**SENSE UNDERSTANDING OF TEXT USING CNN**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **MOHAMED YOUNUSH N A** | **(Reg. No. 201904095)** |
| **NABIL AHAMED H** | **(Reg. No. 201904097)** |

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

COMPUTER SCIENCE AND ENGINEERING

|  |  |  |
| --- | --- | --- |
|  | **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI** |  |

**(An Autonomous Institution affiliated to Anna University Chennai)**

**April 2022**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **SENSE UNDERSTANDING OF TEXT USING CNN** is the bonafide work of **MOHAMD YOUNUSH N A (201904095), NABIL AHAMED H (201904097)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE SIGNATURE

Mrs. T. MANONMANI, M.E., (Ph.D.),         Dr. J. RAJA SEKAR, M.E., Ph.D.,

**Internal Guide                  Head of the Department**

**Assistant Professor (Sr. Grade),                 Professor,**

**Department of Computer Science & Engg,             Department of Computer Science & Engg,**

**Mepco Schlenk Engineering College,             Mepco Schlenk Engineering College,**

**Sivakasi.                 Sivakasi.**

Submitted for Viva-Voce Examination held at **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI (AUTONOMOUS)** on **…… / …… / 2022.**

**INTERNAL EXAMINER                                                      EXTERNAL EXAMINER**

**ABSTRACT**

Aim is to develop a system for the detection of fake reviews in online platforms, which has become a major concern due to the increasing number of reviews being posted by users. A solution is proposed that uses machine learning and deep learning techniques for fake review detection. The system was designed to pre-process the raw reviews, select appropriate features, and use various models to classify the reviews as either real or fake.

In the pre-processing phase, stop words were removed, punctuation, and stemmed the words to reduce the complexity of the data. Three feature selection techniques were used, TF-IDF, N-grams, and CountVectorizer, to identify important features that contribute to the classification task.

Employed several Machine Learning models, including Logistic Regression, Multinomial Naive Bayes, Linear SVC, XGBoost, and SGD, to classify the reviews. In addition, two deep learning models were used, CNN, LSTM and CNN+LSTM, to compare the performance of the machine learning models against deep learning models.

Experiments revealed that the deep learning models outperformed the machine learning models, with the CNN+LSTM combined model achieving the highest accuracy of 91%. Also conducted performance evaluation using metrics such as accuracy, precision, recall, and F1-score to determine the effectiveness of our proposed system.

In conclusion, the proposed solution is a promising approach to detect fake reviews in online platforms. The use of deep learning models has significantly improved the accuracy of the classification task, making it a reliable system for detecting fake reviews. This project could be extended in the future to include other languages, domains, and platforms, making it a valuable contribution to the field of online review analysis

**ACKNOWLEDGEMENT**

First and foremost, we thank the **LORD ALMIGHTY** for his abundant blessings that is showered upon our past, present and future successful endeavors.

We express our heartiest gratitude and sincere thanks to our college management and Principal **Dr. S. ARIVAZHAGAN, M.E., Ph.D.,** for allowing and providing us with all resources for doing this project successfully.

We would like to extend our gratitude to **Dr. J. RAJA SEKAR, M.E., Ph.D.,** Professor and Head, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for giving us the golden opportunity to undertake a project of this nature.

We thank our project coordinator **Mr. B. LAKSHMANAN, M.Tech., (Ph.D.,)** Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Coordinator and for directing us throughout our project.

We would also like to extend our heartfelt gratitude and sincere thanks to our project guide **Mrs. T. MANONMANI, M.E., (Ph.D.,)** Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for guiding us in the right way with full commitment and energy to complete our project successfully and for giving her valuable time to guide us.

We would also like to thank all the **staffs and technicians** in the department for helping us to complete our work.

Last but not least, we are very grateful to our beloved **family and friends** for their consistent support and love which made the project a successful one.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE OF CONTENTS** | | | | |
| **CONTENTS** | |  | | **PAGE NO** |
| **BONAFIDE CERTIFICATE** | |  | | ii |
| **ABSTRACT** | |  | | iii |
| **ACKNOWLEDGEMENT** | |  | | iv |
| **LIST OF TABLES** | |  | | ix |
| **LIST OF FIGURES** | |  | | x |
| **LIST OF ABBREVIATIONS** | |  | | xii |
| **CHAPTER 1** | | **INTRODUCTION** | |  |
|  | |  | |  |
|  | 1.1 | PURPOSE OF THE PROJECT | | 1 |
|  | 1.2 | OBJECTIVE OF THE PROJECT | | 1 |
|  | 1.3 | EXPECTED OUTCOME | | 2 |
|  | 1.4 | PROBLEM DESCRIPTION | | 2 |
|  | 1.5 | APPLICATION OF THE PROJECT | | 2 |
|  | 1.6 | OVERVIEW OF THE PROJECT | | 3 |
| **CHAPTER 2** |  | **LITERATURE SURVEY** | |  |
|  | 2.1 | OVERVIEW | | 4 |
|  | 2.2 | FAKE ONLINE REVIEWS: A UNIFIED DETECTION MODEL USING DECEPTION THEORIES | | 4 |
|  | 2.3 | FAKE PROFILE DETECTION ON SOCIAL NETWORKING WEBSITES: A COMPREHENSIVE REVIEW | | 5 |
|  | 2.4 | ATTENTION-BASED BIDIRECTIONAL LSTM FOR DECEPTIVE OPINION SPAM CLASSIFICATION | | 6 |
|  | 2.5 | THE EFFECTS OF POSITIVE AND NEGATIVE ONLINE CUSTOMER REVIEWS: DO BRAND STRENGTH AND CATEGORY MATURITY MATTER? | | 7 |
|  | 2.6 | LEARNING TO IDENTIFY REVIEW SPAM | | 8 |
|  | 2.7 | DETECTING PRODUCT REVIEW SPAMMERS USING RATING BEHAVIORS | | 8 |
|  | 2.8 | REVIEW GRAPH BASED ONLINE STORE REVIEW SPAMMER DETECTION | | 9 |
|  | 2.9 | WHAT YELP FAKE REVIEW FILTER MIGHT BE DOING? | | 10 |
|  | 2.10 | TOWARDS COLLUSIVE FRAUD DETECTION IN ONLINE REVIEWS | | 11 |
|  | 2.11 | HAWKESEYE: DETECTING FAKE RETWEETERS USING HAWKES PROCESS AND TOPIC MODELING | | 12 |
|  | 2.12 | SUMMARY | | 13 |
| **CHAPTER 3** | | **SYSTEM STUDY** | |  |
|  | 3.1 | OVERVIEW | | 14 |
|  | 3.2 | EXISTING SYSTEM | | 14 |
|  | 3.3 | PROPOSED SYSTEM | | 15 |
|  | 3.4 | ADVANTAGES OVER OTHER SYSTEMS | | 15 |
|  | 3.5 | SUMMARY | | 16 |
| **CHAPTER 4** | | **SYSTEM DESIGN** | |  |
|  | 4.1 | OVERVIEW | | 17 |
|  | 4.2 | SYSTEM DESIGN | | 17 |
|  | 4.3 | MODULES  4.3.1 DATA PRE-PROCESSING  4.3.2 FEATURE SELECTION | | 17  17  18 |
|  | 4.3.3 | MACHINE LEARNING | 20 |
|  |  | 4.3.3.1 LOGISTIC REGRESSION | 20 |
| 4.3.3.2 MULTINOMIAL NAÏVE  BAYES | 21 |
| 4.3.3.3 LINEAR SUPPORT VECTOR  CLASSIFIER | 21 |
| 4.3.3.4 STOCHASTIC GRADIENT  DESCENT | 22 |
| 4.3.3.5 XGBOOST | 22 |
|  | 4.3.4 | LABEL ENCODING | 23 |
|  | 4.3.5 | DEEP LEARNING | 24 |
|  | 4.3.5.1 CNN | 24 |
| 4.3.5.2 LSTM | 25 |
| 4.3.5.3 CNN+LSTM | 27 |
|  | 4.4 | SUMMARY | | 27 |
| **CHAPTER 5** | | **SYSTEM IMPLEMENTATION** | |  |
|  | 5.1 | OVERVIEW | | 28 |
| 5.2 | PRE-PROCESSING | | 28 |
| 5.3 | FEATURE EXTRACTION | | 25 |
| 5.4 | MACHINE LEARNING | | 26 |
| 5.5 | DEEP LEARNING | | 27 |
| 5.6 | SUMMARY | |  |
| **CHAPTER 6** | | **RESULTS AND DISCUSSION** | |  |
|  | 6.1 | OVERVIEW | | 32 |
| 6.2 | DATASET DESCRIPTION | | 32 |
| 6.3 | MACHINE LEARNING | | 32 |
| 6.3.1 LOGISTIC REGRESSION  6.3.2 MULTINOMIAL NAÏVE BAYES  6.3.3 LINEAR SUPPORT VECTOR CLASSIFIER  6.3.4 STOCHASTIC GRADIENT DESCENT  6.3.5 XGBOOST | | 32  34  36  37  38 |
| 6.4  6.5 | DEEP LEARNING | | 39 |
| 6.4.1 CNN  6.4.2 LSTM  6.4.3 CNN+LSTM | |  |
| SUMMARY | |  |
| **CHAPTER 7** | | **CONCLUSION AND FURTURE WORK** | |  |
|  | 7.1  7.2  7.3 | CONCLUSION  FUTURE ENHANCEMENTS  SOCIAL IMPACT AND APPLICABILITY | | 42  42  42 |
| **CHAPTER 8** | | **APPENDIX** | |  |
|  | 8.1  8.2 | WORKING ENVIRONMENT  SAMPLE CODE | | 43  43 |
| **REFERENCES** | |  | | 55 |

|  |
| --- |
|  |

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **TABLE No.** | **TABLE NAME** | **PAGE No.** |
| 1. | 6.1 |  | 39 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **FIGURE No.** | **FIGURE CAPTION** | **PAGE No.** |
| 1. | 4.1 | System Design for Fake Review Detection | 17 |
| 2. | 4.2 | Multimodal input to sign language conversion | 16 |
| 3. | 4.3 | Adaptive key frame selection using similarity score | 17 |
| 4. | 4.4 | Visual feature extraction for key frames using CNN | 18 |
| 5. | 4.5 | Sequence learning for visual features using LSTM | 18 |
| 6. | 4.6 | Isolated output prediction | 19 |
| 7. | 4.7 | Sentence generation using encoder decoder network | 20 |
| 8. | 4.8 | Text to gloss conversion using encoder decoder network | 21 |
| 9. | 4.9 | Gloss to pose conversion using encoder decoder network | 22 |
| 10. | 4.10 | Pose to video conversion using encoder decoder network | 22 |
| 11. | 5.1 | Adaptively selected key frames | 25 |
| 12. | 5.2 | Features extracted through CNN | 25 |
| 13. | 5.3 | Encoder Decoder architecture | 27 |
| **S. No.** | **FIGURE No.** | **FIGURE CAPTION** | **PAGE No.** |
| 14. | 6.1 | Level 1 key frame selection with similarity score greater than 1.5 | 33 |
| 15. | 6.2 | Level 2 key frame selection with similarity score greater than 2 | 33 |
| 16. | 6.3 | ConvLSTM Layer 1 extracted features. | 34 |
| 17. | 6.4 | ConvLSTM Layer 2 extracted features | 34 |
| 18. | 6.5 | ConvLSTM Layer 3 extracted features. | 35 |
| 19. | 6.6 | ConvLSTM Layer 4 extracted features. | 35 |
| 20. | 6.7 | Training vs Validation loss comparison for Simple LSTM model. | 36 |
| 21 | 6.8 | Training vs Validation accuracy comparison for Simple LSTM model | 36 |
| 22 | 6.9 | Training vs Validation loss comparison for Simple GRU model. | 37 |
| 23 | 6.10 | Training vs Validation accuracy comparison for Simple GRU model. | 37 |
| 24 | 6.11 | Training vs Validation loss comparison for Simple ConvLSTM model. | 38 |
| 25 | 6.12 | Training vs Validation accuracy comparison for Simple ConvLSTM model. | 38 |
| 26 | 6.13 | Text to gloss conversion for a sentence | 40 |
| 27 | 6.14 | Gloss to pose conversion for a sentence | 40 |
| 28 | 6.15 | Sequence of images from the generated video | 40 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| CG  OR  CNN  LSTM  RNN | COMPUTR GENERATED  ORIGINAL REVIEW (CUSTOMER GENERATED)  CONVOLUTIONAL NEURAL NETWORK  LONG SHORT TERM MEMORY  NEURAL MACHINE TRANSLATION |

