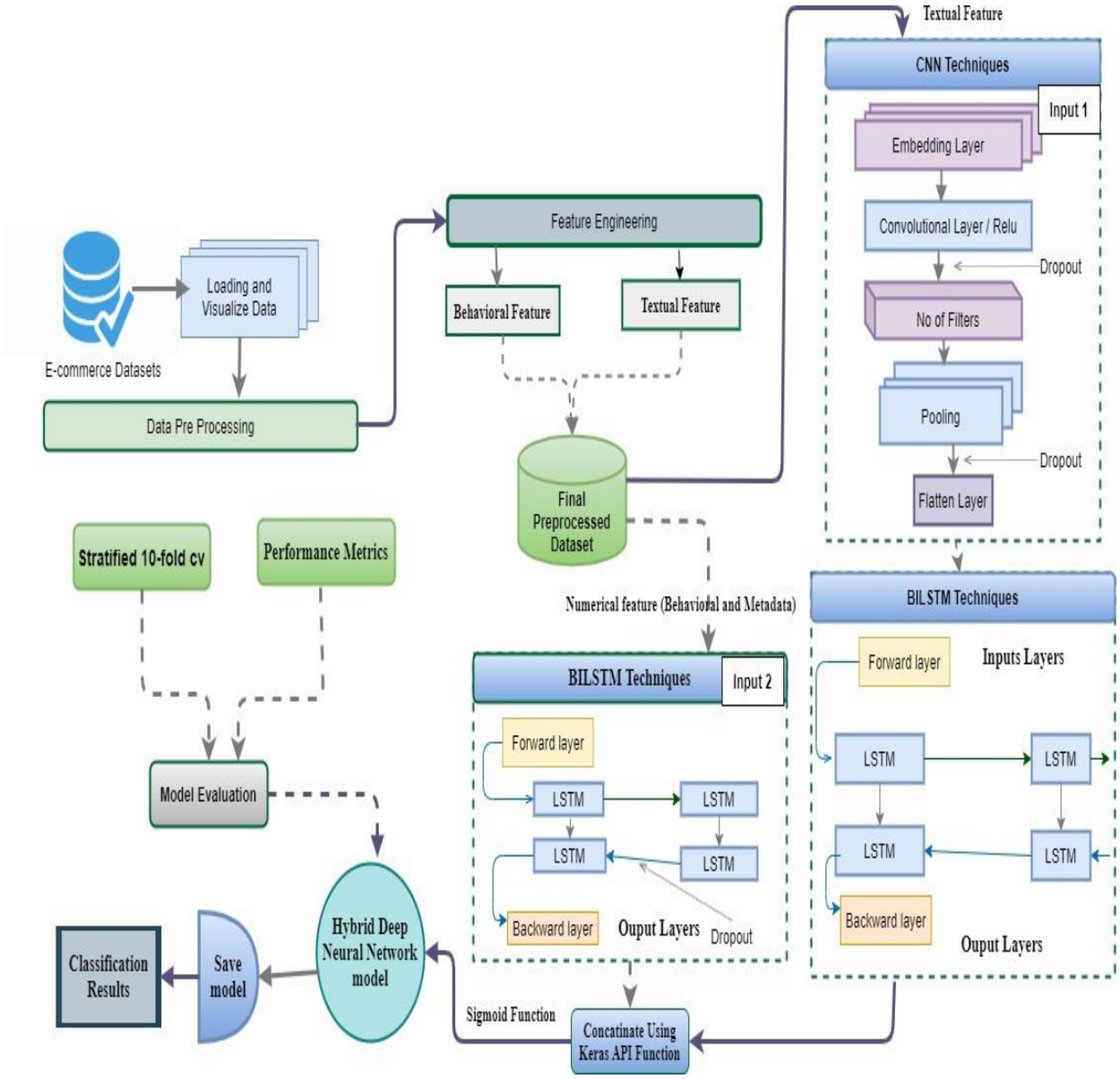

Figure 4-1: Proposed Architecture of fake review detection

4.2. Data preprocessing phase

In machine learning, data preprocessing is a critical step that improves the data quality and facilitates the extraction of meaningful insights from the data (Borowska & Topczewska, 2014). Data preprocessing techniques is a processes of changing raw data into a clean format that is easily readable and understandable for the machine.

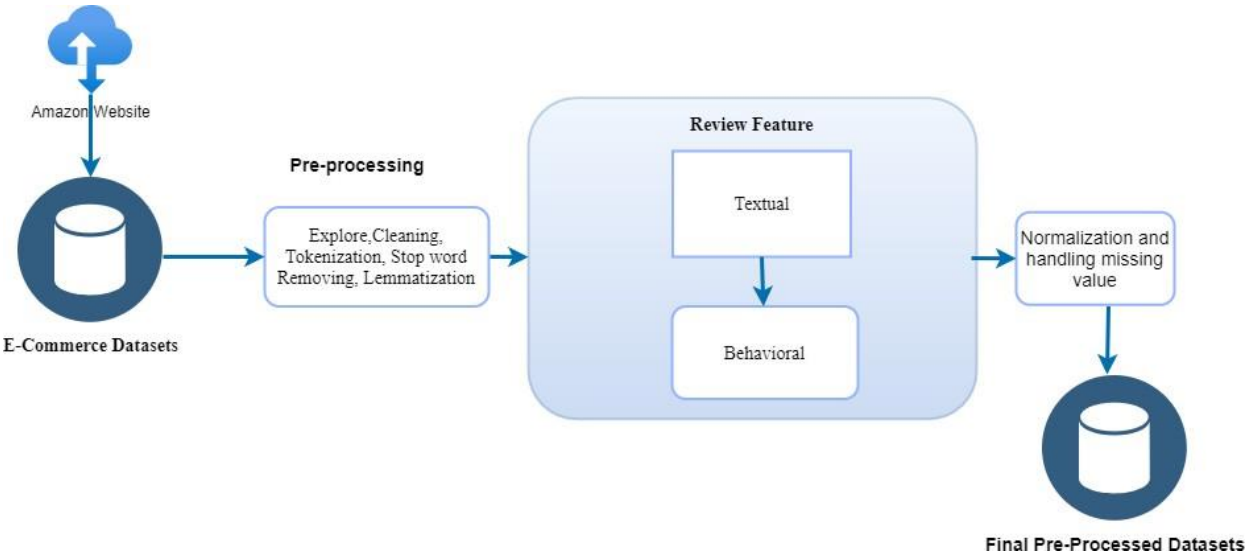

Figure 4-2: The Block diagram of the Dataset Preprocessing phase

Explore Review Dataset- The first step to applying the pre-processing on the review dataset it needs load data into the python environment. Before doing any pre-processing on the data, the data should be made available to the python environment we are doing. To achieve this goal, we use pandas as it supports loading data as data frames. The panda's method uses a read_csv function to reads the file in the commas separated value file and loads it as Data Frames, where instances of the data are accessible via column name. The total dataset before the balancing class distribution is 592837 rows × 11 columns of data.

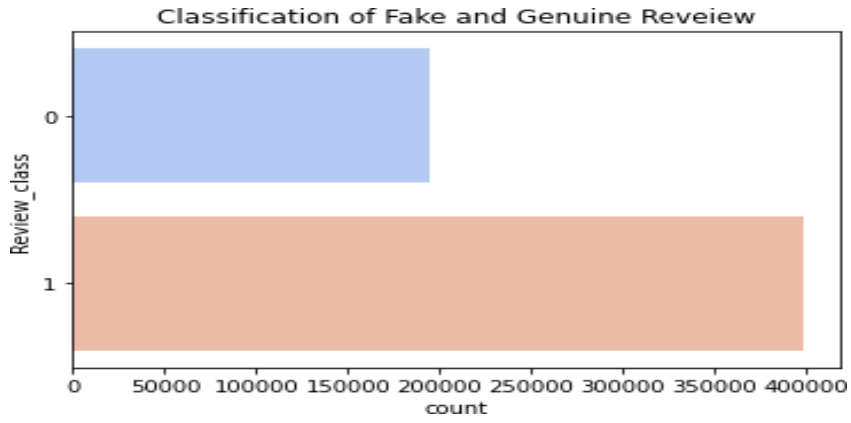

Figure 4-3: Review Dataset Class

Cleaning the Review Dataset- After importing the dataset into our google Colab environment cleaning of irrelevant characters from the dataset. The dataset that was used for our research has duplicate records and the first step was to remove the 0.00087 present duplicated review. We remove the punctuation from the text of the review, special characters, and empty text, and convert the words to lowercase to prepare the review content for machine understanding. Following this, we modified the date field of all the fake reviews to ensure that the formatting was consistent.

Tokenization- We use tokenization to process a long piece of review text into smaller tokens of words by breaking it down. Tokens can be words or sub-words that aid in interpreting the meaning of the text by examining the word sequence. Splitting the review content into an individual sequence of words. To implement tokenization in this study, we used python code that uses space to distinguish between words.

Stop Words Removal - Stop words are one of preprocessing steps that remove the words that hold no significant value and are repeated multiple times. Articles, prepositions, conjunctions, andsome pronouns are considered stop words such as, from, how, in, is, of, on, or, an, are, as, at, be, the, these, was, what, when, where, who, will, etc. We used Python Natural Language Toolkit (NLTK) tools to extract stop words from review datasets.

Lemmatization - This process is also used to reduce a word to its word stem, which is the attached suffixes and prefixes to the roots of words. It returns a word's dictionary form, which must be correct, whereas stemming only returns the root word. For example, "good," "better," or "best" is lemmatized into "good." To stem the token of datasets we used WordNetLemmatizer to make the classification of review that is efficient and effective using most commonly lemmatized the word with the word root due to its performance.

Part of Speech (POS) tagging

Part of Speech Tagging is a well-known NLP topic that seeks to assign each word in a text the appropriate syntactic tag in its presentation context. The automatic Pos labeling of words in a review's contents was done using grammatical tagging. The grammatical classification of the review contents in the part of speech tagging consists of verbs, adjectives, adverbs, nouns, and other grammatical factors. Ambiguity is a fundamental issue that must be addressed as part of speech tagging. It might be challenging to determine the proper tag for a word that appears in a

given sentence because words in most languages behave differently in various contexts. Several ways of automating Pos tagging have been adopted, including transformational-based, rule-based, and probabilistic approaches. The origins of Pos tagging can be traced back to the uncertainty of many words when it comes to their part of speech context. Part of speech tagging is the occurrence or the frequency of each part of speech in the textual feature of the review to determine the class of review content.

4.3. Hybrid model Learning

4.3.1. Multilayer Perceptron (MLP)

The multilayer perceptron is one of the useful neural networks that is fully connected to the layers of nodes with each other. To analyze multidimensional data, a multilayer perceptron must remember patterns in sequential data, which necessitates a high number of parameters and consists of multiple hidden layers to capture more complex relationships that exist in the training dataset. We use Tensor Flow Keras to implement a multilayer perceptron neural network with preprocessed data using the Colab python environment. After the data preparation, split train and testing data set reshape the data into one dimensional using the Numpy function. Train the Multilayer perceptron neural network with a three-layer to get good accuracy and at the output layer use the sigmoid activation.

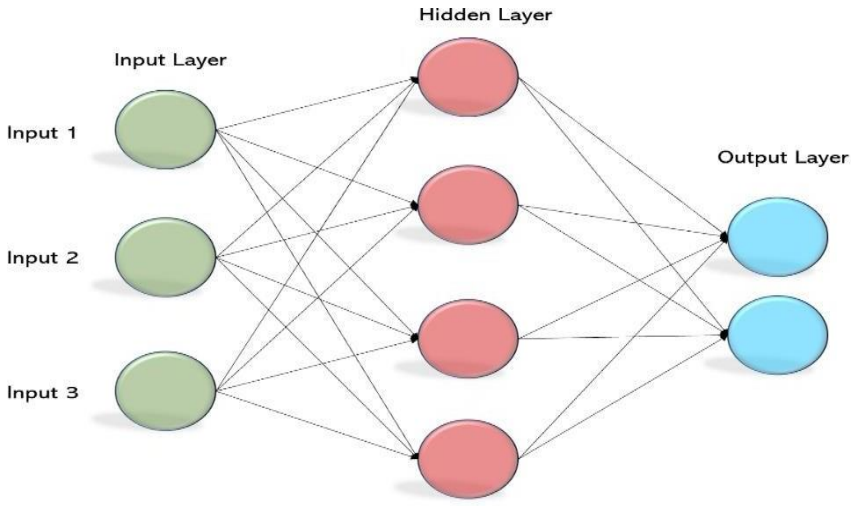

Figure 4-5: Block diagram of Multilayer perceptron neural network

Increasing the number of hidden layers in the network design can increase the representational quality of the network. However, increasing the number of hidden layers arbitrarily has an impact on overall network performance when it comes to generalizing to unseen data of reviews.

4.3.2. Convolutional Neural Network (CNN)

CNN is a deep learning technique mostly used for efficient image recognition and computer vision. It's a feature extraction model that learns to extract important information from review contents using word embeddings. In the convolutional layer, neurons in the higher layers with wider receptive fields receive information from neurons in the lower layers with smaller receptive fields. As the input data flows through the network, CNN learns a hierarchical set of progressively complicated properties and is incapable of effectively interpreting temporal information.