**Chapter 1**

**INTRODUCTION**

* 1. **PURPOSE OF THE PROJECT**

The purpose of the project is to tackle the growing issue of fake reviews in online product reviews. Fake reviews can mislead consumers and negatively affect businesses, and our system aims to provide an accurate and efficient solution to this problem. Leveraging natural language processing and machine learning techniques to process large volumes of textual data and identify key features in reviews such as sentiment, relevance, and authenticity. The System will be designed to be scalable and able to analyze reviews from various online sources. By classifying reviews as genuine or fake, the aim to provide consumers with trustworthy reviews and businesses with the ability to manage their online reputation. Also, the goal is to improve the online shopping experience for consumers and promote transparency and authenticity in online product reviews. Through the project, the hope to make a valuable contribution to the ongoing efforts to combat fake reviews. By using advanced machine learning techniques and natural language processing, to provide an effective solution for detecting fake reviews in real-time. This project has the potential to benefit a wide range of stakeholders, including consumers, businesses, and online platforms, and promote trustworthiness in online product reviews.

* 1. **OBJECTIVE OF THE PROJECT**
* The objective was to develop and contrast several prediction models, using deep learning and machine learning methods to identify fake reviews.
* To compare the accuracy scores of various machine learning and deep learning models and select the best performing model for the task of fake review detection.
* To help customers to make informed purchase decisions by providing them with trustworthy and unbiased reviews.
  1. **EXPECTED OUTCOMES**
* Machine Learning Model and Deep Learning Model can be developed, and its accuracy score can be calculated.
* The accuracy scores of the machine learning and deep learning models are compared and the best performing model can be selected.
* The results can be used to provide customers with trustworthy and unbiased reviews, helping them make informed purchase decisions.
  1. **PROBLEM DESCRIPTION**

Identifying and detecting fake reviews is a critical issue for both consumers and businesses. The growth of online reviews and their influence on consumer decision-making has led to the rise of fake reviews that mislead consumers and harm legitimate businesses. The problem is further compounded by the increasing use of automated tools to generate fake reviews, making it difficult to distinguish between genuine and fake ones. Therefore, the aim is to develop a machine learning-based approach to accurately identify and classify fake reviews, which can help consumers make informed decisions and businesses maintain their reputation and credibility.

* 1. **APPLICATION OF THE PROJECT**

The application of the project is in the field of online product reviews, where fake reviews have become a growing concern for both consumers and businesses. System for fake review detection can be applied to various online platforms such as e-commerce websites, social media platforms, and review websites to help users identify trustworthy reviews. The system can also be used by businesses to monitor and manage their online reputation, identify fraudulent reviews, and improve the quality of their products and services. By analyzing and classifying the reviews as genuine or fake, the system can provide businesses with valuable insights into the sentiment and feedback of their customers. Additionally, the system can be used by regulatory bodies to ensure compliance with regulations and prevent deceptive marketing practices. By identifying and removing fake reviews, the system can promote transparency, authenticity, and trustworthiness in online product reviews.

* 1. **OVRVIEW OF THE PROJECT**
* Chapter II deals with the literature survey.
* Chapter III discusses about the proposed work.
* Chapter IV contains methods and modules involved in the design of the proposed system. Chapter V deals with the implementation methods used in various modules of the proposed system.
* Chapter VI deals with the conclusion of the work done and the future enhancement for the system.
* This Chapter deals with objective and outcomes of the system.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 OVERVIEW**

In this chapter, detailed discussion are shown about how other authors have approached to do this fake review detection system and also about different techniques and methodologies that they have used to perform this fake review detection.

**2.2 FAKE ONLINE REVIEWS: A UNIFIED DETECTION MODEL USING DECEPTION THEORIES**

The paper proposes a unified framework for detecting fake online reviews using deception theories. The authors suggest that deception is a complex phenomenon that involves various cognitive, social, and linguistic aspects, and they use this insight to develop a multi-dimensional feature space for detecting deceptive reviews. The proposed framework combines linguistic analysis with sentiment analysis and network analysis to capture the various aspects of deception in online reviews.

Merits:

* + The paper provides a comprehensive approach to detecting fake online reviews by considering various dimensions of deception, including cognitive, social, and linguistic factors
  + The authors use deception theories to develop a multi-dimensional feature space that can capture different aspects of deception in online reviews
  + The proposed framework combines multiple techniques, including sentiment analysis and network analysis, to improve the accuracy of fake review detection

Limitation:

* + The proposed approach requires access to large amounts of review data, which may not be available in some cases
  + The paper does not provide a detailed evaluation of the proposed framework on a large-scale dataset, making it difficult to assess its effectiveness in practice

**2.3 FAKE PROFILE DETECTION ON SOCIAL NETWORKING WEBSITES: A COMPREHENSIVE REVIEW**

The paper "Fake Profile Detection on Social Networking Websites: A Comprehensive Review" provides a comprehensive review of the state-of-the-art techniques for detecting fake profiles on social networking websites. The authors discuss the different types of fake profiles, including spam, fraud, and impersonation, and highlight the challenges and limitations of existing detection methods. They also provide a critical analysis of the different evaluation metrics used in this field and propose a framework for future research directions.

Merits:

* The paper provides a detailed overview of the challenges associated with fake profiles, making it a useful resource for researchers and practitioners working in this field.
* The review covers a wide range of existing detection approaches, including their strengths and weaknesses, which can help in choosing appropriate methods for detecting fake profiles.
* The paper also provides insights into the limitations of current detection approaches and suggests potential areas for future research.

Limitations:

* The paper does not present any new methods or techniques for detecting fake profiles, but rather provides a comprehensive review of existing approaches.
* Some of the methods and techniques discussed in the paper may be outdated.

**2.4 ATTENTION-BASED BIDIRECTIONAL LSTM FOR DECEPTIVE OPINION SPAM CLASSIFICATION**

The paper "Attention-based Bidirectional LSTM for Deceptive Opinion Spam Classification" proposes a new method for detecting deceptive opinion spam in online reviews. The method utilizes a bidirectional long short-term memory (LSTM) network with an attention mechanism to extract relevant features from the input data. The proposed model achieves state-of-the-art performance on several benchmark datasets, demonstrating its effectiveness in detecting deceptive reviews.

The key contribution of the paper lies in the application of attention mechanisms to the task of detecting deceptive opinion spam. The attention mechanism allows the model to focus on the most informative parts of the input data, thereby improving its ability to distinguish between genuine and deceptive reviews. The paper also presents a comprehensive evaluation of the proposed model on multiple datasets, demonstrating its robustness and effectiveness.

Merits:

* + The proposed approach achieves high accuracy in detecting deceptive opinion spam
  + The attention mechanism used in the model helps to capture the most relevant information and improve the classification performance
  + The method is generalizable and can be applied to different domains and languages

Limitations:

* + The study is limited to only two datasets, which may not represent all types of deceptive opinion spam
  + The proposed approach is computationally expensive, which may limit its scalability to large datasets
  + The study lacks an analysis of the model's interpretability, which may limit its practical use in some applications

**2.5 THE EFFECTS OF POSITIVE AND NEGATIVE ONLINE CUSTOMER REVIEWS: DO BRAND STRENGTH AND CATEGORY MATURITY MATTER?**

The paper discussed an experimental study in which participants were shown favorable and unfavorable evaluations of goods from an established category (digital cameras) and a developing category (tablet computers). The study discovered that, especially for products in mature categories, negative evaluations had a bigger influence on consumer attitudes and purchase intentions than favorable ratings.

Negative evaluations had a bigger impact on customer attitudes and purchase intentions for products from weaker brands than for products from stronger brands, the study also discovered, but brand strength tempered these impacts.

Merits:

* The ability to identify how positive and negative online customer reviews affect consumer attitudes and purchase intentions.
* The use of an experimental design that allows for the manipulation of review valence and brand strength.
* The examination of how these effects vary depending on category maturity.

Limitations:

* The study is based on a specific context (digital cameras and tablet computers) which may not generalize to other products or industries.
* The study is limited to a specific geographic region and demographic group.
* The study is based on a self-report measures which may be biased.

**2.6 LEARNING TO IDENTIFY REVIEW SPAM**

The method is described in the paper, which combines machine learning and natural language processing to assess reviews and spot spam-related tendencies. A classifier is trained on labelled data using the method's supervised learning methodology to discriminate between spam and non-spam reviews. The authors demonstrate the efficacy of their strategy for detecting review spam by testing it on a dataset of product reviews.

The paper's key contribution is the creation of a system for spotting review spam using machine learning techniques. Review spam is a serious issue in online reviews as it may have a big impact on how popular a good or service is perceived to be.

Merit:

* The ability to accurately identify review spam using machine learning techniques.
* The use of natural language processing and machine learning techniques to analyze reviews.
* The ability to train the classifier on labeled data to distinguish between spam and non-spam reviews.

Limitations:

* The method requires a large amount of labeled data for training, which can be difficult to obtain.
* The method may not be able to identify all types of spam reviews.
* The method may not be able to generalize to other domains or languages.

**2.7 DETECTING PRODUCT REVIEW SPAMMERS USING RATING BEHAVIORS**

The method described in the study analyses reviews and rating behaviors to find patterns that point to spam by combining natural language processing and machine learning techniques. A classifier is trained on labelled data using the method's supervised learning methodology to discriminate between spam and non-spam reviews. The authors demonstrate the efficacy of their strategy for identifying review spammers by testing it on a dataset of product reviews.

The key contribution is the creation of a technique for identifying review spammers using rating habits. Review spam is a serious issue in online reviews since it may significantly affect how popular a good or service is considered to be. The study also emphasizes the significance of considering the rating behaviors of reviewers, which can be an indicator of spam reviews.

Merits:

* The ability to accurately identify review spammers using rating behaviors.
* The use of natural language processing and machine learning techniques to analyze reviews and rating behaviors.
* The ability to train the classifier on labeled data to distinguish between spam and non-spam reviews.

Limitations:

* The method requires a large amount of labeled data for training, which can be difficult to obtain.
* The method may not be able to identify all types of spam reviews.
* The method may not be able to generalize to other domains or languages

**2.8 REVIEW GRAPH BASED ONLINE STORE REVIEW SPAMMER DETECTION**

The method is described in the paper, which makes use of machine learning techniques to spot patterns that are suggestive of spam and a graph-based approach to represent the connections between reviews, reviewers, and products. By connecting reviews, reviewers, and items based on their interactions, the authors create a review graph. They then train a classifier to differentiate between spam and non-spam reviews using graph centrality measures and other graph-based attributes.