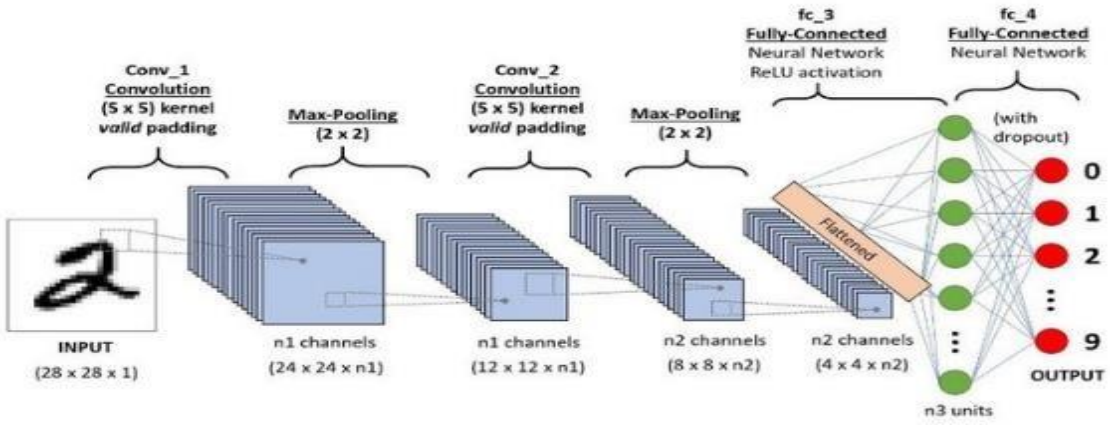

Figure 4-6: Block diagram of Convolutional Neural Network adopted from (Saha, n.d.)

4.3.3.Gated Recurrent Units (GRU)

A gated recurrent unit is a recurrent neural network that performs machine learning tasks like memory and clustering in speech recognition by using connections between nodes. GRU networks are similar to LSTM networks, a more recent variant of RNN with two gates that makes it faster. Update Gate combines the forget gate and the input gate. The forget gate determines which information to discard and which to store in memory. Second, Reset Gate the previous information to remove the gradient explosion. The Reset Gate determines how much information from the past should be deleted.

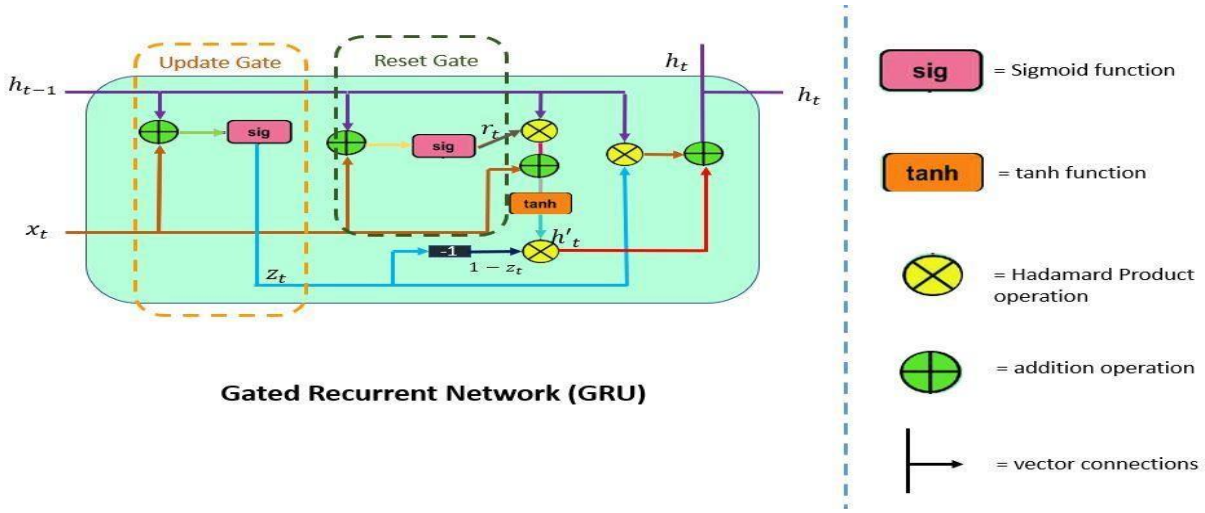

Figure 4-7: Block diagram of Gated Recurrent Units

4.3.4. Long Short Term Memory (LSTM)

[28]Long short-term memory networks are sequentially techniques to recall patterns over long periods in the deep learning category for text classification. It's an excellent choice for modeling sequential data and, as a result, it's frequently employed to learn complex human activity dynamics. LSTMs are expressly designed to avoid the problem of long-term dependency and designed to handle time-series data. The fundamental distinction between the LSTM and the RNN is the addition of a sequence of memory gates, including the forget gate, input gate, and output gate. They don't have to work hard to recall information for extended periods; it comes naturally to them. All recurrent neural networks are made up of a series of neural network modules that are repeated over and over again[29]

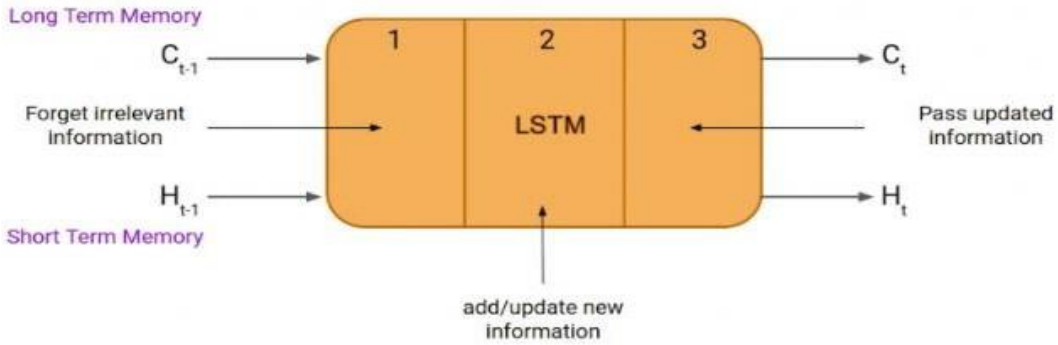

Figure 4-8: Long Short Term Memory Diagram

[29] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

Whereas H (t-1) represents the hidden state of the previous timestamp, and Ht represents the hidden state of the current timestamp in the LSTM. A cell state of the long-term memory, represented by C (t-1) in prior status and C (t) current timestamps.

4.3.5. Bi-directional long short-term memory (BI-LSTM)

Bidirectional long-short term memory allows any neural network to store sequence information in both the future and the past. The goal is to look at a sequence from both front-to-back and back-to-front. As a result, the network constructs a context for each character in the text that is based on both their history and future. Bidirectional long short-term memory is identical to its unidirectional sibling in appearance. The difference is that the network is linked to both the past and the future. Because our input runs in two directions in a bidirectional LSTM, it differs from a conventional LSTM.

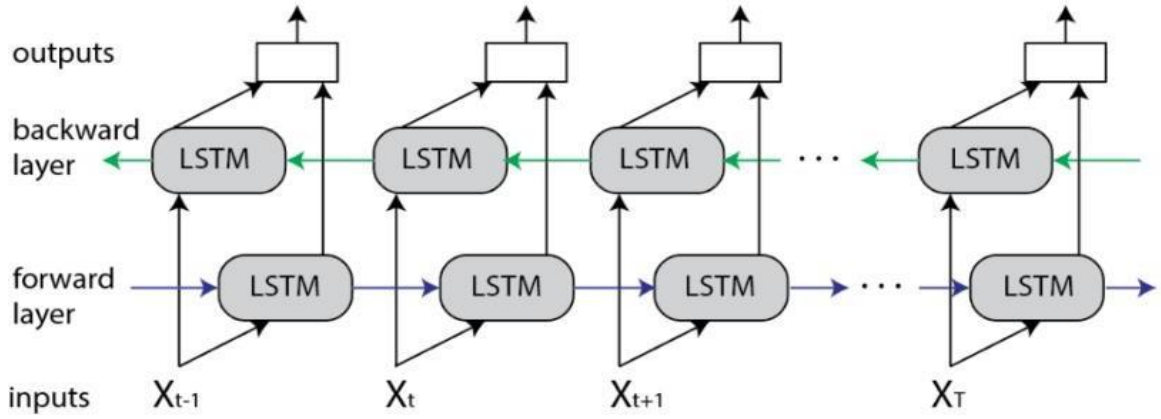

Figure 4-9: Block diagram of BI-LSTM Networks[30]

4.3.6. Hybrid CNN-BILSTM model

Hybrid model: - combines the CNN's capacity to extract local features with the BILSTM ability to learn long-term dependencies (Pendyala, 2019). The CNN neural network is used to learn the sentence representation, next by the BILSTM neural network to learn the document representation, and directly utilized as the feature to identify fake reviews from the guanine reviews. A CNN is used to process the input vectors, extracting the local features that exist at the contextual level. The BiLSTM model is useful in several NLP applications, such as sentence categorization, translation,

[30] https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm

and entity recognition. It includes a backward direction LSTM layer of data flow or the model used in both directions the layer can work. The BILSTM layer utilizes the local features retrieved by the CNN layer as input and learns the long-term relationships of the local characteristics of reviews to categorize them as fake or genuine reviews.

The Keras deep learning library's functional API model is used to implement this hybrid deep learning model for fake review detection. There are multiple layers of neurons in the functional API model. The first layer of the neural network is to initialize the embedding layer with a vector matrix and the number of features with Keras. The embedding matrix uses the pre-trained glove word embeddings and a textual input with a dropout layer through this input layer. The next layer is the convolutional layer construction, which extracts local features by using 256-filter-sized filters, padding, and Relu activation. Create the MaxPooling layer with the pool size and append the flattened layer variables to the pooling layer to downsample the feature vectors, reduce the number of parameters, and thus the computations without affecting the network's efficiency.

The BILSTM layer's second functional API input is computed by first computing the features with filters, padding, relu, and then adding the dropout layer to the model. After inserting numerical features of the behavioral attribute, appending with a BILSTM layer. Concatenate both inputs using the Keras API function and activate the sigmoid layer. Execute the compile function using the optimizer called adaptive moment estimation (Adam) to define the learning rate in each iteration. We used a binary cross entropy as the loss function for the model, and the accuracy of the model for the evaluation of results. The training is performed for 50 epochs using a batch size of 250, and the fitness function is computed using epoch, Xtrain, and Ytrain values. Finally, visualize and evaluate the model with Xtrain and Ytrain using the confusion matrix and plot curve. Composed of multiple building blocks of a convolutional neural network has three types of layers such as:

Convolutional Layers - comprised of filters and feature maps. Convolution is the technique of applying a function to a matrix to extract specific information from review text. It transforms the dataset text into a feature map, which represents what the kernel has learned from the data. The feature map is translated into another feature map, and the filter argument determines how many feature maps are generated for each convolution. A convolutional filter, often known as a kernel, is a function that is implemented as a sliding window through a dimensional matrix.

Pooling Layers – are a building block in CNN that comes after the convolutional layer and is used to organize layers inside a convolutional neural network that may be repeated one or more times in a given model. It operates individually on each feature map of the layer to generate a new set of pooled feature maps with the same number of features. The function of a pooling layer is to reduce the dimensionality of the input text characteristics and apply subsampling to the output of the convolutional layer matrices by merging neighboring parts. The most significant element is selected using the max pooling approach by the filter. As a result, the output from the max-pooling layer would be a feature map that contained the most prominent features from the prior feature map. The pooling layers are also required to lower CNN's processing time and overfitting concerns.

Fully-connected Layers

The CNN's fully connected layers enable it to mix signals of information between each input dimension and each output class, allowing it to make decisions based on the entire dimension of inputs reviews and assign a class to it. We don't need to do feature engineering because CNN's capture better representations of data. The classification is handled by the fully connected layer, which converts it to a single vector that can be used as an input for the next stage. Finally, the output probabilities for each class are computed using the sigmoid function. In other words, it determines which values are passed as output and which are not.

Algorithm of Hybrid CNN-BILSTM

Input: Preprocessed Ecommerce Review Dataset

Output: Accuracy

Begin Procedure ( )

Initialize the embedding layer with a vector matrix, and several features.

Add the textual input one with the embedding and add the dropout layer

Construct the Convolutional layer using filters, padding, and Relu activation function

Produce the MaxPooling layer using pool size

Append the Flatten layer variables with the pooling layer

Compute the BILSTM layer using filters, padding, and Relu activation function

Add the BILSTM layer with the dropout layer

Insert the Input two of a numerical feature of behavioral attribute

Append with BILSTM layer using filters and Relu activation function

Concatenate both inputs using the Keras API function.

Activate the Sigmoid layer

Execute compile function using the optimizer and metrics accuracy

Accuracy is measured using the compile function

For every epoch in model

Compute the fitness function using epoch, Xtrain, and Ytrain values

Evaluate the model with Xtrain and Ytrain

Compute the total accuracy

End For

Return Accuracy

End Procedure

To categorize the review as genuine or fake content, we implement the binary cross-entropy loss function, which is generally applied after the sigmoid activation layer. An epoch is a complete sample of the entire training data in the fit algorithm. Reduce the learning rate from 0.1 to 0.001 as the loss approaches the minimum accuracy in classification review. The next critical step is to

evaluate the proposed trained model's performance to see if it has produced a satisfying result. For equivalent performance measures, training the model for 50 epochs was sufficient.

4.4. Prediction module for review Detection

The purpose of evaluating and testing is to describe the designed system's evaluation metrics and compare them to the test results. We need to check the accurate estimation of the trained model's generalization error on the provided dataset to evaluate and test the deep learning models' ability to learn the trained fake review detection dataset. In this study, the experiment is based on review detection classes, and each evaluation metric associated with each sentiment polarity class is evaluated. To achieve this, we employ various performance evaluation strategies to assess the effectiveness of our proposed hybrid deep learning and neural network algorithms. Moreover, we used metrics such as F-measure (F1-score), confusion matrix, accuracy, precision, and recall.

4.5. Model Prototyping

We develop a sample prototype on the flask web server after the model train and test the models using cross-validation and metrics evaluation, to predict whether the model detects it is guanine or a fake review. We save the trained model first using pkl method as a file. Then loaded onto the flask, which makes it accessible to users via the WSGI HTTP server.

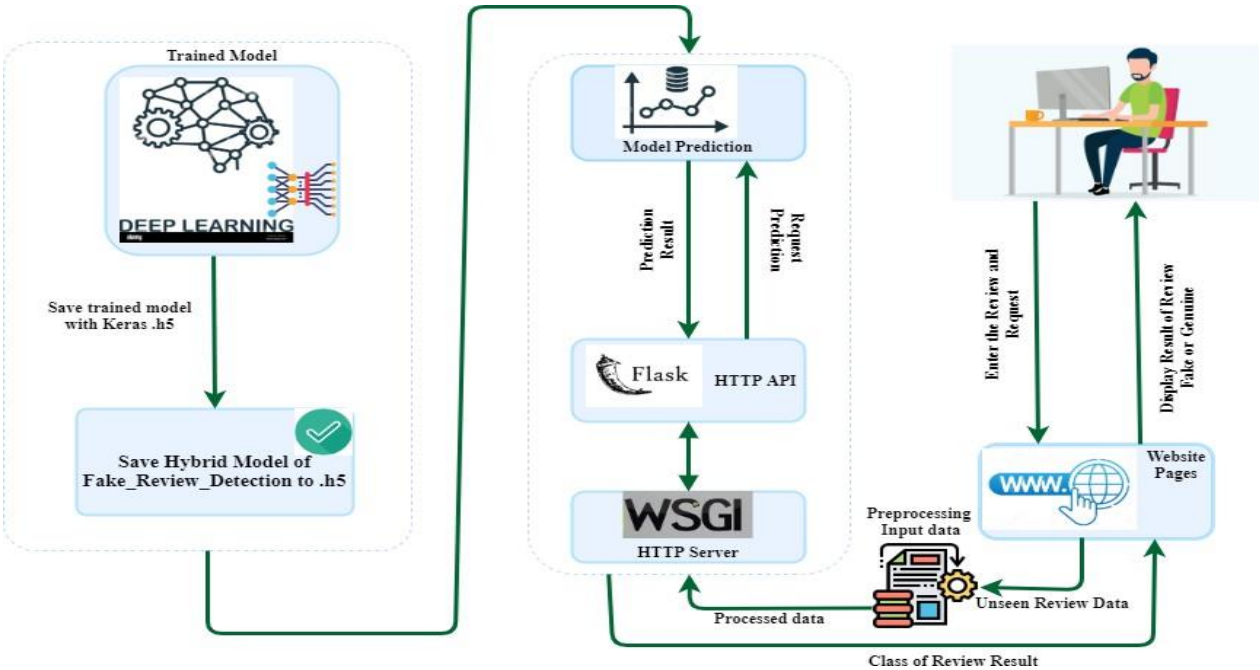

Figure 4-10: Deployment Architecture of Hybrid CNN-BILSTM Model

CHAPTER FIVE

5. EXPERIMENT AND IMPLEMENTATION

5.1. Introduction

In this chapter, we explain the experiment and implementation of the proposed solution for fake review identification, utilizing the methodology presented in chapters three and chapter four. The goal of this study is an experimental strategy to find the best experimental result of the deep neural network algorithm and detailed implementation of preprocessing, model building, and testing final model comparison.

5.2. Working Implementation Environment

Various tools and packages are used to implement the fake review detection model. Python is our preferred programming language because it is a simple yet powerful programming language with excellent neural network processing capabilities. We started with data preprocessing and end with model construction. This study used the python programming language to implement and test each recommended solution in chapter three. We also used it to test the proposed classifiers model that had been constructed.

5.2.1. Tools and python packages

We discuss the modules that have been implemented, as well as the libraries and header files that have been used in this research. Those libraries are listed, along with their version and description.

Table 3: Tools and packages used for work implementation environment

Library	Version	Descriptions
TensorFlow[31]	2.6.0	To build our deep neural learning model, we use this framework to easily access the library and function in building the model.
Keras[32]	2.4.0	Install Keras using google Colab and Jupyter notebook to import the library and function useful.
Scikit-learn[33]	0.23.2	Used this library for training and testing models.
Nltk[34]		A Python library used for tokenization and pre-processing.

[31] https://www.tensorflow.org/
[32] https://keras.io/getting_started/intro_to_keras_for_researchers/
[33] https://scikit-learn.org/stable/getting_started.html
[34] https://www.nltk.org/

NumPy[35]	1.21.1	To convert text to numeric data for features.
Pandas[36]	1.3.3	It is used to read, manipulate, write, and handle the data frame.
Genism[37]		Used for topic modeling and document indexing. To constrict the glove word embedding model.
Seaborn[38]	0.9.0	For statistical graphs and to represent mathematical diagrams.
Matplotlib[39]	3.4.3	A plotting library for Python and its NumPy numerical math extension used data and results visualization
TextBlob[40]	0.16.0	To calculate the polarity and subjectivity of the review.

5.2.2. Package manager and Environments

Google Colab[41] is a platform that allows developers to use Google Cloud to write and execute python code in the browser. We use Colab to run our code and install the library that is useful for the function used to build the model. This can improve the efficiency of the python programming language to develop deep learning applications using popular libraries such as TensorFlow, Keras, etc.

Anaconda Navigator[42] is a graphical user interface for the anaconda individual edition that allows you to easily launch applications and manage packages and environments without having to use command-line tools. To implement the proposed approach we used an anaconda application version with 64-bit support, which is used to install the latest version of Python together with all of its numerous modules and IDEs.

5.2.3. Deployment Environment

We used Google Colab to implement the model, which is a free GPU and TPU computation platform provided by Google. A Google collaboration that includes a Nivida K-80 GPU and 12 GB of RAM. The tools which are discussed above have been deployed on a desktop computer with an Intel® CoreTM i5-4590 processor running at 3.30GHz, 12.0 Gigabytes of physical memory

[35] https://numpy.org/
[36] https://pandas.pydata.org/docs/user_guide/dsintro.html#dataframe
[37] https://radimrehurek.com/gensim/index.html
[38] https://seaborn.pydata.org/
[39] https://matplotlib.org/
[40] https://textblob.readthedocs.io/en/dev/
[41] https://colab.research.google.com/
[42] https://docs.anaconda.com/anaconda/navigator/

(RAM), and a hard disk storage capacity of 936 Gigabytes or 1TB with Windows 10 Pro, 64-bit is the operating system used for the research deployments.

5.3. Dataset Description and Loading of Dataset

5.3.1. Preprocessing Implementations

Data preparation in machine learning is the process of getting the raw data cleaned and organized so that it can be used to create and train machine learning models. We began by loading the review dataset file with the corresponding annotation file before beginning any of those preparation techniques (Goyal, 2021).

Exploration of Dataset

This research uses the python programming language and modules to implement the fake review detection data loading and reading. We use the code in figure 5-1 to perform the entire implementation of the methods, which includes importing important library packages and loading the dataset file to the disk. Throughout the implementation of this study, load using read_csv file as a data frame and access the data through column name of datasets.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
#Review DataSet from grive using Pandas Library
import pandas as pd
path_datasets ='/content/drive/MyDrive/Hybrid_Fake_Review_Detection_Model_Final_code/upate_final_final_dataset.csv'
data_df = pd.read_csv(path_datasets,header=0, parse_dates=[0], index_col=0)
```

Figure 5-1: Python Code for Review Dataset using Pandas library

Handling missing values

We implement imputation techniques to handle missing values in the review dataset. In these techniques, missing values are replaced with the average values of the dataset's non-missing data. Mean imputation, which simply computes the mean of the observed values for that variable for all non-missing individuals.

```
#Handling missing Values of review feature
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values = np.nan, strategy='mean', verbose=0)
imputer= imputer.fit(X2_train)
X2_train= imputer.transform(X2_train)
```

Figure 5-2: Handling missing value using simple Imputed

Balancing Dataset values

Balancing is essential to achieve successful results for unbalanced datasets. We used random under-sampling to balance the dataset by reducing the enormous dataset to the smallest one. SMOTE was not working for the string column, and random oversampling duplicated the rows of data that caused oversampling in the model. To choose the optimal balancing strategies using fictitious review data, we examine random oversampling and undersampling techniques.

```
#random undersample
from imblearn.under_sampling import RandomUnderSampler
rus = RandomUnderSampler(random_state=0)
X, y = rus.fit_resample(X, y)
```

Figure 5-3: Random Undersampling code

```
#random oversample
from imblearn.over_sampling import RandomOverSampler
ros = RandomOverSampler(random_state=0)
X, y = ros.fit_resample(X, y)
```

Figure 5-4: Random Oversampling code

Tokenization and Removal of stop words

Tokenize the data once it has been cleaned and normalized so that the model can recognize the context. We used the Keras Tokenizer to accomplish this by selecting the maximum number of words to be indexed. Convert a document into a list of tokens and lowercase the documents; the output is a list of final tokens. Remove stop words and remove words with two or fewer characters.

```
import gensim
# Obtaining Additional Stopwords From nltk for fake reveiew detection
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
# Removing Stopwords And Remove Words With 2 Or Less Characters
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3 and
        token not in stop_words:
            result.append(token)

    return result
# Applying The Function To The Dataframe and Lower casing and removing punctuations
data_df['clean'] = data_df['review_Text'].astype(str).apply(preprocess)
```

Figure 5-5: Python Code for Tokenization and Removal of stop words

Lemmatization Implementation

We use lemmatization to extract a root word from the vocabulary. It returns a word's dictionary form, which must be correct.

```
# Import libraries
# import these modules
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
from textblob import TextBlob
from textblob import Word
# Lemmatization
data_df['clean'] = data_df['review_Text'].astype(str).
apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
```

Figure 5-6: Python Code for cleaning the dataset

5.3.2. Sentimental analysis Polarity and Subjectivity of reviews

To accurately examine the reviewer's opinions, sentiment analysis tools must be able to determine between polarity and subjectivity. We utilize polarity to evaluate content to determine the strength of an opinion and the degree to which a person is personally invested in an object.

```
#Sentimental analaysis using textblob library to determine polarity and Subjectivity
from textblob import TextBlob
reviews= data_df['review_Text'].tolist()
#Calculate the polarity and subjectivity
score = []
subjectivity=[]
review_sentiment=[]
for rev in reviews:
  testimonial = TextBlob(str(rev))
  score.append(testimonial.sentiment.polarity)
  subjectivity.append(testimonial.sentiment.subjectivity)
data_df['Polarity'] = score
data_df['Subjectivity'] = subjectivity
#Polarity of review content
positive = 0
negative = 0
for score in data_df['Polarity']:
    if score > 0:
        positive += 1
    elif score < 0:
        negative += 1
```

Figure 5-7: Python Code for Sentimental analysis

5.3.3. Part of Speech (POS) tagging

In our research, we concentrate on the potential of tagging in identifying fake reviews and obtaining contextual information to improve the performance of models that use Pos tagging.

```
#Number of Negative words in a review
reviews = data_df['review_Text'].tolist()
negative = []
for rev in reviews:
    words = str(rev).split()
    neg = 0
    for w in words:
        testimonial = TextBlob(str(w))
        score = testimonial.sentiment.polarity
        if score < 0:
            neg += 1
    negative.append(neg)
data_df['Neg_Count'] = negative
#Total Word
data_df['Word_Count'] = data_df['review_Text'].str.split().str.len()
reviews = data_df['review_Text'].str.lower().str.split()
#unique words in review
data_df['Unique_words'] = reviews.astype(str).apply(set).apply(len)
```

Figure 5-8: Word length and unique word count of fake review detection

```
POS = ['Noun', 'Adj', 'Verb', 'Adv', 'Pro', 'Pre', 'Con', 'Art', 'Nega', 'Aux']
Values = [Noun, Adj, Verb, Adv,Pro,Pre, Con, Art, Nega, Aux]
i = 0
for x in POS:
    data_df[x] = pd.Series(Values[i])
    data_df[x] = data_df[x].fillna(0)
    data_df[x] = data_df[x].astype(float)
    i += 1
data_df = data_df.assign(Authenticity = lambda x:
                         (x.Pro + x.Unique_words - x.Neg_Count) / x.Word_Count.astype('float'))
```

Figure 5-9: Pos tagging of Fake review detection

5.3.4. Implementation Word Embedding

For this study's contextual feature representation mechanism, word embedding was applied. Pretrained word embedding is a way to create an embedding vector with closely related and comparable words represented in the vector space. We use the pre-trained word embedding as a glove, which also serves as a computationally effective predictive model for extracting word embedding from unprocessed text. We used a python gensim package, which is used to implement various embedding techniques, to implement the glove.

```
#Word embedding
from numpy import array
from numpy import asarray
from numpy import zeros
#create Embedding dictionary
embeddings_dict = dict()
#load the pretrained word embedding
glove_path = open('C:/Users/Success/Desktop/Thesis/glove.6B.100d.txt',  "r", encoding='utf-8')
#vectorize the file
for line in glove_path:
    records = line.split()
    word = records[0]
    vector_dim = asarray(records[1:], dtype='float32')
    embeddings_dict[word] = vector_dim

glove_path.close()
#create the matrix of embedding
embedding_matrix = zeros((vocab_size, 100))
for word, index in tokenizer.word_index.items():
    embedding_vector = embeddings_dict.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector
```

Figure 5-10: Glove Word embedding of Fake review detection

5.3.5. Implementation of Feature Scaling

We employed standardization feature scaling to more closely approximate a behavioral feature with good statistical features. The outcome of normalized data is a value that ranges from mean zero to variation one.

```
#Feature Scaling using Standardization
from sklearn.preprocessing import StandardScaler
#training data for the neural net
def data_normalization(X2_train,X2_valid, X2_test):
    scaler = StandardScaler()
    # fitting the scaler to training data
    scaler1 = scaler.fit(X2_train)
    scaler2 = scaler.fit(X2_valid)
    scaler3 = scaler.fit(X2_test)
    # Transforming training and test data using scaler
    X2_train = scaler1.transform(X2_train)
    X2_test = scaler3.transform(X2_test)
    X2_valid = scaler2.transform(X2_valid)
    return X2_train, X2_valid, X2_test

# Applying normalization to data
X2_train, X2_valid, X2_test = data_normalization(X2_train,X2_valid, X2_test)
```

Figure 5-11: Feature scaling for the statistical feature

5.4. Model Implementation

Detect fake reviews, using our proposed deep learning models such as hybrid deep neural network, and the models are tested using stratified k-fold cross-validation as well as performance measures metrics. Now execute all of the designs explored in this paper because the preprocessed data and an embedding matrix, which the hybrid CNN-BiLSTM model requires to be trained. The proposed method imports a python library package to create a model. To implement the model, all the techniques and necessary python libraries are imported as shown below.