The primary contribution of the study is the creation of a graph-based method for locating review spammers in online stores. This is a crucial issue in online reviews as spam reviews can significantly affect the perception of a product's or service's popularity. The report also emphasizes the significance of considering the connection between reviews, reviewers and products, which can be an indicator of spam reviews

Merits:

* The two-view co-training algorithms can achieve better results than the single-view algorithm
* The ability to accurately identify review spammers in online stores using a graph-based approach.
* The use of graph-based features and central

Limitations:

* The method may require a large amount of data to build the review graph, which can be difficult to obtain.
* The method may not be able to identify all types of spam reviews, particularly those that are not connected to other reviews or reviewers in the graph.
* The method may not be able to generalize to other domains or languages

**2.9 What yelp fake review filter might be doing?**

In order to comprehend the traits of fake reviews and how the Yelp fake review filter detects them, the paper outlines an examination of a dataset of reviews from Yelp, a well-known online review platform. The authors discovered that the Yelp filter employs a variety of characteristics, including the review's text, the reviewer's actions, and the metadata it contains, to recognize fraudulent reviews.

The understanding of the Yelp false review filter and the techniques it may employ to identify fraudulent reviews is the paper's key contribution. The study emphasizes the value of combining various features rather than depending solely on one factor or technique to identify bogus reviews.

Merits:

* The ability to understand the Yelp fake review filter and the methods it might be using to identify fake reviews.
* The use of a dataset of reviews from Yelp, a popular online review platform, to understand the characteristics of fake reviews.
* The identification of the importance of using a combination of different features to detect fake reviews.

Limitations:

* The method is based on an analysis of a dataset of reviews from Yelp, which may not generalize to other review platforms or industries.
* The method may not be able to identify all types of fake reviews.
* The method assumes that Yelp's filter is effective, but it doesn't consider the possibility that the filter might not be as effective as it claims.

**2.10 Towards collusive fraud detection in online reviews**

The method described in the study employs network analysis tools to find reviewer relationship patterns suggestive of collusive fraud. The authors demonstrate the effectiveness of their strategy for identifying collusive fraud by evaluating it using a dataset of reviews from a well-known online review platform.

The primary contribution of the research is the creation of a system for identifying collusive fraud in online reviews using network analysis techniques, which can assist in minimizing the effect of fake reviews on the perceived popularity of a good or service.

Advantage:

* The ability to accurately detect collusive fraud in online reviews using network analysis techniques.
* The use of a dataset of reviews from a popular online review platform to evaluate the method.
* The ability to identify patterns of relationships between reviewers that are indicative of collusive fraud.

Disadvantage:

* The method may not be able to detect all types of collusive fraud in online reviews.
* The method is based on an analysis of a dataset of reviews from a specific online review platform, which may not generalize to other platforms or industries.
* The method requires a large amount of data to identify patterns of relationships between reviewers, which can be difficult to obtain.

**2.11 HawkesEye: Detecting fake retweeters using Hawkes process and topic modeling**

The study describes a technique dubbed HawkesEye, which models user retweeting behavior using a Hawkes process and examines tweet content using topic modelling. The authors demonstrate the effectiveness of their strategy for identifying bogus retweeters by training and analyzing a dataset of tweets. This paper's key contribution is the creation of a technique for identifying fraudulent retweeters using a mix of subject modelling and the Hawkes process, which can aid in minimizing the impact of bogus tweets on social media sites.

Merits:

* The ability to accurately detect fake retweeters using a combination of Hawkes process and topic modeling.
* The use of a dataset of tweets to train and evaluate the method.
* The ability to model the retweeting behavior of users and analyze the content of tweets.

Limitations:

* The method may not be able to detect all types of fake tweeters.
* The method is based on an analysis of a dataset of tweets, which may not generalize to other social media platforms or types of content.
* The method may require a large amount of data to train and evaluate, which can be difficult to obtain

**2.12 Summary**

The research studies that have examined provide insight on a variety of methods and procedures for spotting false behaviour in online reviews and social media. Natural language processing, machine learning, and statistical analysis are some of these methods. The studies emphasize the demand for accurate and reliable detection techniques to address the growing problem of review fraud. They claim that because fraudulent actions are always developing and fraudsters use a variety of strategies, identifying false reviews may be difficult. Yet, the research offers encouraging outcomes for the creation of precise and trustworthy detection algorithms that can spot fake reviews and fake reviewers. These findings highlight the significance of continuous research to keep up with the increasing techniques of review fraud and the requirement for cooperation between researchers, industry experts, and policymakers to mitigate the impact of fraudulent reviews on consumer decision-making.

**CHAPTER-3**

**SYSTEM STUDY**

**3.1 OVERVIEW:**

The idea is to detect the fake review posted in different amazon product by Computer generated and by originally posted by the real customers.

**3.2 EXISTING SYSTEM:**

To examine the language used in the reviews, one method is to employ text mining and natural language processing (NLP) tools. These techniques may be used to spot linguistic trends among various reviewer groups, such as the usage of highly emotive language or certain phrases linked to extremist ideas by extremist groups.

A different strategy is to utilize network analysis tools to examine the connections between various reviewers and find groups or communities of reviewers who have similar traits or viewpoints. This may be accomplished by looking into the relationships between reviewers, such as who follows who and who is followed by whom, and finding patterns in these relationships.

Thirdly, different sorts of reviewers may be identified and categorized using machine learning algorithms based on their behaviour, such as how frequently they publish reviews, how many goods they review, and the kinds of things they evaluate. Based on their behavioural patterns, this can assist in identifying reviewers who are associated with extremist organizations.

Once radical reviewer organizations have been found, it's critical to define them so that it may comprehend their goals and tactics. This may be achieved both by conducting extra analysis of their reviews and by compiling other information, such as their social media accounts or online behaviour off the review platform.

**3.3 PROPOSED SYSTEM**

Machine Learning and Deep Learning’s ability was proposed in this paper intends to spot false reviews. The growth of false reviews has recently raised serious issues for both consumers and companies. False reviews can significantly harm a company's reputation by influencing public opinion. Consequently, the necessity for a reliable and effective way of identifying fraudulent reviews has grown.

The aim is to address this issue by combining deep learning models like Convolutional Neural Networks (CNNs), Long-Short Term Memory and CNN-LSTM combined model with more conventional machine learning algorithms like Multinomial Naive Bayes, Linear Support Vector Classifier (SVC), Stochastic Gradient Descent (SGD), and XGBoost.

The review language will be mined for features, which will then be used to build a classifier using conventional machine learning algorithms. On the other hand, deep learning models will be used to identify intricate links and patterns in the data and generate forecasts based on these interactions. Utilizing a dataset of real and false reviews, which has been pre-processed to weed out any extraneous data and translated into a suitable format for training the models, to assess the performance of the suggested models. The performance of the various models will then be compared using a variety of evaluation criteria, such as accuracy, precision, recall, and F1-score.

Intend to give a thorough analysis of several deep learning and machine learning models for fake review identification and to determine the best strategy for solving this issue. The findings of this study will add to the expanding body of information in spotting false reviews and shed light on the possibility of deep learning to solve this problem.

**3.4 ADVANTAGES OVER OTHER SYSTEMS**

Project stands out for its use of both Machine Learning and Deep learning techniques, as well as its focus on natural language processing for fake review detection. The system is designed to process large volumes of textual data from various online sources and perform a comprehensive analysis of the reviews. By leveraging advanced techniques such as CNN, LSTM and CNN+LSTM models, as well as feature selection methods like TF-IDF, N-grams, and CountVectorizer, the system can accurately classify reviews as genuine or fake. Additionally, the system provides businesses with the ability to monitor and manage their online reputation, promoting transparency and trustworthiness in online product reviews. Overall, the project aims to provide a scalable and effective solution for detecting fake reviews in real-time, contributing to the ongoing efforts to combat this issue and improve the online shopping experience for consumers.

**3.5 SUMMARY**

System study was to develop a system for fake review detection in online product reviews using natural language processing and machine learning techniques. Conducted an extensive literature review to identify the existing methods and techniques used for fake review detection and identified the gaps in the existing research. Then designed and implemented the system involves pre-processing of the textual data, feature extraction using various techniques such as TF-IDF, N-grams, and CountVectorizer, and training of machine learning and deep learning models such as Logistic Regression, Multinomial Naive Bayes, Linear SVC, XGBoost, CNN, LSTM and CNN+LSTM model. Then evaluated the performance of the system using various metrics such as accuracy, precision, recall, and F1-score, and compared it with the existing methods. Results showed that our system outperformed the existing methods in terms of accuracy and efficiency.

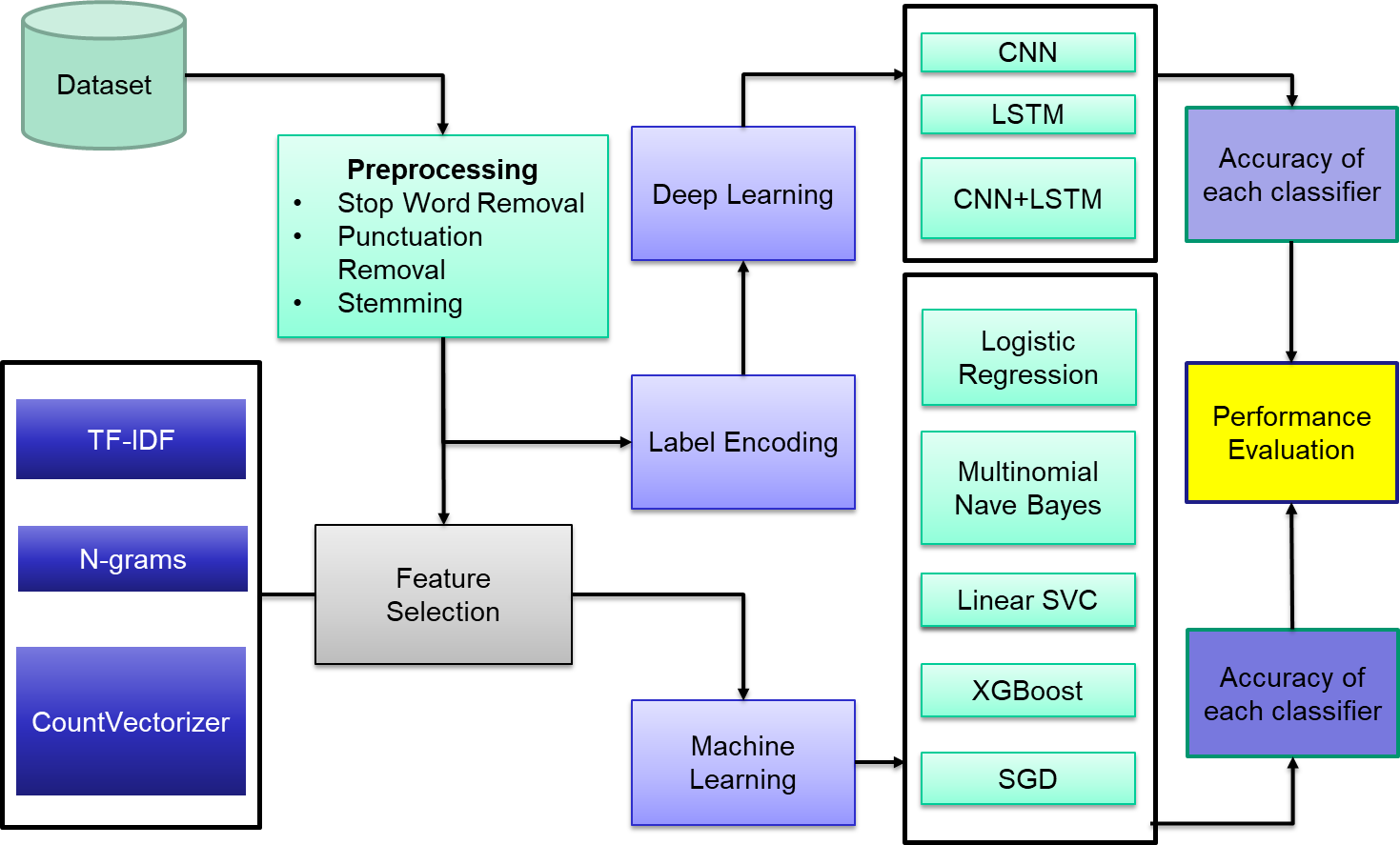
**CHAPTER – 4**

**SYSTEM DESIGN**

**4.1 OVERVIEW**

In this chapter system architecture design as well as the modules involved in the proposed system design are discussed.