```
import pandas as pd
import numpy as np
import re
from numpy import array
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.layers import GlobalMaxPooling1D
from keras.models import Model
from keras.layers.embeddings import Embedding
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer
from keras.layers.merge import Concatenate
from sklearn.utils import shuffle
from tensorflow.keras import regularizers
from tensorflow.keras.optimizers import Adam
from keras.layers.convolutional import MaxPooling1D, Conv2D, Convolution1D,
MaxPooling2D, AveragePooling1D
from tensorflow.keras.layers import Dense, Activation, Flatten, Input, LSTM,
Conv1D, MaxPool1D, Bidirectional, Dropout
```

Figure 5-12: Python Library of Hybrid of CNN-BILSTM model

5.4.1. Tuning Hyperparameter

Tuning a hyperparameter is critical before the final implementation of the model for a different purpose. It has direct control over the training algorithm's performance. The use of automatic hyperparameter tuning significantly decreases the tiredness of the researcher. After the experiment, we chose the best hyperparameter and recorded it in the table below.

Table 4: Parameters name and value

Parameter name	**Optimum Value**
Learning rate	0.001
Epoch	50
Batch size	256
Activation	Sigmoid, and ReLu
Pooling	MaxPooling
Kernel size	32
Dropout:	0.45
Optimizer	Adam
No filter size	256

input_1 = Input (shape= (maxlen,))
input_2 = Input (shape= (11,))

After preprocessing the review and other attributes, we employed both the input of behavioral and textual features for the model.

Convolutional Neural Network: The parameters of neural network configuration are achieved by a fine-tuning process because the number of epochs to be iterated is not found exactly at once. We used maximum dropout to remove unnecessary bias from the network (0.45). **Bidirectional Long Short-Term Memory-** the LSTM model is another neural network classifier technique we used. We use the Keras **functional API** to handle the models' two different inputs, such as the textual and behavioral characteristics of the reviews.

```
embedding_layer = Embedding(vocab_size, 100, weights=[embedding_matrix])(input_1)
conv3 = Conv1D(filters=256, kernel_size=32, activation='relu')(embedding_layer)
drop3 = Dropout(0.5)(conv3)
pool3 = MaxPooling1D(pool_size=2)(drop3)
LSTM_Layer_1 = Bidirectional(LSTM(256, dropout=0.5, return_sequences=True,
                                  kernel_regularizer=regularizers.l2(0.01)))(pool3)
drop2=Dropout(0.35)(LSTM_Layer_1)
#glob=GlobalMaxPool1D()(drop2)
bat=BatchNormalization()(drop2)
drop3=Dropout(0.5)(bat)
den2=Dense(128, activation = "relu")(drop3)
drop4=Dropout(0.5)(den2)
den3=Dense(64, activation = "relu")(drop4)
drop5=Dropout(0.5)(den3)
# Flattening the layer ( multidimentional data into vector)
flaten2=Flatten()(drop5)
# prevents over-fitting (randomly remove some neurons)
dense_layer_1 = Dense(256, activation='relu')(input_2)
dense_layer_2 = Dense(128, activation='relu')(dense_layer_1)
concat_layer = Concatenate()([flaten2, dense_layer_1])
#Add the layer for both feature
dense_layer_3 = Dense(10)(concat_layer)
drop6=Dropout(0.45)(dense_layer_3)
output = Dense(2, activation='sigmoid')(drop6)
model = Model(inputs=[input_1, input_2], outputs=output)
#compile the model
model.compile(optimizer = Adam(learning_rate=0.001),loss ='binary_crossentropy',
              metrics=['accuracy'])
print(model.summary())
```

Figure 5-13: Implementation code of Hybrid of CNN-BILSTM model

5.4.2. Implementation of Model Evaluation

The model testing and evaluation return a confusion matrix with the corresponding accuracy for each class, holding the performance value in each fold iteration. We used stratified k-fold cross-validation with different k values and various performance metrics such as accuracy, confusion matrix, precision, recall, and f1-score to test our models. The cross_val_score () and cross_val_predict () methods are used to implement stratified k-fold CV. Cross_val_predict returns the predicted label, while cross_val_score determines the average accuracy of the 10 folds.

```
# Fit data to model
history = model.fit(x=[X1_train, X2_train], y=y_train,batch_size=512,
                    epochs=epochs, validation_data=([X1_valid, X2_valid], y_valid), verbose=1)

# Generate generalization metrics
scores = model.evaluate(x=[X1_valid, X2_valid], y=y_valid, verbose=1)
print(f'Score for fold {fold_no}: {model.metrics_names[0]} of {scores[0]}; .{model.metrics_names[1]}
      of {scores[1]*100}%')
acc_per_fold.append(scores[1] * 100)
loss_per_fold.append(scores[0])
```

Figure 5-14: Evaluate the training model

5.4.3. Word Cloud

The word clouds give us an idea of what words represent in each class's dataset, but they don't always provide accurate information, especially when comparing classes. We decided to use word clouds to visualize each polarity with word clouds just to get an idea of what our final training data would look like. The word clouds for three of the polarities are shown below and the result is displayed on APPENDIX F.

```
real_review = review_df.review_Text
real_titles_ls = [text for text in real_review]
# print(alls)
real_all_words = ' '.join(map(str, real_review))
wordcloud_real = WordCloud(background_color='white',
    width= 800, height= 500,
    max_font_size = 180,
    collocations = False).generate(real_all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud_real, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Figure 5-15: Word cloud implementation code

CHAPTER FIVE

6. RESULT AND DISCUSSION

6.1. Introduction

This chapter discusses the findings of an experiment utilizing deep neural network techniques to detect fake reviews. This study utilizes various deep learning techniques such as CNN, BILSTM, CNN-MLP, CNN-GRU, CNN-LSTM, and CNN-BILSTM, implemented using the TensorFlow, Keras, and sklearn python libraries. Finally, we discuss the hyperparameter tuning and outcomes of each experiment in this study.

6.2. Model Performance Results

6.2.1. Stratified K-Fold Cross Validation Results with K-Values

In this study, we used a preprocessed, labeled dataset to test the performance of the proposed model. Accuracy, precision, recall, and F1 score values of various performance measures were used to evaluate the performance of the proposed model. We built six different deep learning models for our experiments and evaluated the models using benchmarking metrics. To evaluate the performance of each model under both 5-fold and 10-fold tests, we conducted several time series experiments to determine the mean accuracy through 5-fold and 10-fold sequential cross-validation.

Experiment 1:

Table 5: Model Test Results Using 5-Fold Cross Validation

Model Test Results Using Stratified 5-Fold Cross Validation				
Techniques	Performance Evaluation			
	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
CNN	87%	91%	89%	88.99%
MLP	76%	78%	75%	75.12%
BILSTM	90%	93%	92%	90.91%
CNN-MLP	87%	86%	86%	87.06%
CNN-LSTM	91%	92%	91%	92.01%
CNN-GRU	91%	91%	92%	92.35%
CNN-BiLSTM	95%	95%	95%	95.28%

The review dataset that has been divided into five folds iterates to make the prediction when k = 5, designating each fold as a test and the remaining folds as training. According to the results of this experiment, the model works well for classes of fake and genuine reviews. An accuracy of **95.28**% is provided by the model's overall performance, which involves doing the k-fold 5 times.

Table 6: Model Test Results Using 10-Fold Cross Validation

Model Test results using Stratified 10-Fold Cross-Validation				
Techniques	**Performance Evaluation**			
	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
CNN	91%	90%	90%	89.07%
MLP	78%	78%	77%	77.35%
BILSTM	92%	93%	90%	92.35%
CNN-MLP	89%	88%	88%	88.79%
CNN-LSTM	92%	91%	93%	92.12%
CNN-GRU	94%	92%	93%	93.86%
CNN-BiLSTM	**96%**	**95%**	**95%**	**95.72%**

When k = 10, the test set is switched between each fold in each evaluation loop, just like in other stratified k-fold cross-validations. The results of the experiment show that utilizing 10-fold cross-validation is better than using 5-fold for our case. The CNN-BiLSTM performs best in the 10-fold CV test, with an accuracy of 95.7 and an F1 score of 95. Each fold's evaluation is combined, and the average is taken into a model that works best for classes of fake and genuine reviews. CNN-GRU is the second-best performing model, with an accuracy of 93.86 and an F1 score of 93. Finally, it is demonstrated that the accuracy of the average accuracy model, which includes doing the stratified k-fold ten times, is 95.72%.

6.2.2. Model evaluation with metrics

With varied train-test ratio values, the proposed hybrid deep neural network of the CNN-BILSTM model produced different results. To achieve the best results for fake review identification, we experimented with different train test ratios. The accuracy performance of the proposed model

with binary class labels in terms of training and testing datasets is quite impressive. Generally, we analyze the performance of deep learning methods to compare the performance of each method.

Convolutional Neural Networks (CNN) is one of the models we employed in this research; they produce results for each class, namely fake and genuine reviews. We used the CNN model alone before hybrid with other techniques to compare performance in our dataset and evaluate it with performance metrics.

Experiment 2:

Table 7: CNN overall performance evaluation

Class of Review	**Evaluation Metrics for CNN model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.87	0.89	0.88	
Genuine	0.90	0.91	0.90	
Overall	0.88	0.90	0.89	
				89.04%

The confusion matrix shown in Figure 6.1 demonstrates that out of the 39,173 testing samples in the dataset, the CNN model properly identifies 17,301 occurrences, with just 2,275 cases being misclassified into the genuine class. 11 percent of the dataset's 2,074 fake cases are labeled as false negatives, whereas 17,503 reviews are real positive fake reviews. However, for the class Genuine, 12% of the cases in total are categorized as false negatives, while the remaining 88 % of instances in the class are correctly identified. And 11% of the cases in the fake review class are classified as false negatives, while the remaining 89% of reviews are classified correctly as genuine.

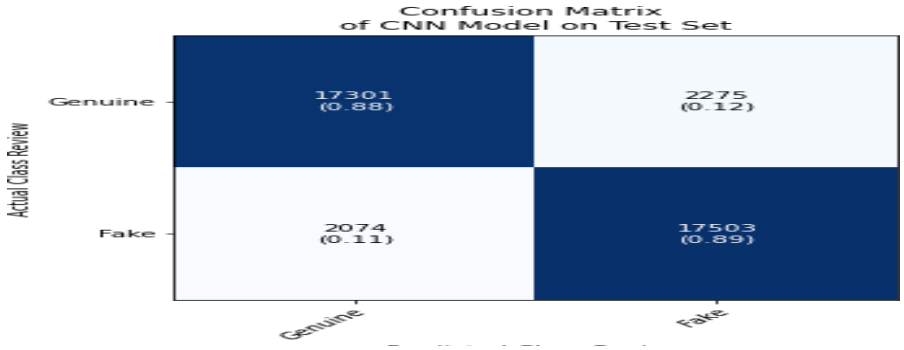

Figure 6-1; Confusion Matrix for CNN model

Multi-Layer Perceptron (MLP) is one of the deep learning techniques we employed in this research; they produce results for each class, namely fake and genuine reviews. We used the MLP model alone before hybrid with CNN techniques to compare performance in our dataset and evaluate it with performance metrics.

Table 8: MLP overall performance evaluation

Class of Review	**Evaluation Metrics for MLP model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.70	0.72	0.70	
Genuine	0.71	0.71	0.72	
Overall	0.72	0.70	0.73	
				72.64%

The confusion matrix shown in Figure 6.2 demonstrates that out of the 39,173 testing samples in the dataset, the MLP model properly identifies 14,104 occurrences, with just 5,472 cases being misclassified into the genuine class. 25 percent of the dataset's 6,157 fake cases are labeled as false negatives, whereas 13,420 reviews are real positive fake reviews. However, for the class Genuine, 28% of the cases in total are categorized as false negatives, while the remaining 72 % of instances in the class are correctly identified. And 31% of the cases in the fake review class are classified as false negatives, while the remaining 69% of reviews are classified correctly as genuine.

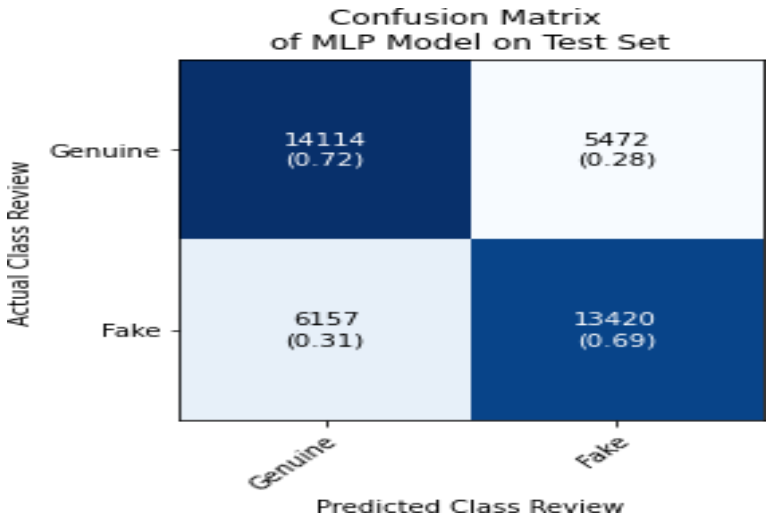

Figure 6-2: Confusion Matrix for MLP model

Bidirectional Long Short Term Memory (BILSTM) – is an RNN technique that employs the bidirectional LSTM algorithm. The second method we used to identify fake reviews was called Bidirectional Long Short Term Memory (BILSTM), which performed more accurately than the CNN model in our dataset and parameters.