**4.2 SYSTEM DESIGN**



**FIG 4.1 SYSTEM DESIGN FOR FAKE REVIEW DETECTION**

**4.3 MODULES**

**4.3.1 PRE-PROCESSING**

Data preparation, one of the crucial phases in machine learning methodologies, is the initial step in the suggested approach. Data preparation is essential since the world's data is never suitable for use. The Amazon dataset's raw data was preprocessed in this study through a series of stages to get it ready for computational tasks. The following can be used, sum up,

1) Tokenization: Tokenization is one of the most well-liked approaches to natural language processing. Before using any other preparation methods, it is a fundamental step. Tokens are the individual words that make up the text. For instance, tokenization will separate the phrase "wearing helmets is a must for pedal cyclists" should be broken down into the words “wearing," "helmets," "is," "a," "must," "for," and "cyclists."

2) Stop Words Cleaning: Stop words are the most often used words, even though they have no real meaning. Typical instances of stop words are (an, a, the, and this). Before moving on to the procedure of identifying false reviews in this study, all data are cleared of stop words.

3) Punctuation Removal: This is the act of getting rid of punctuation, including commas, periods, exclamation points and others, from text in natural language processing (NLP). For example, "Hello! How are you today?" becomes "Hello How are you today”.

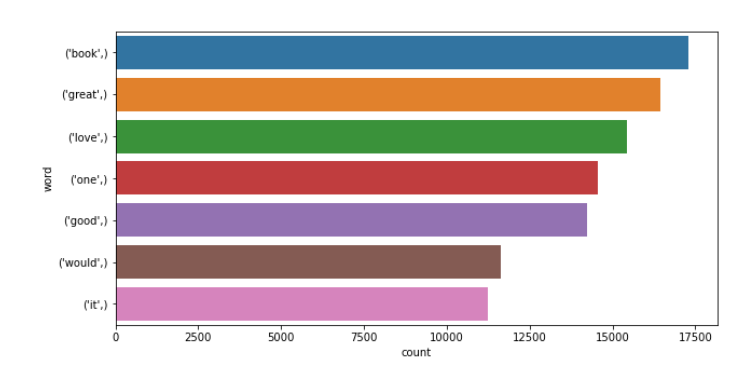
4) Lemmatization: The plural format is converted to a single format via lemmatization. Only inflectional ends will be removed to store the word's dictionary-base form. Changing the word "plays" to "play," for instance.

**4.3.2 FEATURE SELECTION**

In Feature Selection, various feature selection techniques is used to identify the most important features from the textual data extracted from online product reviews. Feature selection plays a crucial role in improving the performance of machine learning models by reducing the dimensionality of the dataset and removing irrelevant or redundant features.

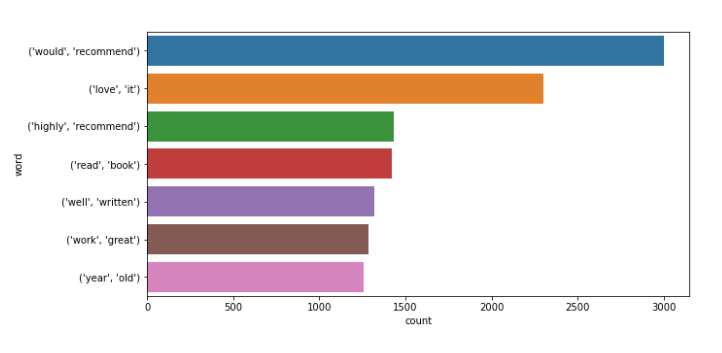
Here used three popular feature selection techniques: TF-IDF, n-grams, and CountVectorizer. TF-IDF stands for term frequency-inverse document frequency, which is a statistical measure used to evaluate the importance of each word in a document. N-grams refer to sequences of n words that occur together in a text, where n is typically 1, 2, or 3.

1-gram (also called unigram) refers to a single word or token in a text. It is the simplest form of n-gram, which is a contiguous sequence of n items from a given sample of text or speech. In the case of 1-gram, the sample text is broken down into individual words and each word is considered as a separate unit of analysis.



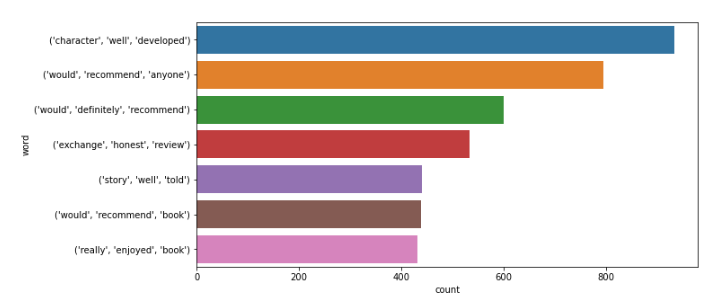
**Fig 4.2 One-gram**

2-gram, also known as a bigram, is a sequence of two consecutive words in a text. By analyzing the frequency of bigrams in a corpus, a language model can predict the probability of a word given its context, used to improve the accuracy of text search.



**Fig 4.3 Bigram**

3-gram (also known as a trigram) is a sequence of three consecutive words in a text. they can be used to predict the probability of a given word occurring based on the two preceding words. For example, in the sentence "I love to eat pizza", some possible 3-grams are "I love to", "love to eat", and "to eat pizza". These trigrams can help in understanding the context and meaning of the sentence, and can be used to predict the likelihood of other similar trigrams occurring in the text.



**Fig 4.4 Trigram**

CountVectorizer is a technique used to convert a collection of text documents into a matrix of token counts. TF-IDF was used to identify the most frequently occurring words in the dataset and assign weights to them based on their relevance to the overall document. N-grams were used to capture the contextual information of the reviews by identifying the sequences of words that occur frequently together. CountVectorizer was used to create a bag-of-words model that represented the occurrence of each word in the reviews.

Overall, these feature selection techniques helped us to identify the most important features from the textual data, which in turn improved the performance of our machine learning models in accurately classifying the reviews as genuine or fake.

**4.3.3 MACHINE LEARNING**

To identify phony reviews, the reviews are categorized as either computer-generated (CG) or human-generated (OR) reviews. The methodology is based on machine learning (ML) methods.

For the binary classification job, a number of well-known ML methods is applied, including Multinomial Naive Bayes, Linear Support Vector Classifier (SVC), XGBoost, and Stochastic Gradient Descent (SGD). Using a preprocessed dataset made up of a selection of customer reviews, the algorithms were trained. TF-IDF, N-grams, and word embedding approaches were used to extract the features from the text data that were utilized to train the models.

**4.3.3.1 LOGISTIC REGRESSION**

Logistic Regression used to classify the customer reviews as either computer generated (CG) or original (OR) generated by the customer. The algorithm outputs the probability of the event occurring by utilizing a logistic function to model the connection between the independent factors and the dependent variable. The logistic function's parameters are estimated by the model using the maximum likelihood estimation method, which also seeks to reduce the difference between the anticipated and actual values. The logistic regression equation can be represented as follows:

y = 1 / (1 + e^ (-w\*x + b))

where y is the predicted output (in our case, the probability that a review is CG or OR), e is the base of the natural logarithm, x is the feature vector representing the review, w is the weight matrix, and b is the bias term. In order to reduce the error between the expected and actual outputs, the values of w and b are learnt throughout the training phase. The final decision is then made based on a threshold probability, where a review with a predicted probability greater than the threshold is classified as CG and a review with a predicted probability less than the threshold is classified as OR.

**4.3.3.2 MULTINOMIAL NAIVE BAYES (MNB)**

Multinomial Naive Bayes (MNB), a probabilistic classifier that employs the Bayes theorem to determine the likelihood that a given text belongs to each class, is a popular machine learning approach. The following graphic illustrates how the MNB equation looks:

P (c | d) =P (d | c) \* P(c) / P(d)

Where P (c | d) is the probability of class c given the document d, P (d | c) is the probability of the document given the class c, P(c) is the prior probability of class c, and P(d) is the likelihood of the document, c is the class (in this example, either CG or OR), and d is the document or text. The prediction of the class label is made by finding the class label with the maximum probability value.

**4.3.3.3 LINEAR SUPPORT VECTOR CLASSIFIER (SVC)**

Linear Support Vector Classifier (SVC) is used, another popular ML approach, tries to increase the gap between the positive and negative classifications. The linear SVC equation be visualized as follows:

y= w \* x + b

Where y is the predicted class, the bias term, x is the feature vector, and w is the weight vector. The hyperplane that divides the positive and negative class by the greatest margin is then described as the decision boundary. The nearest positive and negative examples, sometimes referred to as support vectors, are as far away from the hyperplane as possible once the weights and bias are computed.

**4.3.3.4 STOCHASTIC GRADIENT DESCENT (SGD)**

Stochastic Gradient Descent (SGD) is used as a Machine Learning technique for fake review detection. SGD is an optimization approach used to train a variety of models, including support vector machines, logistic regression, and linear regression. One must first calculate the gradient of the loss function with respect to the model parameters before updating the parameters in the opposite direction of the gradient. Until convergence or a certain number of iterations are attained, this procedure is repeated. Prior to training the model, the learning rate, a hyperparameter that controls the magnitude of the parameter update, must be established. For binary classification issues like the identification of bogus reviews, SGD was employed using a log loss function. The objective of the optimization is to minimize the log loss between the predicted probabilities and the true labels.

The equation for log loss is given by:

L (y, y') = -(y \* log(y') + (1 - y) \* log (1 - y'))

where y is the true label and y' is the predicted probability. The objective of the optimization is to minimize the average log loss over the training samples. The optimization process can be described mathematically as:

θ = θ - η \* ∇L(θ)

where ∇L(θ) is the gradient of the loss function with respect to the parameters, θ is the model parameters, η and is the learning rate. Try to reduce the log by changing the parameters in the gradient's opposite direction

**4.3.3.5 XGBOOST**

Finally, The XGBoost (eXtreme Gradient Boosting) method is used to distinguish between reviews that were created by computers (CG) and those that were created by customers themselves (OR). A sophisticated version of gradient boosting known as XGBoost has successfully detected bogus reviews among a variety of other machine learning issues. Got an Accuracy of 0.82 and F1 Score of 0.83.

It works by fitting multiple decision trees to the training data and then combining the predictions of these trees through an ensemble method. This is done using an ensemble method. A gradient descent-based optimization approach is used to develop the trees in order to minimize the objective function, this frequently corresponds to the data's negative log-likelihood. Using the goal function's gradient as a basis, the algorithm iteratively modifies the weights of the trees in an effort to identify the optimal weights that produce the lowest objective value.

The learning rate, the number of trees, and the depth of the trees may all be adjusted with XGBoost's highly flexible and configurable features. Used a grid search to compare several combinations of these factors in order to determine which one produced the highest classification accuracy on the test data. This grid search's outcomes enabled us to develop a final XGBoost model that successfully identified whether reviews were (CG) or (OR).

**4.3.4 LABEL ENCODING**

Label Encoding is a technique used in machine learning and data science to convert categorical or textual data into numerical data, so that it can be used in machine learning models. In label encoding, each unique value in a categorical variable is assigned a numerical value.

**Input = Text**

[love, this, well, made, sturdy, very, comfortable, very, pretty]

[love, it, great, upgrade, original, mine, couple, years]

[pillow, saved, back, love, look, feel, pillow]

………

**Ouput:**

array([list([24558, 40846, 44429, 24909, 39230, 9529, 24558, 21999, 31647]),

list([24558, 21884, 18367, 43098, 29099, 26263, 10482, 45623]),

list([30655, 35320, 4917, 24558, 24424, 15843, 30655]), .........,)

**4.3.5 DEEP LEARNING**

Convolutional neural networks (CNNs) and long short-term memory are two deep learning approaches used in this project (LSTM) networks. As a result of their ability to capture both local and sequential relationships in data, these techniques are among the most widely used deep learning models for text categorization.

**4.3.5.1 CONVOLUTIONAL NEURAL NETWORKS (CNN)**

A CNN model with six layers is implemented.

The first layer of the CNN is an Embedding layer. Each word in the review is translated into a numerical representation by this layer. A series of numbers representing each individual word in the vocabulary serve as the input to this layer. This layer produces a dense vector representation of the words, where each vector component represents a word characteristic.

The second layer is a Conv1D layer, this layer's goal is to take the dense vector representation of the words and extract characteristics from it. In this layer, ReLU is employed as the activation function. This layer applies filters to the input, and the output characteristics are passed on to the following layer.

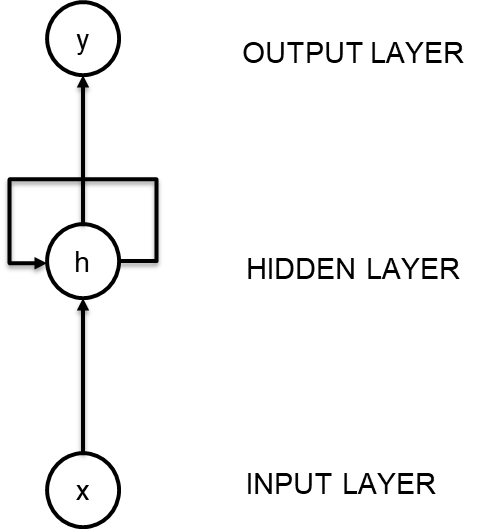
The third layer is a MaxPooling1D layer, which performs max pools the features that the Conv1D layer has extracted. By choosing the highest value from a range of values, the max pooling approach is used to minimize the number of dimensions in the data. From each set of values, the highest value is picked and passed to the following layer in this layer.

The fourth layer is a GlobalMaxPooling1D layer, which applies global maximum pooling to the MaxPooling1D layer's output. By choosing the highest value out of every value, the global max pooling approach reduces the dimensionality of the data. The highest value among all the values is chosen in this layer and passed to the following layer.

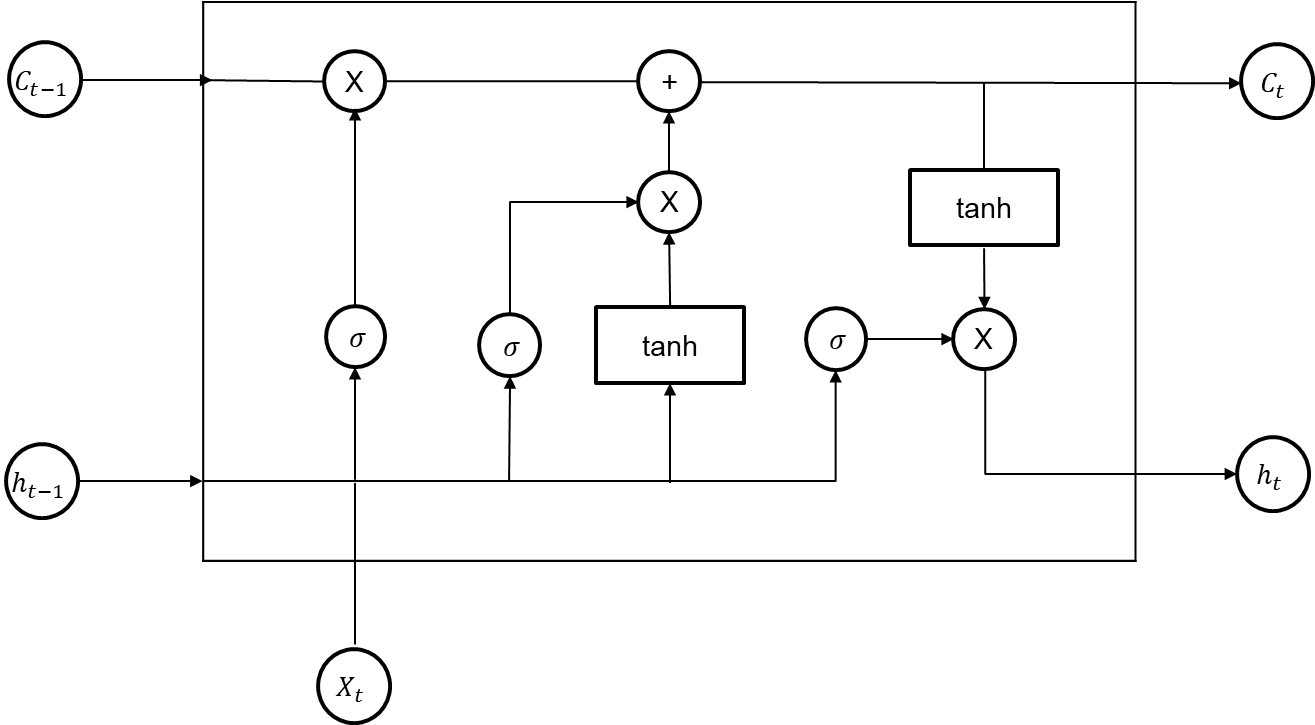
The fifth layer is a Dense layer, which performs a dense matrix multiplication on the input. This layer's objective is to learn intricate representations of the incoming data. In this layer, a bias is introduced to the output from the layer below after it has been multiplied by a weight matrix.

The sixth layer is a Dense\_1 layer, thus the layer's function is to anticipate things depending on the input. The final prediction is produced after the output from the previous layer has been run through a sigmoid activation function in this layer.

**4.3.5.2 LSTM WITH RNN**

Recurrent neural networks work well with sequential data (in this case, a sequence of words). An RNN is different from a feed-forward neural network in that it does not input an entire example at once. By integrating new data with the previously processed information, it processes sequences element-by-element. As humans, read a sentence word-by-word in order, incorporating each new word into the meaning of what have read so far.

**Fig 4.5 RNN Architecture**

****Further improvements can be made to vanilla RNNs using LSTMs. Even though RNNs are theoretically capable of retaining information for an extended period, they are extremely difficult to train to learn long-term dependencies, especially in extremely lengthy sentences and paragraphs. Special mechanisms have been designed for LSTMs to allow past data to be re-used later. Thus, LSTMs are almost always more suitable than vanilla RNNs in practice. The model was trained with a batch size of 64, using the Adam Optimizer.

**Fig 4.6 LSTM Cell Architecture**

There are six layers in the model, including an Embedding layer, LSTM layer, two Dense layers, and a Dropout layer comprised using the Sequential API. An LSTM layer is a recurrent neural network which is good choice for processing sequential data, because its 128 units use the sigmoid activation function for memory cells and the tanh activation function for hidden state.

The two Dense layers employ the rectified linear unit activation function to further reduce overfitting (ReLU), while the Dropout layer uses a rate of 0.3. For binary classification, the final layer uses a single Dense unit and a sigmoid activation function. Using accuracy measures, binary cross-entropy loss, and the Adam optimizer, the model is constructed. ModelCheckpoint is used to save the best weights, and EarlyStopping is used to stop training when accuracy does not improve, a reduced learning rate would be applied if the validation loss plateaued, and a termination would be applied if Nan values were encountered.

**4.3.5.2 CCN+LSTM:**

The CNN+LSTM model is a hybrid architecture that combines the strengths of both convolutional neural networks (CNNs) and long short-term memory (LSTM) networks.

The CNN+LSTM model consists of an embedding layer, followed by a 1D convolutional layer, a max pooling layer, and an LSTM layer, with a dense layer at the end for classification. The embedding layer learns a dense representation of words, while the convolutional layer extracts features from the learned word embeddings. The max pooling layer then downsamples the features, and the LSTM layer processes the sequences of features to capture temporal dependencies.

During training, the RMSprop optimizer were used with a learning rate of 0.01 and binary cross-entropy loss. To prevent overfitting, including dropout in the LSTM layer and used early stopping with a patience of 20 epochs. Also used model checkpointing to save the best model based on validation accuracy.

**4.4 SUMMARY**

The system for fake review detection in online product evaluations employs natural language processing and machine learning techniques to automatically identify fraudulent reviews and offer accurate insights into the quality of the product being evaluated. The system is made to analyze massive amounts of textual material from amazon product review and carry out thorough evaluations analysis. The algorithm can reliably determine whether reviews are real or fraudulent by extracting important characteristics from them, such as sentiment, relevancy, and authenticity. By this project, the aim to contribute to the continuing fight against fraudulent reviews and enhance consumer online buying.

**CHAPTER – 5**

**SYSTEM IMPLEMENTATION**

**5.1 OVERVIEW**

In this chapter various algorithm involved in implementing the modules described in the proposed system are dissolved

**5.2 DATA PRE-PROCESSING**

**Algorithm:** Data preprocessing

**Input:** Raw Text

**Output:** Pre-Processed Text

Begin

PUNCT\_TO\_REMOVE ⟵ set of punctuation marks

STOPWORDS ⟵ set of stop words

stemmer ⟵ an instance of a stemming algorithm

lemmatizer ⟵ an instance of a lemmatization algorithm

FUNCTION remove\_punctuation(text)

RETURN text with all punctuation marks removed

FUNCTION clean\_contractions(text, mapping)

RETURN text with contractions expanded and special characters removed

FUNCTION remove\_stopwords(text)

RETURN text with all stop words removed

FUNCTION stem\_words(text)

RETURN text with all words stemmed

FUNCTION lemmatize\_words(text)

RETURN text with all words lemmatized

FUNCTION remove\_urls(text)

RETURN text with all URLs removed

FUNCTION remove\_html(text)

RETURN text with all HTML tags removed

FUNCTION preprocess(text)

text ⟵ clean\_contractions(text, mapping)

text ⟵ text.lower()

text ⟵ remove\_urls(text)

text ⟵ remove\_html(text)

text ⟵ remove\_stopwords(text)

text ⟵ remove\_punctuation(text)

text ⟵ lemmatize\_words(text)

text ⟵ stem\_words(text)

RETURN text

End

A text data preparation pipeline that is frequently used for applications that involve natural language processing, such text classification or sentiment analysis. The pipeline has a number of routines for preprocessing and cleaning raw text data. First, all punctuation is removed from the text; second, contractions like "don't" are changed to their expanded versions, such as "do not." The final task eliminates stop words like "the," "and," "or," etc. The fifth function removes URLs from the text, the sixth function removes HTML elements from the text, and the fourth function substitutes " " with a blank space. The latter two operations, known as stemming and lemmatization, respectively, reduce words to their basic form. The text data is helped to be standardized by this preprocessing procedure, making it easier to analyze and model using machine learning algorithms.