Table 9: BILSTM overall performance evaluation

Class of Review	**Evaluation Metrics for the BILSTM model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.91	0.89	0.90	
Genuine	0.92	0.94	0.91	
Overall	0.91	0.92	0.90	
				91.85%

Figure 6.3 demonstrates that out of the 39,173 testing samples in the dataset, the BiLSTM model properly identifies 18,001 occurrences, with just 1,540 cases being misclassified into the genuine class. From the testing data, 1860 fake cases are labeled as false negatives, whereas 17126 reviews are real positive fake review classes. However, for the genuine class, 8% of the cases in total are categorized as false negatives, while the remaining 92 % of instances in the class are correctly identified. And 10% of the cases in the fake review class are classified as false negative, while the remaining 90% of reviews are classified correctly as genuine. Finally, the BILSTM model was performed with 91.85% accuracy in the electronics review dataset.

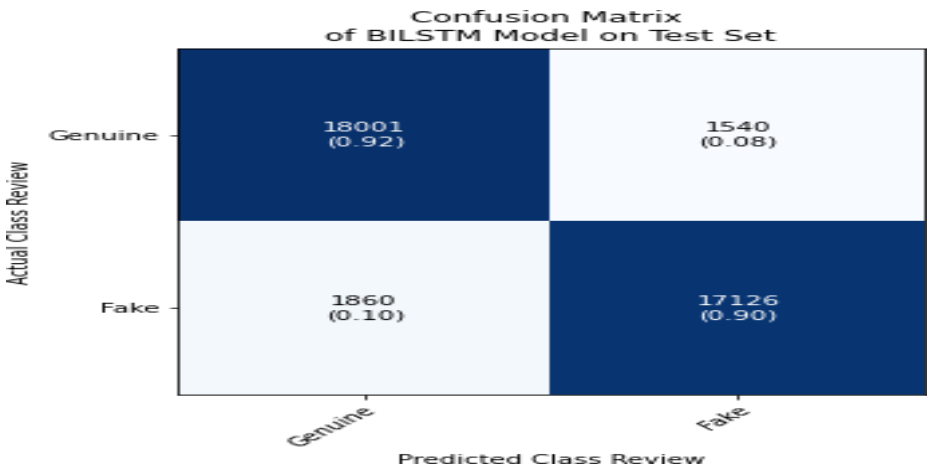

Figure 6-3: Confusion Matrix for BILSTM model

CNN-Multi-Layer Perceptron (CNN-MLP)- We build multi-layer perceptron deep learning with CNN techniques for classifying the review using both contextual and behavioral attributes of the review. MLP has multiple layers that connected all layers and dense hidden layers because of that the model accuracy may be a low-performance model and integrated with CNN techniques.

Table 10: CNN- MLP overall performance evaluation

Class of Review	**Evaluation Metrics for CNN- MLP model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.83	0.84	0.85	
Genuine	0.87	0.89	0.89	
Overall	0.85	0.87	0.87	
				86.39%

From table 6.5, the hybrid of CNN and multi-perceptron model performs overall with 85% precision, 87% recall, and 87% f1-score. In addition, from the confusion matrix 6.1, the algorithm correctly identified 17,036 reviews as real positive evaluations from the testing instances, whereas 2,540 reviews were identified as genuine negative reviews. The model predicted the correct 16,508 reviews were marked as fake reviews, with 3,078 fake negative reviews. In both classes, the model performed with an average accuracy of 86.36%.

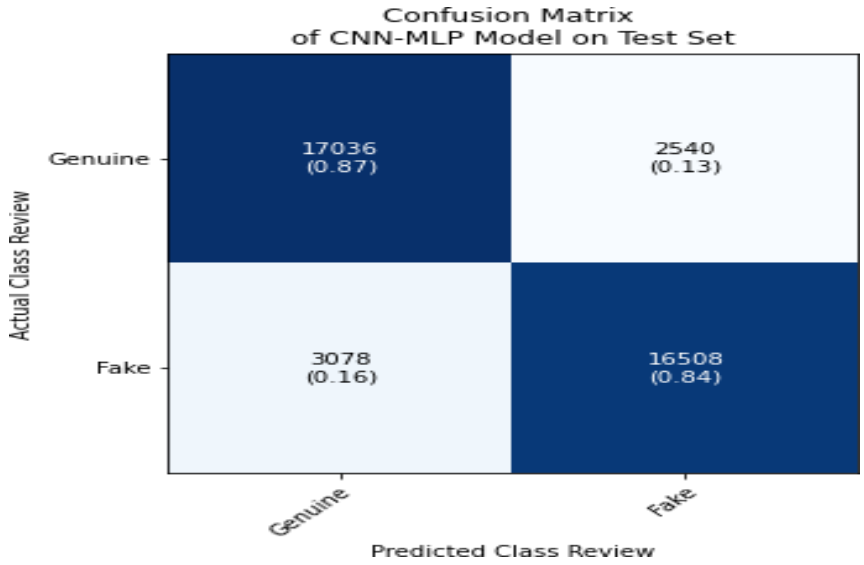

Figure 6-4: Confusion Matrix for CNN - MLP model

CNN-Gated Recurrent Units (CNN-GRU): GRU has one type of RNN branch that has good performance in deep learning and, in our case, also better performance than the CNN-MLP hybrid model with the integration of CNN techniques.

Table 11: CNN - GRU overall performance evaluation

Class of Review	**Evaluation Metrics for CNN-GRU model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.90	0.89	0.91	
Genuine	0.97	0.96	0.98	
Overall	0.94	0.93	0.95	
				94.02%

The GRU and CNN hybrid models outperform the good results of the CNN-MLP hybrid models. From the confusion matrix figure 6.3, the model performed accurately 18,768 genuine reviews, and 859 genuine negative reviews from testing instances. And there are 18,326 instances of fake positive reviews with misclassification of 1,660 negative fake reviews. Table 6.6 results show 94% of precision, 93% of recall, and 95% F1-score are performed with this model. This model performed better than the CNN, BILSTM, and CNN-MLP with the true class, but it performed poorly with the fake class, achieving 94.02% accuracy.

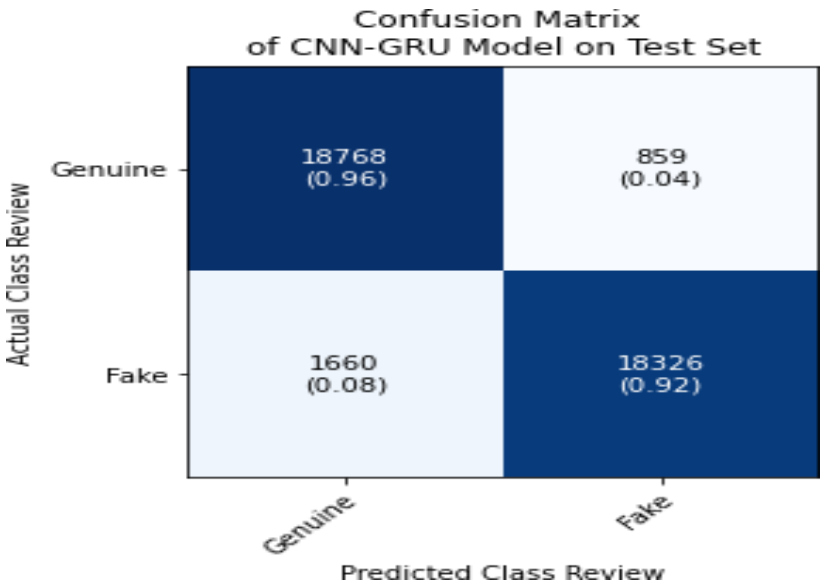

Figure 6-5: Confusion Matrix for CNN-GRU model

CNN-Long Short-Term Memory (CNN-LSTM): We test the performance of the techniques with the evaluation metrics. We analyze both the individual and overall accuracy of the models.

Table 12: CNN-LSTM overall performance evaluation

Class of Review	**Evaluation Metrics for CNN-LSTM model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.92	0.93	0.93	
Genuine	0.95	0.94	0.92	
Overall	**0.93**	**0.94**	**0.95**	
				94.81%

The other model we created used CNN-LSTM hybrid techniques and performed well on our review dataset. The model classified 95% of the testing data 18,568 genuine instances in the datasetas true positives, while 1,059 instances misclassify as a false class and only 5% of them were classified as false negatives. And for the fake class, out of a total of 19,586 instances, 93% of themare classified as true positives and 1,460 instances misclassify as are false negatives class. And 94.81% accuracy was performed with both classes in review detection.

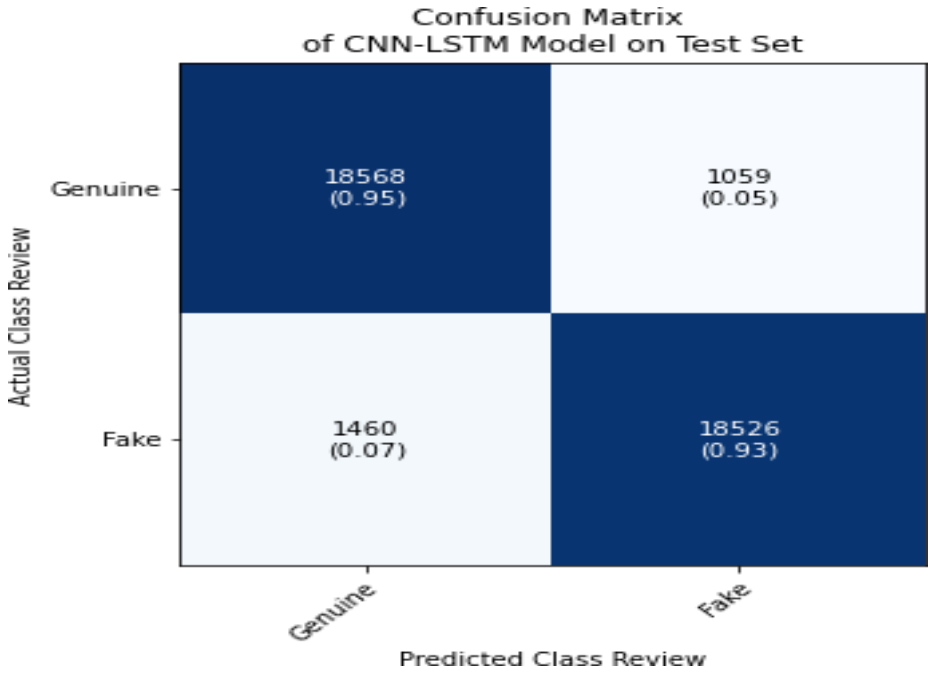

Figure 6-6: Confusion Matrix for LSTM model

Hybrid deep neural network CNN-BILSTM model – the CNN and BILSTM model perform good results than other deep learning methods we used for this study when we compare it.

Table 13: CNN-BILSTM overall performance evaluation

Class of Review	**Evaluation Metrics for CNN- BILSTM model**			
	Precision	Recall	F1-Score	Accuracy
Fake	0.96	0.98	0.98	
Genuine	0.98	0.99	0.97	
Overall	0.97	0.98	0.97	
				96.7%

Then, we applied the CNN and BILSTM hybrid techniques to fake review detection to outperform 97% precision, 98% recall, and 97% F1-score. Of the total testing instances, 19,054 are correctly classified as true guanine and only 659 are misclassified as false genuine. And 18,705 instances are accurately classified as positive fake reviews, while 70 instances are classified as false fake reviews. In the dataset, 97% are classified correctly, and the remaining 3% are false negatives. But, for fake instances, only 96% are true positives, and the rest 4% are false negatives, as classified by the CNN-BILSTM model.

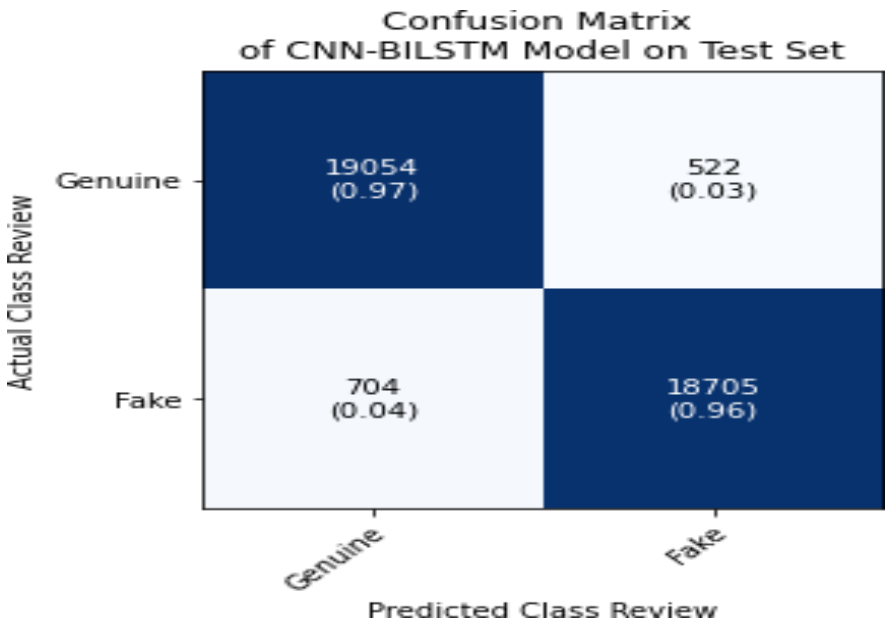

Figure 6-7: Confusion Matrix for CNN-BILSTM model

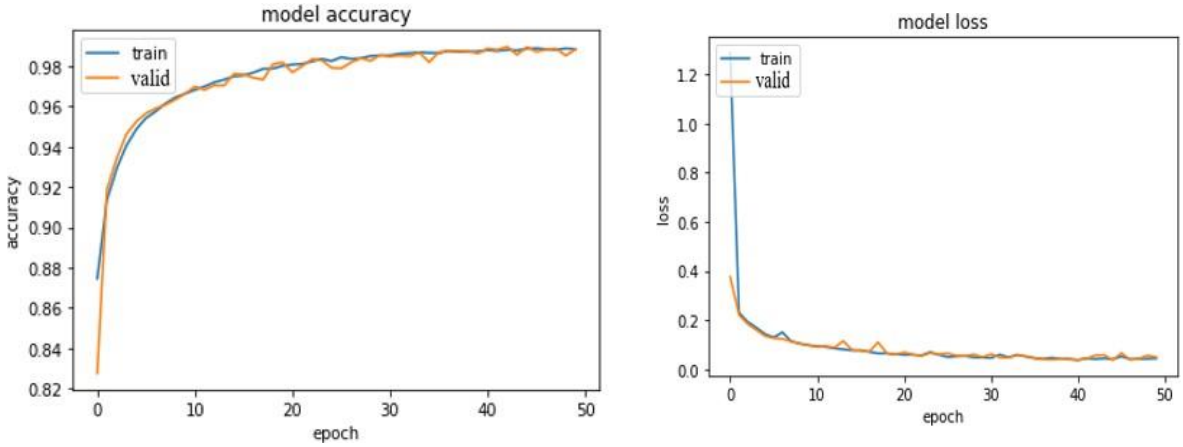

Figure 6-8: Hybrid CNN-BILSTM Plot learning curve

Experiment 3: Compare the model using the sentiment analysis feature as the review attribute to check the effectiveness of the model by doing both with and without.