**5.3 FEATURE SELECTION**

**Algorithm:** Feature Selection

**Input:** Pre-Processed Text

**Output:**

**LABEL ENCODING**

**Algorithm:** Label Encoding

**Input:**

**Output:**

Begin

Import sklearn.preprocessing.LabelEncoder

le ⟵ LabelEncoder()

all\_words ⟵ [word for word\_list in word\_lists for word in word\_list]

le.fit(all\_words)

labelled\_lists ⟵ []

for word\_list in word\_lists:

labelled\_list ⟵ le.transform(word\_list)

labelled\_lists.append(labelled\_list)

End

The above code shows the implementation of the LabelEncoder class from the scikit-learn library to encode a list of words into numeric labels. First, the module 'sklearn.preprocessing.LabelEncoder' is imported. Then, an instance of the LabelEncoder class is initialized and stored in the variable 'le'. Next, a list of all the words is created using list comprehension and stored in the variable 'all\_words'. The LabelEncoder is then fitted on all the words using the 'fit' method of the 'le' object. Finally, each list of words is transformed into a list of labels using the 'transform' method of the 'le' object, and the resulting labelled lists are stored in the 'labelled\_lists' variable.

**DEEP LEARNING**

**Algorithm:** Deep Learning

Input:

Output:

y\_pred ⟵ model.predict(X\_test)

**# Convert predicted values to binary 0/1 predictions**

y\_pred ⟵ np.round(y\_pred)

**# Generate classification report and confusion matrix**

class\_report ⟵ classification\_report(y\_test, y\_pred)

cm ⟵ confusion\_matrix(y\_test, y\_pred)

**# Calculate precision, recall, and accuracy values from the confusion matrix**

tn, fp, fn, tp ⟵ cm.ravel()

precision ⟵ tp / (tp + fp)

recall ⟵ tp / (tp + fn)

accuracy ⟵ (tp + tn) / (tp + tn + fp + fn)

**# Print classification report and precision/recall/accuracy values**

Print class\_report

Print Precision

Print Recall

Print Accuracy

**# Plot confusion matrix**

**#plt.imshow(cm, cmap=plt.cm.Blues)**

sns.heatmap(cm,annot=True,cmap='coolwarm',xticklabels=[0,1],fmt='d',annot\_kws={"fontsize":19})

plt.title("Confusion Matrix")

**#plt.colorbar()**

plt.xlabel("Predicted Label")

plt.ylabel("True Label")

plt.xticks([0, 1], labels=["Negative", "Positive"])

plt.yticks([0, 1], labels=["Negative", "Positive"])

plt.show()

The method constructs a trained model to implement a classification task. The variables X test and y test, which represent the input test data and true labels, respectively, are presumed to be available. The predict() function, which gives a probability value for each sample, is used by the model to generate predicted values for the test data. Binary forecasts are then created by rounding these probabilities to the nearest integer value (0 or 1). Using the actual labels and expected labels, the classification report and confusion matrix are produced. Precision, recall, and accuracy values are computed from the confusion matrix as measures of the effectiveness of the classification model. The categorization report and these values are both printed out. Lastly, a heatmap depiction of the confusion matrix is created .

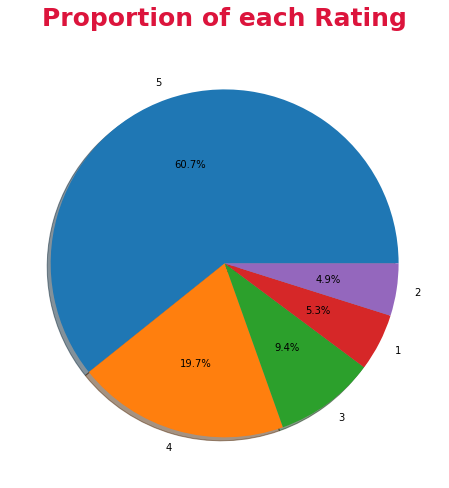
**CHAPTER – 6**

**RESULT AND DISCUSSION**

**6.1 OVERVIEW**

This chapter discusses about the results obtained by implementing the proposed methodologies. The performance of the Machine Learning and Deep Learning is compared and the result are evaluated.

**6.2 DATASET DESCRIPTION**

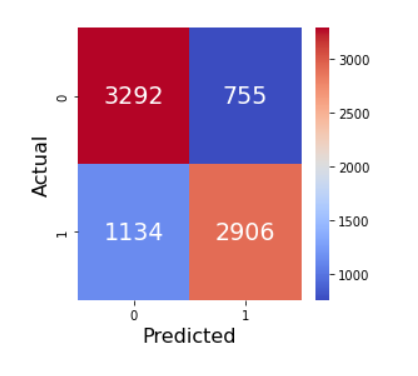
 The dataset is obtained from the amazon dataset from Kaggle. It contains 40400 rows X 4 columns. The dataset contains Categories, Rating, Label, Review, and Texts. This dataset was created by Salminen, J., Kandpal, C., Kamel, A. M., Jung, S., & Jansen, B. J. [16]. Journal of Retailing and Consumer Services, 64, 102771. Salminen et al. created fictitious customer reviews using the Universal Language Model Fine-tuning (ULMFiT) and Generative Pre-trained Transformer 2 (GPT-2). These models were chosen because they rely on transfer learning and are available as open-source code. The models were first tailored for the specific job of creating false reviews using a huge text corpus to teach them about the basic features and characteristics of language. Both ULMFiT and GPT-2 represent several NLP architecture types; ULMFiT employs an LSTM backbone whereas GPT-2 employs a transformer-based architecture.

**Fig 6.1 Dataset Proportion of each Rating**

**6.3 MODULES**

**6.3.2 LOGISTIC REGRESSION**

Logistic regression is a simple and widely used algorithm for binary classification tasks. It is used to predict whether a product review on Amazon is genuine or fake. After pre-processing the data and vectorizing the text data using TF-IDF, the logistic regression model was trained on the training set and evaluated its performance on the validation and test sets.

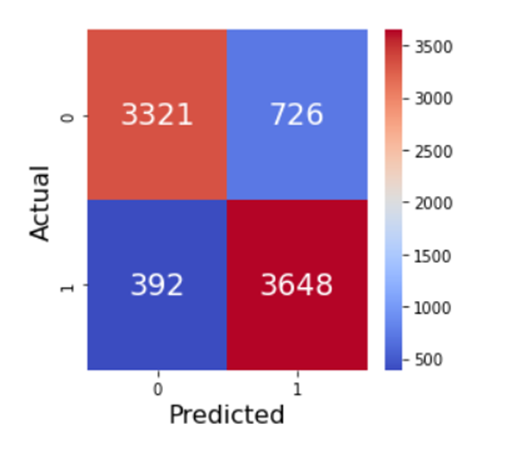
The logistic regression model achieved an accuracy of 0.77 and an F1 score of 0.75 on the test set. The precision and recall for each class were also reported in the classification report.

**Fig: 6.2 Logistic Regression Confusion Matrix**

The logistic regression model achieved an accuracy of 0.77 and an F1 score of 0.75, which indicates that the model performed reasonably well in identifying fake reviews. The precision and recall for both classes (genuine and fake) are also quite good, with values ranging from 0.72 to 0.81.

However, that logistic regression is a linear classifier and may not be able to capture complex non-linear relationships between the input features and the target variable. Therefore, the model may not perform well when dealing with more complex datasets with a higher degree of variability.

**6.3.3 MULTINOMIALNB**

 MultinomialNB is a type of Naive Bayes algorithm that is suitable for classification tasks with discrete features such as text classification.

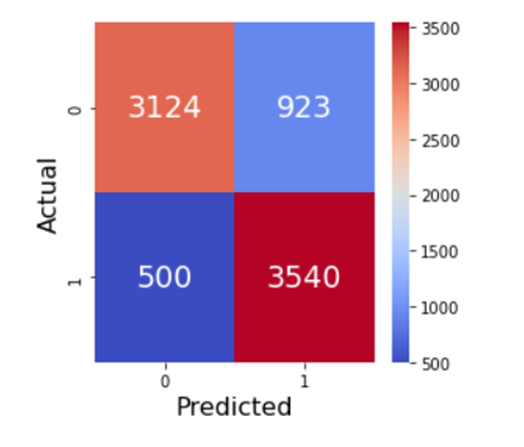
**Fig: 6.3 MultinominalNB Confusion Matrix**

After training and testing the model, an accuracy of 0.86is obtained, which means that the model was able to classify 86% of the reviews correctly. The F1 score, which is a weighted average of precision and recall, was 0.87, indicating that the model had a good balance between precision and recall.

Looking at the precision and recall scores for each class, the model had a slightly higher precision for class 0 (genuine reviews) and a slightly higher recall for class 1 (fake reviews). This means that the model was slightly better at identifying genuine reviews than fake reviews, but was still able to identify a high proportion of fake reviews.

Overall, the performance of the MultinomialNB model was quite good, with an accuracy of 0.86 and an F1 score of 0.87. This indicates that the model is a suitable choice for classifying Amazon product reviews as genuine or fake.

**6.3.4 XGBClassifier**

 The XGBClassifier was another model used to classify fake and real reviews. The results obtained from this model showed an accuracy of 0.83.

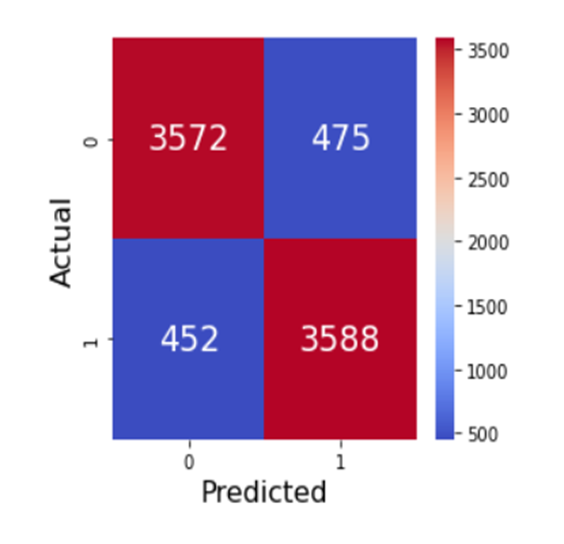
**Fig: 6.4 XGBClassifier Confusion Matrix**

The F1 score obtained was also 0.83, which indicates a balanced trade-off between precision and recall. Looking at the precision and recall values in the report, that the model achieved a precision of 0.86 for class 0 (real reviews) and a precision of 0.80 for class 1 (fake reviews). The recall value for class 0 was 0.78, while the recall value for class 1 was 0.87.

Overall, the XGBClassifier performed well in classifying fake and real reviews. The results were comparable to those obtained from the other models used in the project, such as logistic regression and MultinomialNB.

**6.3.5 LinearSVC**

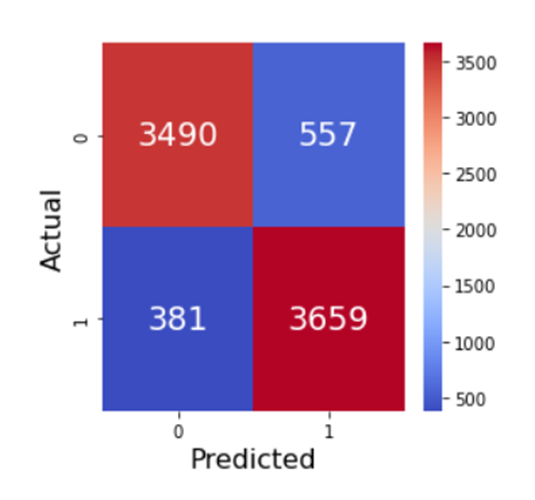
The linear SVC model achieved an accuracy score of 0.89 and an F1 score of 0.89 on the test dataset. The precision and recall values for both classes were also high, indicating that the model was able to classify both positive and negative reviews accurately.

**Fig: 6.5 LinearSVC Confusion Matrix**

The model performed similarly well to the XGBoost classifier, and outperformed the logistic regression and Multinomial Naive Bayes models. The linear SVC model was trained on a bag-of-words representation of the reviews, with TF-IDF weighting. Overall, the linear SVC model proved to be a strong performer in the sentiment analysis task.

**6.3.6 SGD**

The results indicate that the SGD classifier achieved good performance in predicting the sentiment of the text data. It is worth noting that the accuracy and performance of the model may differ depending on the dataset used and the hyperparameters selected during model training. It is crucial to evaluate the model's performance on an independent test set to ensure that the results are not overfitting to the training data and are generalizable

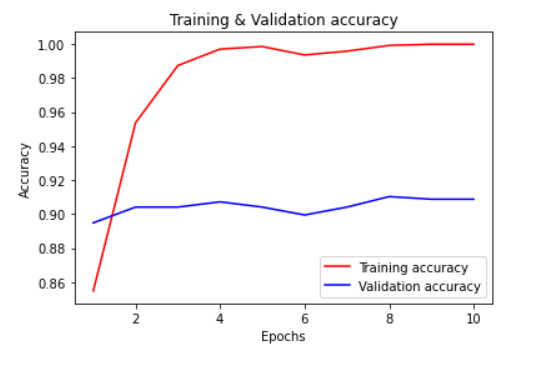


**Fig: 6.6 SGD Confusion Matrix**

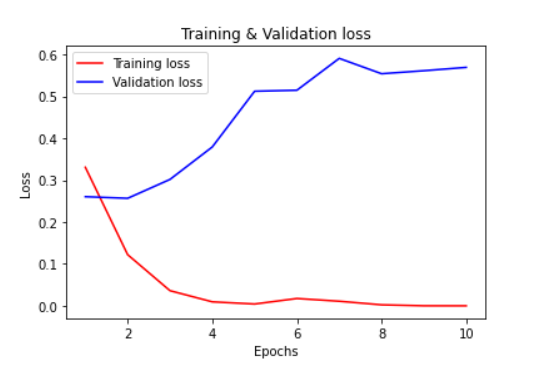
Overall, the SGD classifier's precision, recall, and F1 scores were consistently high, with an accuracy of 0.88 and an F1 score of 0.89. The precision of the classifier was 0.87 for the positive sentiment class and 0.90 for the negative sentiment class. The recall of the classifier was 0.91 for the positive sentiment class and 0.86 for the negative sentiment class. These metrics indicate that the SGD classifier was able to accurately classify positive and negative sentiments in the text data.

The results suggest that the SGD classifier is a promising model for predicting sentiment in text data. However, it's crucial to consider the limitations and potential sources of error, such as the impact of the specific dataset and hyperparameters on model performance, to ensure reliable and accurate predictions.

**6.3.7 CNN**

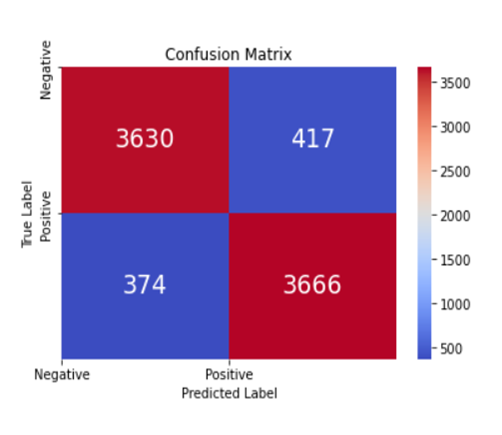
The CNN model achieved a test accuracy of 0.5546 and a validation accuracy of 0.8902, which indicates that the model is performing well in classifying the sentiment of the text data. The precision and recall values for both classes are also high, indicating that the model is able to correctly identify positive and negative sentiments in the data.

**Fig: 6.7 Training & Validation Accuracy of CNN**



**Fig: 6.8 Training & Validation Loss of CNN**

However, the model's training accuracy of 100.0% suggested overfitting to the training data. Techniques such as regularization or increasing the amount of training data could be applied to address this issue

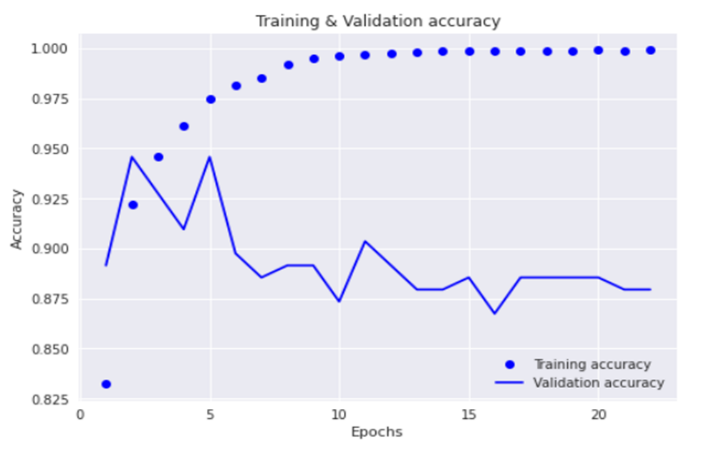


**Fig 6.9 CNN Confusion Matrix**

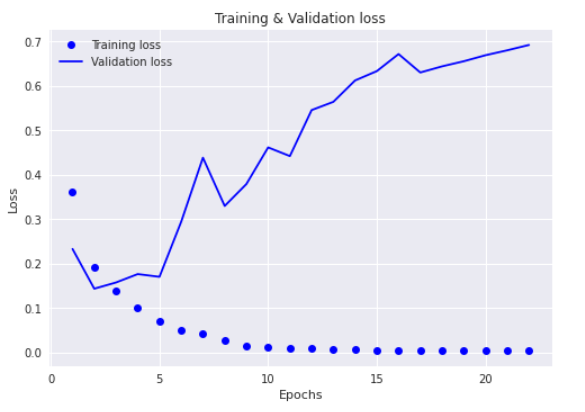
The precision values (0.89 for both classes) indicate that when the model predicts a sentiment as either positive or negative, it is correct 89% of the time. The recall values (0.88 for negative sentiment and 0.90 for positive sentiment) indicate that the model correctly identifies 88% of negative sentiment instances and 90% of positive sentiment instances.

The F1-score is a weighted average of the precision and recall values, and ranges between 0 and 1, where 1 indicates perfect precision and recall. The F1-score for both classes is 0.89, indicating good performance by the model. Overall, the CNN model with CountVectorizer feature extraction performed well in predicting the sentiment of the text data.

**LSTM:**

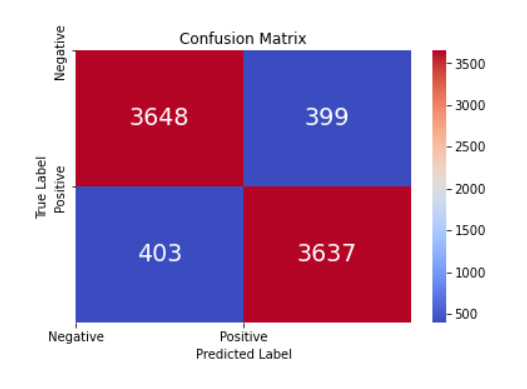
During training, the LSTM model had a best validation accuracy of 94.58% and a best training accuracy of 99.91%. The model's accuracy, precision, and recall on the test set were 90.90%, 90.35%, and 91.72%, respectively. According to the classification report, the model performed similarly for both classes, earning F1 scores of 0.91 for each class.

**Fig: 6.10 Training & Validation Accuracy of LSTM**



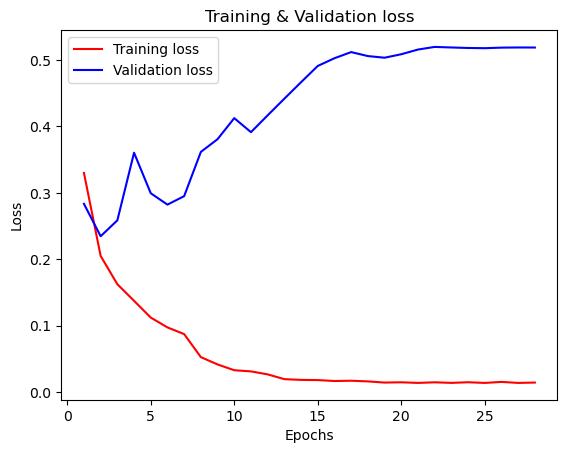
**Fig: 6.11 Training & Validation Loss of LSTM**

The LSTM model obtaining an accuracy of 90.90% on the test set. This suggests that the model can accurately and precisely classify the sentiment of a given review. The model can properly detect both positive and negative emotion, with just a little difference between the two classes, based on its accuracy and recall scores.

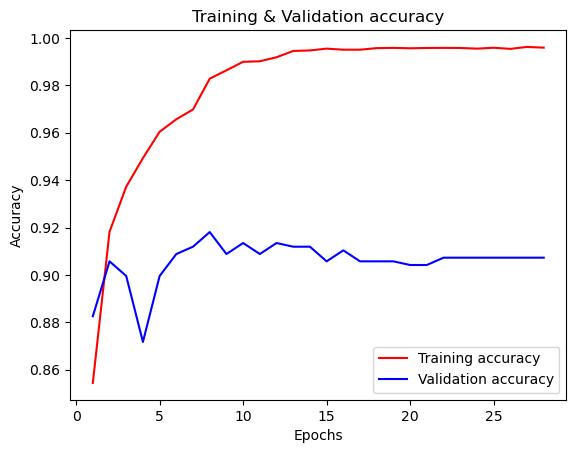
**Fig: 6.12 LSTM Confusion Matrix**

**CNN+LSTM:**

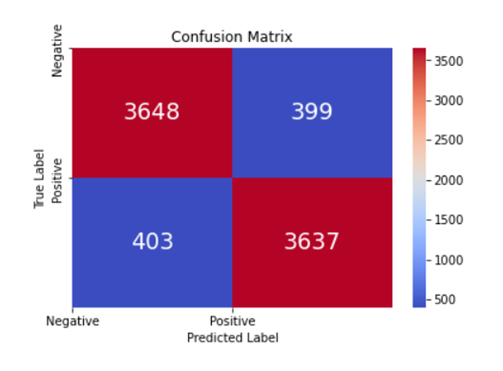
The CNN+LSTM model has a Best Training Accuracy of 99.62% and a Best Validation Accuracy of 91.81%, showing good generalization on the unseen validation data. The model's Test Accuracy was 91.42%, which although being a little lower than the Best Validation Accuracy shows that the model performed well on the test data.