Table 14: Effective sentimental analysis of the proposed model

Proposed Hybrid Model	Performance Evaluations Metrics			
	Precision %	Recall %	F1-score %	Accuracy %
Without sentimental analysis	91%	90%	92%	**91.50%**
With sentimental analysis	97%	96%	97%	**96.70%**

Finally, we ran an experiment with the sentimental analysis feature to see how much accuracy our model gained with and without the sentimental analysis feature.

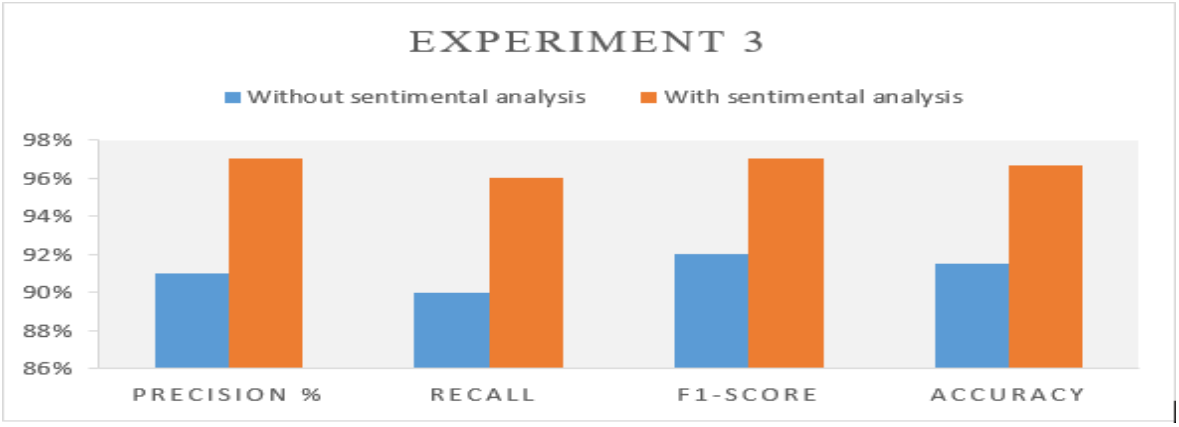

Figure 6-9: Effective of sentimental analysis plots

The model without sentimental analysis performed 91% of precision, 90% of recall, 92% of F1-score, and 91.50% of accuracy. The experiment shows the model with sentimental analysis performs 96.70% accuracy with review datasets and adds 5.20% to the model. To get the best measure of train test data size, we perform four ratios of the dataset and apply the best ratio for our mode.

Experiment 4: We also test the review with the textual feature of reviews and behavioral features of individuals to conclude which feature is more accurate in this model.

Table 15: Proposed model with contextual and behavioral features

Ratio of train and test	Metrics evaluation with textual feature				Metrics evaluation with behavioral feature			
	Precision	Recall	F1score	Accuracy	Precision	Recall	F1score	Accuracy
40/60	0.52	0.50	0.54	0.534	0.87	0.90	0.91	0.908
30/70	0.59	0.60	0.61	0.601	0.91	0.93	0.93	0.922
20/80	0.63	0.62	0.65	0.644	0.93	0.95	0.94	0.942
10/90	0.65	0.67	0.64	0.651	0.94	0.95	0.94	0.945

To test CNN-BILSTM for precision, recall, F1 score, and accuracy, we obtained the findings in table 11. In the table, various train test ratios 40/60, 30/70, 20/80, and 10/90 were employed to evaluate how well the model performed in terms of both the textual and behavioral aspects of the reviews. Following the testing, the 80:10:10 ratio was chosen since it produced better outcomes than the other ratios for training, validation, and testing. Comparing training with behavioral features to contextual review features yields good results.

6.2.3. Tuning Hyper-parameters

We have carefully selected hyperparameter to fully utilize the capabilities of our model. For the deep learning algorithm, hyperparameter adjustment is crucial for a variety of reasons, and the effectiveness of the training algorithm is directly under its control. The grid search technique is one of several strategies for fine-tuning hyperparameter and requires computing time since it

explores all possible hyperparameter combinations. As a result, we have used the standard rule of thumps to support our manual hyperparameter tuning technique.

Experiment 5: To address the issue, we use a model with manual tuning and low simplification errors, as well as short-term and low memory costs.

Table 16: Hyperparameter has been chosen for tuning

#No. of Trial	Learning rate	Activation function	Kernel size	No. of Epochs	Batch size	Dropout
1	0.001	Relu	8	50	128	0.5
2	0.001	Softmax	32	30	256	0.45
3	0.01	Relu	16	20	64	0.35
4	0.0001	Relu	5	10	128	0.25
5	0.0001	Softmax	13	30	256	0.15
6	0.0001	Relu	4	20	32	0.35
7	0.001	Softmax	8	5	256	0.25
8	0.001	Relu	32	50	256	0.45
9	0.01	Relu	8	20	256	0.20
10	0.001	Softmax	32	15	128	0.35

Depending on the hyperparameter tunning we perform different experiments were tested using the above table, and the shaded row with the best results was chosen (X. Zhang et al., 2019).

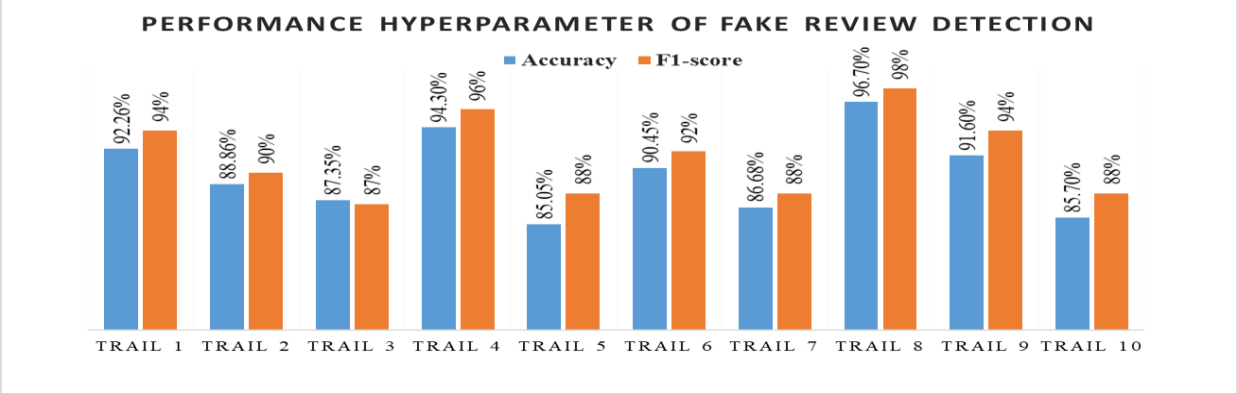

Figure 6-10: Performance of each hyperparameter

We chose our CNN-BILSTM model and trained it. The number of filters and learning rate has a significant impact on updating the network's weights, which affects the model's performance. To achieve the best results, the number of filters and learning rate should be adjusted to 256 and 0.001, respectively. The ideal set of hyperparameter is searched after in this demonstration. The overall accuracy and average F1 score of each class are displayed below as the outcome of tuning for hyperparameter.

Experiment 6: We conduct comparative study on the word embedding such as FastText, Word2vec and glove to get good performance on the fake review detection model. Use the pretrained word embedding and test the performance with the proposed model to detect the fake reviews.

Table 17: Effective word embedding on the proposed model

Word Embedding	Performance Evaluations Metrics			
	Precision %	Recall %	F1-score %	Accuracy %
Fast Text	89%	90%	89%	**90.50%**
Word2vec	95%	94%	95%	**95.10%**
glove	97%	96%	97%	**96.70%**

From the experiment we conduct the glove has better result according the performance evaluation we did in this experiments.

6.3. Discussion

The technique of identifying fake reviews requires extracting useful information from user opinions in an intelligent manner to understand the thoughts and feelings of a target audience. It is challenging to manage people's emotions or opinions as a result of the content on social networking sites. The study is to design a hybrid deep learning-based fake review identification model. We used six deep learning techniques like CNN, MLP, BILSTM, CNN-MLP, CNN-GRU, CNN-LSTM, and hybrid CNN-BILSTM neural network techniques. We have accomplished several tasks to achieve this goal.

We did five experiments in this study such as: The first experiment, was cross-validation with 5 and 10 folds cross-validation to measure the performance of proposed techniques. Experiment one shows the 10-fold stratified has better results than the five k-fold cross-validation evaluation. The second experiment was on selected techniques to choose an outperformed model from the six deep neural networks discussed in the proposed model. The experiments also show the hybrid of CNN and the BILSTM model performs better than other techniques used in this study. We answered the RQ3 with the experiment one and two specifically the CNN-BILSTM has good performance.

The third experiment was on the sentimental analysis with fake review detection as an attribute to increase the accuracy of the model, and the result also performed well when using sentiments. The third experiment also answered the RQ1 by applying the sentiment and polarity as a behavioral feature. The fourth experiment in this study compares the textual and behavioral features' performances without integrating them. The experiment also shows the behavioral attribute has a more promising performance than the contextual feature model. The fourth experiment also answered the RQ2 by compare the textual and behavioral feature. The fifth experiment on hyperparameter tuning aims to find an optimum value for the model by varying the values.

Based on the performance evaluation, the hybrid CNN-BiLSTM technique achieves the highest accuracy. To evaluate the performance of each technique, we used cross-validation and performance evaluation metrics such as accuracy, F1-score, precession, and recall. We use both behavioral and textual feature evaluation for fake review detection. The first model we used is the CNN technique, which achieved an accuracy of 89.04%, an F1 score of 89%, precision of 88%, and a recall of 90% using both textual and behavioral features of product reviews. The second model is BILSTM techniques, which have an accuracy of 91.85%, an F1-score of 90%, a recall of 92%, and a precision of 91%.

The third model we used, a hybrid of CNN and MLP models, achieved an accuracy of 86.39%, an F1 score of 87%, a precision of 85%, and a recall of 87% by using both the textual and behavioral features of product reviews. The fourth model we used, a hybrid of CNN and GRU models, achieved an accuracy of 94.02%, an F1 score of 95%, a precision of 94%, and a recall of 93% by using both textual and behavioral feature techniques. The fifth hybrid model of CNN and LSTM techniques achieved an accuracy of 94.81%, an F1 score of 95%, a precision of 93%, and a recall

of 94% by using a textual and behavioral features of product reviews. Finally, we discuss the result of our proposed hybrid model of CNN-BiLSTM techniques achieved an accuracy of 96.70%, an F1 score of 97%, a precision of 97%, and a recall of 96% by using textual and behavioral feature techniques. The summary of this performance evaluation is described in the table.

Table 18: Summary of Performance Evaluation

Techniques	Performance Evaluation Metrics			
	Precision	**Recall**	**F1-Score**	**Accuracy**
CNN	88%	90%	89%	89.04%
MLP	72%	71%	70%	72.46%
BILSTM	91%	92%	90%	91.85%
CNN-MLP	85%	87%	87%	86.39%
CNN-LSTM	93%	94%	95%	**94.81%**
CNN-GRU	94%	93%	95%	94.02%
CNN-BiLSTM	97%	96%	97%	**96.70%**

We used Pos tagging and sentimental analysis to get good performance accuracy. Apply different scientific experiments to get the optimum feature of review and plot the feature correlation between the behavioral features. We got the behavioral feature to perform better than the textual feature to classify the review. To improve the performance of the review detection, we used the combination of multiple inputs with different features to combine the review attributes using Keras API functional methods.