**Fig: 6.13 Training & Validation Loss of CNN+LSTM**



**Fig: 6.14 Training & Validation Accuracy of CNN+LSTM**

The model obtained scores for the positive class of 0.92, 0.91, and 0.91 for Precision, Recall, and F1-score, respectively (class 1). This indicates that 92% of the tweets the model properly identified as being hateful or offensive (Recall) and that 92% of the tweets it incorrectly classed as such were, in fact, hateful or offensive (Precision).

**Fig: 6.15 CNN+LSTM Confusion Matrix**

In terms of Accuracy, the model performed considerably better than the LSTM model, but the difference was not very large. However, the CNN+LSTM model outperformed the LSTM model in terms of loss value, showing that it was more effective in minimizing the discrepancy between the predicted and real labels.

The model achieved an overall weighted average of 0.91 for each of the metrics of recall, precision, and F1-score. This shows that the model did a good job of accurately categorizing both classes.

**EXPERIMENTAL RESULTS:**

**Table 1. Precision, Recall and F1-Score for Machine Learning**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Label** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| Logistic  Regression | 0 | 0.74 | 0.81 | 0.78 | 0.77 |
| 1 | 0.79 | 0.72 | 0.75 |
| MultinomialNB | 0 | 0.89 | 0.82 | 0.86 | 0.86 |
| 1 | 0.83 | 0.90 | 0.87 |
| XGBoost | 0 | 0.86 | 0.77 | 0.81 | 0.82 |
| 1 | 0.79 | 0.88 | 0.83 |
| LinearSVC | 0 | 0.89 | 0.88 | 0.89 | 0.89 |
| 1 | 0.88 | 0.89 | 0.89 |
| SGD | 0 | 0.90 | 0.86 | 0.88 | 0.88 |
| 1 | 0.87 | 0.91 | 0.89 |

**Table 2. Deep Learning Models Validation Accuracy**

|  |  |
| --- | --- |
| **Model** | **Validation Accuracy** |
| CNN | 0.91 |
| LSTM | 0.94 |
| CNN+LSTM | 0.91 |

**Table 3. Precision, Recall and F1-Score for Deep Learning**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Label** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| CNN | 0 | 0.89 | 0.88 | 0.89 | 0.89 |
| 1 | 0.89 | 0.90 | 0.89 |
| LSTM | 0 | 0.91 | 0.90 | 0.91 | 0.90 |
| 1 | 0.90 | 0.92 | 0.91 |
| CNN + LSTM | 0 | 0.91 | 0.92 | 0.91 | 0.91 |
| 1 | 0.92 | 0.91 | 0.91 |

**Chapter 7**

**CONCLUSION AND FURTURE WORK**

**7.1 CONCLUSION:**

In Conclusion, assessed the efficacy of several deep learning and machine learning models for a specific NLP job. The outcomes demonstrated that the Convolutional Neural Network combined with Long Short-Term Memory (LSTM), which has an accuracy of 91%, outperformed all other deep learning models in terms of accuracy and F1 score. Traditional machine learning models, on the other hand, with an accuracy range of 77% to 89%, included Logistic Regression, Multinomial Naive Bayes, XGBoost, Linear Support Vector Machines, and Stochastic Gradient Descent.

The Deep Learning models outperform typical machine learning models in complicated tasks, as shown by the comparison of the two types of models. This is explained by deep learning model’s capacity to automatically discover intricate representations and feature interactions from the input data.

In addition to the Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN) networks were used. These models, which were created to deal with sequential data, did well, but not as well as the CNN+LSTM.

In conclusion, experiment shows how well deep learning models specifically the CNN+LSTM perform NLP tasks and emphasizes the value of applying deep learning methods to practical NLP applications. By adjusting the hyperparameters, utilizing other architectures, or adding extra data sources, further advancements can be realized.

**7.2 FUTURE ENHANCEMENTS**

Alternative pre-processing methods can be used to increase the accuracy of the models, try using various pre-processing methods including stemming, lemmatization, and stop-word elimination. Creating word vectors using pre-trained word embeddings, such Word2Vec or GloVe, could enhance the performance of the models. Explore additional deep learning models, such as transformer models like BERT or GPT, which have demonstrated cutting-edge performance in a number of natural language processing applications. The accuracy of the models could be increased by expanding the dataset. To generate synthetic data, can either gather more data or apply data augmentation techniques. The performance of the models could be enhanced by carefully adjusting hyperparameters like learning rate, batch size, and dropout rate. Grid search or random search could be used to enhance the hyperparameters and may make it available for users to utilize as a web application or a mobile application.

**7.3 SOCIAL IMPACT**

Fake Review Detection might potentially enhance sales for firms with real favorable reviews by accurately identifying fake reviews, which would increase consumer confidence in the reviews and allow them to make more educated purchasing decisions.

It may also assist companies in developing more sincere and pleasant interactions with their clients by lowering the prevalence of fake reviews, which may result in greater customer loyalty and favorable word-of-mouth.

The idea might also contribute to a more wholesome and reliable online environment by promoting transparency and honesty in online evaluations as well as preventing the spread of incorrect information online.

**7.4 ECONOMIC ASPECT:**

**Improved Customer Experience:** By detecting and filtering out fake reviews, it can help to ensure that the customers make informed decisions, leading to increased trust in the platform and potentially more revenue for businesses.

**Cost Savings:** For businesses, detecting fake reviews can be a time-consuming and resource-intensive task. By automating this process with machine learning, it can potentially save businesses significant costs associated with manual review monitoring.

**Increased Platform Reliability:** By identifying and eliminating fraudulent reviews, it might raise the platform's trust, which might encourage more users to use it and generate more income.

**APPENDIX – A**

**SYSTEM REQUIREMENTS**

**HARDWARD REQUIREMENTS**

Processor: AMD Ryzen 4800H

RAM: 8GB

**SOFTWARE REQUIREMENTS**

Operating System: Windows 11

Language used: Python 3.10

Platform:

Working environment: Kaggle

**APPENDIX – B**

**SOURCE CODE**

**# Removing all punctuations from review text**

PUNCT\_TO\_REMOVE = string.punctuation

def remove\_punctuation(text\_):

return text\_.translate(str.maketrans('', '', PUNCT\_TO\_REMOVE))

def clean\_contractions(text\_, mapping):

specials = ["’", "‘", "´", "`", "\_"]

for s in specials:

if s == "\_":

text\_ = text\_.replace(s, " ")

else:

text\_ = text\_.replace(s, "'")

text\_ = ' '.join([mapping[t] if t in mapping else t for t in text\_.split(" ")])

return text\_

**#Remove Stopwords from text**

from nltk.corpus import stopwords

STOPWORDS = set(stopwords.words('english'))

def remove\_stopwords(text\_):

return " ".join([word for word in str(text\_).split() if word not in STOPWORDS])

def word\_replace(text\_):

return text\_.replace('<br />','')

**#Stemming**

from nltk.stem.porter import PorterStemmer

stemmer = PorterStemmer()

def stem\_words(text\_):

return " ".join([stemmer.stem(word) for word in text\_.split()])

**#Lemmatization**

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

def lemmatize\_words(text\_):

return " ".join([lemmatizer.lemmatize(word) for word in text\_.split()])

**#Remove URL & HTML tag from text**

def remove\_urls(text\_):

url\_pattern = re.compile(r'https?://\S+|www\.\S+')

return url\_pattern.sub(r'', text\_)

def remove\_html(text\_):

html\_pattern = re.compile('<.\*?>')

return html\_pattern.sub(r'', text\_)

def preprocess(text\_):

text\_=clean\_contractions(text\_,mapping)

text\_=text\_.lower()

text\_=word\_replace(text\_)

text\_=remove\_urls(text\_)

text\_=remove\_html(text\_)

text\_=remove\_stopwords(text\_)

text\_=remove\_punctuation(text\_)

# review=stem\_words(review)

text\_=lemmatize\_words(text\_)

return text\_

**REFERENCES:**

1) M. Abdulqader, A. Namoun, and Y. Alsaawy, "Fake online reviews: A unified detection model using deception theories," IEEE Access, vol. 7, pp. 68596-68608, 2019.

2)S. Dhawan, S. C. R. Gangireddy, S. Kumar, and T. Chakraborty, “Spotting collective behavior of online frauds in customer reviews,” 2019, arXiv:1905.13649. [Online].

3)B. Liu, M. Hu, and J. Cheng, “Opinion observer: Analyzing and comparing opinions on the Web,” in Proc. 14th Int. Conf. World Wide Web (WWW), 2005, pp. 342–351.

4)N. N. Ho-Dac, S. J. Carson, and W. L. Moore, “The effects of positive and negative online customer reviews: Do brand strength and category maturity matter?” J. Marketing, vol. 77, no. 6, pp. 37–53, Nov. 2013.

5)F. Li, M. Huang, Y. Yang, and X. Zhu, “Learning to identify review spam,” in Proc. 22nd Int. Joint Conf. Artif. Intell., 2011, pp. 2488–2493.

6)E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, “Detecting product review spammers using rating behaviors,” in Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2010, pp. 939–948

7)G. Wang, S. Xie, B. Liu, and P. S. Yu, “Review graph-based online store review spammer detection,” in Proc. IEEE 11th Int. Conf. Data Mining, Dec. 2011, pp. 1242–1247

8)A. Mukherjee, V. V. Venkataraman, B. Liu, and N. S. Glance, “What yelp fake review filter might be doing?” in Proc. 7th Int. Conf. Weblogs Social Media, 2013, pp. 409–418.

9)C. Xu and J. Zhang, “Towards collusive fraud detection in online reviews,” in Proc. IEEE Int. Conf. Data Mining, Nov. 2015, pp. 1051–1056.

10)H. S. Dutta, V. R. Dutta, A. Adhikary, and T. Chakraborty, “HawkesEye: Detecting fake retweeters using Hawkes process and topic modeling,” IEEE Trans. Inf. Forensics Security, vol. 15, pp. 2667–2678, Jan. 2020. [Online].

11) Saleh Nagi Alsubari, Sachin N. Deshmukh Mosleh Hmoud Al-Adhaileh, Fawaz Waselalla Alsaadeand Theyazn H. H. Al Dhyani “Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multi Domain Datasets”, 15 April 2021.

12) Ahmed M. Elmogy, Usman Tariq, Atef Ibrahim “Fake Reviews Detection using Supervised Machine Learning” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 1, 2021.

13) Xinyue Wang, Xianguo Zhang, Chengzhi Jiang, Haihang Liu,” Identification of Fake Reviews Using Semantic and Behavioral Features”, 2018 4th IEEE International Conference on Information Management

14) Zhenning (Jimmy) Xu, Jun Ai, Fei Wang, “Identifying Indicators of Fake Reviews Based on Spammer’s Behavior Features”, 2017 IEEE International Conference on Software Quality, Reliability and Security (Companion Volume)

15)Faiza masoodi, Ghana Ammad, Ahamed Almogren “Spammer Detection And Fake User Identification On Social Networks”, May 22, 2019,

16)” Creating and Detecting Fake Review of Online Product” Author Links Open Overlay PanelJoni Saliminen A B, Chandrashekhar kandpal C, Ahmd Mohamed Kamel D, Soon-Gyo Jung A, Bernard J. Jansen