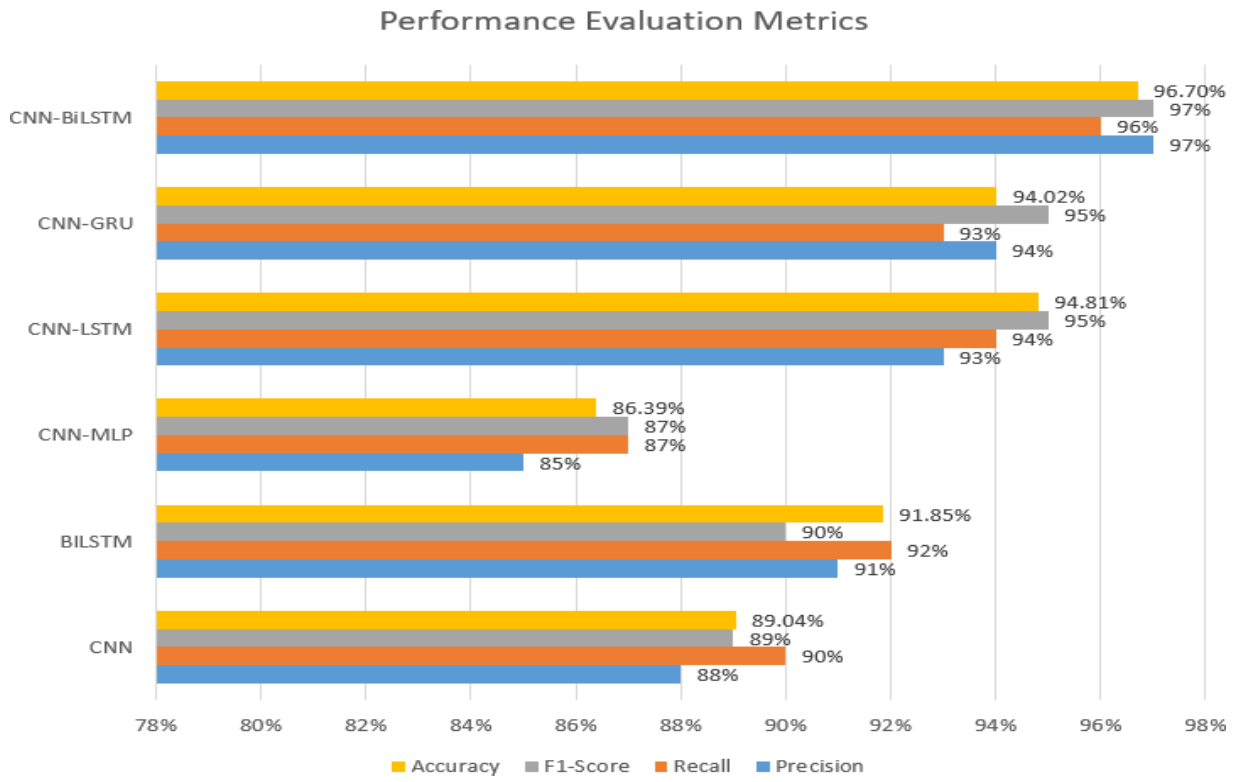

Figure 6-11: Performance evaluation metrics plots

Finally, the CNN-BiLSTM model continued to deliver the best results for the suggested false review detection, with an accuracy of 96.70% and an F1 score of 98%. Using the CNN-BiLSTM model, the prototype is created. Our prototype's GUI was created, and we used FLASK API to deploy the model on a web server.

CONCLUSION AND FUTURE WORK

The fake review detection using hybrid neural networks in the case of the electronics retailer model and findings are concluded in the conclusion chapter. It also discusses the fake review detection recommendations and future work directions.

Conclusion

This study aims to develop a solution to detect fake reviews in electronics retailers using hybrid neural network learning techniques. Customer decisions and businesses are heavily influenced by online user reviews. Reviews assist in the decision-making process for purchasing a specific product and are misled by fake reviews when making purchasing decisions. On e-commerce sites, fake reviews are used to promote or demote product services. It can damage a company's reputation as a good service or product provider, resulting in financial loss. Fake reviews can sometimes lead to financial gain for a company or firm. Customers and businesses are both harmed by fake reviews. This study attempted to develop, implement, and compare deep learning textual and behavioral feature methods specifically for fake review detection.

The study found detection results on an electronic fake review dataset using textual and behavioral features. Our findings point to some previously unexplored opportunities in the fake review detection field, such as the use of behavioral and contextual features to improve the classification model for detecting fake reviews across domains. In terms of efficiency and accuracy, the proposed models outperformed existing state-of-the-art and baseline approaches to fake review detection methods. The experimental results also showed that the proposed hybrid models with combined sentimental polarity, subjectivity, Pos tagging and behavioral features were the most effective for larger datasets, implying their potential applications in real-life scenarios. We conduct experimental on pre-trained word embedding and use the GloVe model 100-dimensional vectors with textual features of the similarity of words and representations to achieve high accuracy in such a setting.

As discussed in chapter two, understanding and defining fake review detection and methods, exploring and understanding various existing techniques used to address the problem, and understanding the model were all necessary for the study to be successful. Furthermore, the various methods used to implement and design models are capable of detecting fake reviews. These methods include loading datasets, preprocessing, feature extraction using word embedding’s like

a glove, training the model using CNN, MLP, BILSTM, CNN-MLP, CNN-GRU, and CNN-LSTM techniques, and testing each model with performance measurement.

Based on the evaluation metric results and cross-validation, we compare the models. These electronic datasets with textual and behavioral features were used in the experiment. Textual feature models based on CNN-BILSTM with glove word embedding performed better than CNN, MLP, BILSTM, CNN-MLP, CNN-GRU, and CNN-LSTM in terms of accuracy. Furthermore, the approach we recommend gives the user the ability to recommend the most accurate reviews so that the buyer can make an informed decision about the product. And the use of various factors, such as adding new vectors such as network features, time features, and emoji have improved review detection accuracy.

Contribution of the study

- The study, outperforms state-of-the-art approaches in performance accuracy, highlighting the hybrid model for the detection of fake reviews.
- A fake review detection model can integrate the sentimental analysis with other review attributes to get good results and compare the proposed approach with state-of-the-art techniques.
- Another contribution is that we created a novel hybrid architecture for fake reviewdetection using the Keras API, which includes both behavioral and textual features as input. We
- provide a way for other researchers to focus on this area to develop with other online business sectors, and the results of this article can also be used as input for an additional researcher for online business reviews in the context of our country.

Future Work

As a result of the review domain's scope, we recommend the left task and method for improving the model's performance. In general, there are many tasks left to cover all the domains of the review model that address the entire problem of fake review detection. As a result, we recommend the tasks below for future work.

- Future researchers will continue their research in the direction of fake review detection strategies that can employ transformers, such as Berta's pretrained model.
- Focus to use the feature in future work to detect phishing attacks, which are a type of social engineering.
- Concentrate on real-world applications such as unlabeled data, using unsupervised machine learning techniques.
- Look for a more effective and accurate detection method that can detect fake information in a variety of fields, including group spammers and rumors, in the future.

SPECIAL ACKNOWLEDGMENTS

This research was funded by Adama Science and Technology University with grant number**:**

ASTU/ SM-R/485/22

Adama, Ethiopa

References

Abdolrasol, M. G. M., Suhail Hussain, S. M., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S., & Milad, A. (2021). Artificial neural networks based optimization techniques: A review. *Electronics (Switzerland)*, *10*(21). https://doi.org/10.3390/electronics10212689

Almeida, F., & Xexéo, G. (2019). *Word Embeddings: A Survey*. *1991*. http://arxiv.org/abs/1901.09069

Anusha P, P. K. L. (2020). Survey on Fake Online Reviews Using Machine Learning Algorithms. *Journal of Critical Reviews*, *7*(18), 2752–2758.

Barbado, R., Araque, O., & Iglesias, C. A. (2019). A framework for fake review detection in online consumer electronics retailers. *Information Processing and Management*, *56*(4), 1234–1244. https://doi.org/10.1016/j.ipm.2019.03.002

Bhuvaneshwari, P., Rao, A. N., & Robinson, Y. H. (2021). *Spam review detection using self attention based CNN and bi-directional LSTM*. 18107–18124.

Borowska, K., & Topczewska, M. (2014). *ADVANCES IN COMPUTER SCIENCE RESEARCH DATA PREPROCESSING IN THE CLASSIFICATION OF THE IMBALANCED DATA*. *11*, 31–46.

Brechon, S. (2014). Learning to trust consumer review and comparison sites as a marketing tool. *Journal of Aesthetic Nursing*, *3*(1), 42–46. https://doi.org/10.12968/joan.2014.3.1.42

Brownlee, J. (2017). Deep Learning for Natural Language Processing : Develop Deep Learning Models for Natural Language in Python. *Machine Learning Mastery*, 414.

Buda, M. (2017). *A systematic study of the class imbalance problem in convolutional neural networks SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION A systematic study of the class imbalance problem in convolutional neural networks*. 49.

Cao, C., Li, S., Yu, S., & Chen, Z. (2021). Fake Reviewer Group Detection in Online Review Systems. *IEEE International Conference on Data Mining Workshops, ICDMW*, *2021-Decem*, 935–942. https://doi.org/10.1109/ICDMW53433.2021.00122

Cardoso, E. F., Silva, R. M., & Almeida, T. A. (2018a). Towards automatic filtering of fake reviews. *Neurocomputing*, *309*, 106–116. https://doi.org/10.1016/j.neucom.2018.04.074

Cardoso, E. F., Silva, R. M., & Almeida, T. A. (2018b). Towards automatic filtering of fake reviews. *Neurocomputing*, *309*, 106–116. https://doi.org/10.1016/j.neucom.2018.04.074

Chen, Y., & Xie, J. (2008). Online consumer review: Word-of-mouth as a new element of marketing communication mix. *Management Science*, *54*(3), 477–491. https://doi.org/10.1287/mnsc.1070.0810

Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00561-y

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2018). *Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction*. 1–11. http://arxiv.org/abs/1801.02143

Delbeto, A. B. (2018). Word Sequence Prediction for Afaan Oromo. *A Thesis Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science Addis Ababa , Ethiopia, March*.

Deng, H., Zhao, L., Luo, N., Liu, Y., Guo, G., Wang, X., Tan, Z., Wang, S., & Zhou, F. (2018). Semi-supervised learning based fake review detection. *Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017*, 1278–1280. https://doi.org/10.1109/ISPA/IUCC.2017.00195

Dhamani, N., Azunre, P., Gleason, J. L., Corcoran, C., Honke, G., Kramer, S., & Morgan, J. (2019). *Using Deep Networks and Transfer Learning to Address Disinformation*. http://arxiv.org/abs/1905.10412

Dong, L. yu, Ji, S. juan, Zhang, C. jin, Zhang, Q., Chiu, D. K. W., Qiu, L. qing, & Li, D. (2018). An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Systems with Applications*, *114*, 210–223. https://doi.org/10.1016/j.eswa.2018.07.005

Et. al., D. V. (2021). Online Products Fake Reviews Detection System Using Machine Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(1S), 29–39. https://doi.org/10.17762/turcomat.v12i1s.1548

Gautam, H. (2020). *Word Embedding: BasicsNo Title*. Mar 1, 2020. https://medium.com/@hari4om/word-embedding-d816f643140

Goyal, K. (2021). *Data Preprocessing in Machine Learning: 7 Easy Steps To Follow*. UpGrad.

Gutierrez-Espinoza, L., Abri, F., Namin, A. S., Jones, K. S., & Sears, D. R. W. (2020). *Fake Reviews Detection through Ensemble Learning*. 1–8. http://arxiv.org/abs/2006.07912

Hai, Z., Zhao, P., Cheng, P., Yang, P., Li, X. L., & Li, G. (2016). Deceptive review spam detection via exploiting task relatedness and unlabeled data. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1817–1826. https://doi.org/10.18653/v1/d16-1187

Hajek, P., Barushka, A., & Munk, M. (2020). Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, *32*(23), 17259–17274. https://doi.org/10.1007/s00521-020-04757-2

He, S., Hollenbeck, B., & Proserpio, D. (2020). The Market for Fake Reviews. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3664992

Hussain, N., Mirza, H. T., Ali, A., Iqbal, F., Hussain, I., & Kaleem, M. (2021). Spammer group detection and diversification of customers' reviews. *PeerJ Computer Science*, *7*, 1–31. https://doi.org/10.7717/PEERJ-CS.472

Hussain, N., Turab Mirza, H., Hussain, I., Iqbal, F., & Memon, I. (2020). Spam Review Detection Using the Linguistic and Spammer Behavioral Methods. *IEEE Access*, *8*, 53801–53816. https://doi.org/10.1109/ACCESS.2020.2979226

Jain, N., Kumar, A., Singh, S., Singh, C., & Tripathi, S. (2019). Deceptive Reviews Detection Using Deep Learning Techniques. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11608 LNCS*. Springer International Publishing. https://doi.org/10.1007/978-3-030-23281-8_7

Jeffrey, P; Richard, S. C. D. (2014). GloVe: Global Vectors for Word Representation Jeffrey. *Empirical Methods in Natural Language Processing (EMNLP)*, *31*(6), 1532--1543. https://doi.org/10.1080/02688697.2017.1354122

Jindal, N., & Liu, B. (2007). Review spam detection. *16th International World Wide Web Conference, WWW2007*, 1189–1190. https://doi.org/10.1145/1242572.1242759

Jonathan Marciano. (2021). *Fake online reviews cost $152 billion a year. Here's how e-commerce sites can stop them*. 2021. https://www.weforum.org/agenda/2021/08/fake-online-reviews-are-a-152-billion-problem-heres-how-to-silence-them/

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, *2*, 427–431. https://doi.org/10.18653/v1/e17-2068

Khurshid, F., Zhu, Y., Xu, Z., Ahmad, M., & Ahmad, M. (2018). Enactment of ensemble

learning for review spam detection on selected features. *International Journal of Computational Intelligence Systems*, *12*(1), 387–394. https://doi.org/10.2991/ijcis.2018.125905655

Khurshid, F., Zhu, Y., Yohannese, C. W., & Iqbal, M. (2017). Recital of supervised learning on review spam detection: An empirical analysis. *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017*, *2018-Janua*, 1–6. https://doi.org/10.1109/ISKE.2017.8258755

Kumar, J. (2020). *Fake Review Detection Using Behavioral and Contextual Features*. http://arxiv.org/abs/2003.00807

Lau, R. Y. K., Liao, S. Y., Chi-Wai Kwok, R., Xu, K., Xia, Y., & Li, Y. (2011). Text mining and probabilistic language modeling for online review spam detection. *ACM Transactions on Management Information Systems*, *2*(4), 1–30. https://doi.org/10.1145/2070710.2070716

Lee, K. D. (2016). *2016.06-*이경엽*.Pdf*.

Li, B., Liu, T., Zhao, Z., Wang, P., & Du, X. (2017). Neural bag-of-ngrams. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 3067–3074.

Li, H., Chen, Z., Liu, B., Wei, X., & Shao, J. (2014). Spotting Fake Reviews via Collective Positive-Unlabeled Learning. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *2015-Janua*(January), 899–904. https://doi.org/10.1109/ICDM.2014.47

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a general rule for identifying deceptive opinion spam. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, *1*, 1566–1576. https://doi.org/10.3115/v1/p14-1147

Li, L., Qin, B., Ren, W., & Liu, T. (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, *254*, 33–41. https://doi.org/10.1016/j.neucom.2016.10.080

Li, L., Ren, W., Qin, B., & Liu, T. (2015). Learning document representation for deceptive opinion spam detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9427*, 393–404. https://doi.org/10.1007/978-3-319-25816-4_32

Liu, W., Jing, W., & Li, Y. (2020). Incorporating feature representation into BiLSTM for deceptive review detection. *Computing*, *102*(3), 701–715. https://doi.org/10.1007/s00607-019-00763-y

Liu, Z., Lei, S. hui, Guo, Y. lang, & Zhou, Z. ang. (2020). *The interaction effect of online review*

language style and product type on consumers' purchase intentions. Palgrave Communications; Springer US. https://doi.org/10.1057/s41599-020-0387-6

Martínez Otero, J. M. (2021). Fake reviews on online platforms: perspectives from the US, UK and EU legislations. *SN Social Sciences*, *1*(7), 1–30. https://doi.org/10.1007/s43545-021-00193-8

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.

Mohawesh, R., Tran, S., Ollington, R., & Xu, S. (2021a). Analysis of concept drift in fake reviews detection. *Expert Systems with Applications*, *169*, 114318. https://doi.org/10.1016/j.eswa.2020.114318

Mohawesh, R., Tran, S., Ollington, R., & Xu, S. (2021b). Analysis of concept drift in fake reviews detection. *Expert Systems with Applications*, *169*(June 2020), 114318. https://doi.org/10.1016/j.eswa.2020.114318

Mohawesh, R., Xu, S., Tran, S. N., Ollington, R., Springer, M., Jararweh, Y., & Maqsood, S. (2021a). Fake Reviews Detection: A Survey. In *IEEE Access* (Vol. 9, pp. 65771–65802). https://doi.org/10.1109/ACCESS.2021.3075573

Mohawesh, R., Xu, S., Tran, S. N., Ollington, R., Springer, M., Jararweh, Y., & Maqsood, S. (2021b). Fake Reviews Detection: A Survey. *IEEE Access*, *9*, 65771–65802. https://doi.org/10.1109/ACCESS.2021.3075573

Moyo, V., & Sibanda, K. (2015). The Generalization Ability of Artificial Neural Networks in Forecasting TCP/IP Traffic Trends: How Much Does the Size of Learning Rate Matter? *International Journal of Computer Science and Application*, *4*(1), 9–17. https://doi.org/10.12783/ijcsa.2015.0401.02

Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, 191–200. https://doi.org/10.1145/2187836.2187863

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

Nigam, A., Tang, P., Dambanemuya, H. K., & Ning, B. (2017). *Text Representation for Machine Learning Applications*.

Noekhah, S., Salim, N. binti, & Zakaria, N. H. (2020). Opinion spam detection: Using multi-iterative graph-based model. *Information Processing and Management*, *57*(1), 102140. https://doi.org/10.1016/j.ipm.2019.102140

Otar, C. (2018). *How Review Sites Can Affect Your Business (And What You Can Do About It)*. Oct 5, 2018,08:30am EDT. https://www.forbes.com/sites/forbesfinancecouncil/2018/10/05/how-review-sites-can-affect-your-business-and-what-you-can-do-about-it/?sh=28294843266a

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference*, *June*, 497–501.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *1*, 309–319.

Patil, A., & Rane, M. (2021). Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. *Smart Innovation, Systems and Technologies*, *195*, 21–30. https://doi.org/10.1007/978-981-15-7078-0_3

Patil, G. ; S. R. (2021). *A Machine Learning Model to Predict Fake Review using Classifier on Yelp Dataset*. *10*(08), 464–465.

Paul, H., & Nikolaev, A. (2021). Fake review detection on online E-commerce platforms: a systematic literature review. *Data Mining and Knowledge Discovery*, *35*(5), 1830–1881. https://doi.org/10.1007/s10618-021-00772-6

Pendyala, A. (2019). FAKE CONSUMER REVIEW DETECTION. *California State University, Sacramento*, *8*(5), 55.

Popov, M., Kulnitskiy, B., Perezhogin, I., Mordkovich, V., Ovsyannikov, D., Perfilov, S., Borisova, L., & Blank, V. (2018). Catalytic 3D polymerization of C60. *Fullerenes Nanotubes and Carbon Nanostructures*, *26*(8), 465–470. https://doi.org/10.1080/1536383X.2018.1448388

Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *Proceedings of the First Instructional Conference on Machine Learning*, 29–48.

Raschka, S. (2018). *Model Evaluation , Model Selection , and Algorithm Selection in Machine Learning arXiv : 1811 . 12808v3 [cs . LG] 11 Nov 2020*.

Rayana, S., & Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *2015-Augus*, 985–994. https://doi.org/10.1145/2783258.2783370

Saha, S. (n.d.). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*.

Sedighi, Z., Ebrahimpour-Komleh, H., Bagheri, A., & Kosseim, L. (2019). Opinion spam detection with attention-based neural networks. *Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference, FLAIRS 2019*, 245–248.

Sun, C., Du, Q., & Tian, G. (2016). Exploiting Product Related Review Features for Fake Review Detection. *Mathematical Problems in Engineering*, *2016*(1). https://doi.org/10.1155/2016/4935792

Team, P. (2017). How Online Reviews Influence Sales. *Spiegel Research Center*, 20.

Threats, C. S. S. (n.d.). *Cyber Security: Spam, Scams, Frauds and Identity Theft*. 2021. Retrieved January 20, 2022, from https://mediasmarts.ca/digital-media-literacy/digital-issues/cyber-security/cyber-security-spam-scams-frauds-identity-theft

Tim, May; Malcolm, Williams; Richard, Wiggins; Alan, B. (2021). *Deep Learning based Semantic Textual Similarity for Applications in Translation Technology*. *1996*, 6.

UNCTAD. (2022). *Global Trade Update-February 2022*. 1–8. https://unctad.org/system/files/official-document/ditcinf2022d1_en.pdf

Usop, E. S., Isnanto, R. R., & Kusumaningrum, R. (2017). Part of speech features for sentiment classification based on Latent Dirichlet Allocation. *Proceedings - 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2017*, *2018-Janua*, 31–34. https://doi.org/10.1109/ICITACEE.2017.8257670

V M, N., & Kumar R, D. A. (2019). Implementation on Text Classification Using Bag of Words Model. *SSRN Electronic Journal*, *May*, 241–248. https://doi.org/10.2139/ssrn.3507923

Wang, C. C., Day, M. Y., Chen, C. C., & Liou, J. W. (2018). Detecting spamming reviews using long short-term memory recurrent neural network framework. *ACM International Conference Proceeding Series*, 6–10. https://doi.org/10.1145/3234781.3234794

Wang, J., Kan, H., Meng, F., Mu, Q., Shi, G., & Xiao, X. (2020). Fake review detection based on multiple feature fusion and rolling collaborative training. *IEEE Access*, *8*, 182625–182639. https://doi.org/10.1109/ACCESS.2020.3028588

Wang, X., Liu, K., & Zhao, J. (2017). Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 366–376. https://doi.org/10.18653/v1/P17-1034

You, L., Peng, Q., Xiong, Z., He, D., Qiu, M., & Zhang, X. (2020). Integrating aspect analysis and local outlier factor for intelligent review spam detection. *Future Generation Computer Systems*, *102*, 163–172. https://doi.org/10.1016/j.future.2019.07.044

You, Z., Qian, T., & Liu, B. (2018). An attribute enhanced domain adaptive model for cold-start spam review detection. *COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, 1888–1895.

Zhang, D., Zhou, L., Kehoe, J. L., & Kilic, I. Y. (2016). What Online Reviewer Behaviors Really Matter? Effects of Verbal and Nonverbal Behaviors on Detection of Fake Online Reviews. *Journal of Management Information Systems*, *33*(2), 456–481. https://doi.org/10.1080/07421222.2016.1205907

Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing and Management*, *54*(4), 576–592. https://doi.org/10.1016/j.ipm.2018.03.007

Zhang, X., Chen, X., Yao, L., Ge, C., & Dong, M. (2019). Deep neural network hyperparameter optimization with orthogonal array tuning. *Communications in Computer and Information Science*, *1142 CCIS*, 287–295. https://doi.org/10.1007/978-3-030-36808-1_31

Zhao, S., Xu, Z., Liu, L., Guo, M., & Yun, J. (2018). Towards Accurate Deceptive Opinions Detection Based on Word Order-Preserving CNN. *Mathematical Problems in Engineering*, *2018*, 1–8. https://doi.org/10.1155/2018/2410206

Zou, W. Y., Socher, R., Cer, D., & Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1393–1398.

APPENDIX A: Sample of fake review detection dataset

Index	asin	helpful	overall_Rating	review_Text	review_Time	reviewer_ID	reviewer_Name	summary	Review_Time	Spam_ham
32131	B006CU4PTO	[0, 0]	5.0	works great. phone slides in either facing in ...	12 19, 2012	A4QE71V8VKUP	M Mahon	Love it	1.355875e+09	1
35861	B006EC5X6O	[0, 0]	4.0	These are very nice, even though my daughter h...	11 18, 2013	A2KRVM5NY4K6OK	Kathy	Review	1.384733e+09	1
39659	B006FH7OJW	[0, 0]	5.0	Looks great, feels great, works great! Dropped...	05 11, 2013	A2AGR2O5YAOIJK	Kevin Texas "Kevin"	Magpul iPhone 5 field case	1.368230e+09	1
43242	B006GF67X2	[1, 1]	5.0	Came very quickly and in perfect condition. G...	01 11, 2013	AV306T30SPT0F	D. Coleman	Great Service	1.357862e+09	1
10016	B0091WDAA0	[0, 2]	1.0	I got the card placed in my phone then procede...	05 5, 2013	A220WCFHHMV2X4	Edward Gerry "HAPPYMAN"	attempted to connect	1.367712e+09	0
...	...	...	...	...	...	...	...	...	...	...
14743	B0097EDCAA	[0, 0]	5.0	Perfect fit, and went on just like the other m...	10 8, 2013	A108EEYSHGDL6O	3:23 Tech "txplshrk"	Better than the cheapy ones	1.381190e+09	0

APPENDIX A.2: Sample of data after sentiment analysis and POS tagging calculate

	review_Text	Sentiment	Subjectivity	total_Word	Noun_Count	Adj_Count	Verb_Count	Adv_Count	Authenticity	AT	Review_helpful	Rating	Review_class
0	Outstanding HDTV for the money! Before gettin...	0.180546	0.590205	261.0	8.0	4.0	7.0	2.0	0.141762	28.0	46	5	1
1	Just got a Hopper and I purchased this adapter...	0.093750	0.440909	346.0	16.0	6.0	11.0	3.0	0.109827	30.0	1	4	1
2	Today downloaded 33 Gb in about 40 minutes. Mo...	0.500000	0.500000	21.0	17.0	10.0	13.0	9.0	1.476190	23.0	0	4	1
3	The perfect mesh of gaming and regular keyboar...	0.420000	0.435385	24.0	8.0	1.0	1.0	3.0	1.208333	34.0	0	5	1
4	installed easily in conduit, high quality prod...	0.247658	0.607424	28.0	8.0	1.0	11.0	11.0	1.035714	12.0	0	5	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
592832	I can't see anything wrong with it. Responsive...	0.133690	0.437262	77.0	8.0	2.0	2.0	0.0	0.389610	29.0	0	5	1
592833	Its a great product and have a nice design wit...	0.827500	0.882500	19.0	11.0	5.0	5.0	2.0	1.631579	28.0	0	5	0
592834	Unfortunately, these did not correct my proble...	-0.036508	0.648413	60.0	32.0	16.0	23.0	8.0	0.483333	24.0	0	1	0
592835	These turned out to be a great compliment to m...	0.323250	0.520000	49.0	8.0	6.0	10.0	2.0	0.673469	29.0	0	5	0
592836	The card works in my phone but the reader that...	0.350000	0.633333	47.0	17.0	3.0	13.0	6.0	0.702128	21.0	0	4	0

592837 rows × 13 columns

APPENDIX B: Fake review detection feature and description of the dataset

Feature	Description
ReviewerID	an identifier on Amazon's reviewer ID, e.g. A2JNWBC9SABL5E
ReviewerName	The account name who posted the review.
Asin	identifier for the product id, e.g. B008OHNZI0
ReviewTime	At the time of review, the review is posted e.g. 01, 6, 2016
ReviewText	The review's text content is given by the reviewer.
Overall	Ratings of 1 to 5 stars are assigned to the review.
No of Feedbacks	The review comments are given for the reviewed product.
Helpful	The number of helpful votes associated with the review
UnixReviewTime	If the review is a verified purchase, TRUE is assigned. Else, FALSE.
Summary	The summary of the review

APPENDIX C

Sample Prototype Code

```
 #print(X2_train)
    import tensorflow as tf
    from keras.models import load_model
    path = 'models/Fake_eview_model.h5'
    loaded_model= tf.keras.models.load_model(path )
    app = Flask(__name__)

@app.route("/")
def home():
    return render_template("Fake_Review_detection.html")

@app.route('/handle_data', methods=['POST'])
def handle_data():
    features = np.asarray(X2_train).astype(np.float32)  #Convert to the form [[a, b]] for input to the model
    prediction = loaded_model.predict(features)  # features Must be in the form [[a, b]]
    predict_class=(prediction > 0.5).astype("int32")
    output = [float(np.round(x)) for x in predict_class]
    if(output==[1.0]):
        x="Guinune"
    else:
        x="Fake"
    print(prediction)
    return render_template('Fake_Review_detection.html', prediction_text='Your Review is {}'.format(x))
if __name__ == "__main__":
    app.run(debug=True)
```

APPENDIX D

Notation	Definition
$rating_r$	Rating given in review ‘r’.
$rating_r(p)$	Rating of review ‘r’ on electronics product.
r_α	Review of reviewer ‘a’.
$N_r(p)$	Number of reviews on electronics product.
x	Number of maximum number of reviews word.
$Nr(\alpha)$	Number of reviews posted by reviewer ‘a’

APPENDIX E: Word cloud fake review detection Picture

APPENDIX F: Simple fake review detection prototype