

American International University-Bangladesh (AIUB)

Faculty of Science and Technology

# Hybrid Model Implementation for Sentiment Analysis on Amazon Data

Rahaman, Hasibur (19-40823-2)

Sithi, Sharannaya Dey (19-40811-2)

Rahman, AKM Shahriyar (19-40787-2)

*A Thesis submitted for the degree of Bachelor of Science (BSc) in Computer Science and Engineering (CSE) at*

*American International University Bangladesh in October,2024*

Faculty of Science and Technology (FST)

## Abstract

The goal of this research is to create a hybrid sentiment analysis model that combines several machine learning and deep learning models to improve sentiment prediction accuracy. We compared and contrasted multiple models (VADER, RoBERTa, BERT, TextBlob, SVM, CNN, and LSTM) using product reviews from Amazon.com. We suggested a hybrid model that combines sentiment scores from these various models in order to overcome the shortcomings of the individual models. When compared to standalone models, the hybrid approach performed better, attaining the highest accuracy (77%), as well as better precision, recall, and F1 scores. The hybrid model outperforms well-performing models like SVM and RoBERTa, according to the evaluation metrics, which also showed that the hybrid model effectively balances precision and recall. Deep learning models, on the other hand, like CNN and LSTM, performed worse in recall and F1-measure, suggesting possible difficulties in detecting true positives. Our results show that combining the advantages of several models can greatly improve sentiment prediction for practical uses, like product reviews, and result in a more reliable and accurate framework for analysis.

## Declaration by author

This thesis is composed of our original work, and contains no material previously published or written by another person except where due reference has been made in the text. We have clearly stated the contribution of others to our thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in our thesis. The content of our thesis is the result of work we have carried out since the commencement of Thesis.

We acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate we have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rahaman, Hasibur  (19-40823-2)  Computer Science & Engineering  Faculty of Science & Technology | |  | Sithi, Sharannaya Dey  (19-40811-2)  Computer Science & Engineering  Faculty of Science & Technology | |
|  | | | | |
|  | Rahman, AKM Shahriyar  (19-40787-2)  Computer Science & Engineering  Faculty of Science & Technology | | |  |

## Approval

The thesis titled **“Hybrid Model Implementation for Sentiment Analysis on Amazon Data”** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelors of Science in Computer Science on (**01.10.2024**) and has been accepted as satisfactory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tohedul Islam  Assistant Professor, Faculty  Department Of Computer Science  American International University-Bangladesh | |  | S M Abdullah Shafi  Lecturer, Faculty  Department Of Computer Science  American International University-Bangladesh | |
|  | | | | |
|  | Name of the Head (UG/PG)  Rank & Position | | |  |
| Prof. Dr. Dip Nandi  Associate Dean  Faculty of Science and Technology  American International University-Bangladesh | |  | Mashiour Rahman  Sr. Associate Professor & Dean-in-charge  Faculty of Science and Technology  American International University-Bangladesh | |

## Contributions by authors to the thesis

List the significant and substantial inputs made by different authors to this research, work and writing represented and/or reported in the thesis. These could include significant contributions to: the conception and design of the project; non-routine technical work; analysis and interpretation of research data; drafting significant parts of the work or critically revising it so as to contribute to the interpretation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Rahaman, Hasibur** | **Sithi, Sharannaya Dey** | **Rahman, AKM Shahriyar** | **Contribution (%)** |
|  | *(19-40823-2)* | *(19-40811-2)* | *(19-40787-2)* |
| Conceptualization |  |  |  | 100 % |
| Data curation |  |  |  | 100 % |
| Formal analysis |  |  |  | 100 % |
| Investigation |  |  |  | 100 % |
| Methodology |  |  |  | 100 % |
| Implementation |  |  |  | 100 % |
| Validation |  |  |  | 100 % |
| Theoretical derivations |  |  |  | 100 % |
| Preparation of figures |  |  |  | 100 % |
| Writing – original draft |  |  |  | 100 % |
| Writing – review & editing |  |  |  | 100 % |

## Acknowledgments

We would like to express our gratitude to Tohedul Islam, Assistant Professor, Faculty of the Department of Computer Science, and our thesis advisor, for his unwavering support and guidance. We sincerely appreciate all his advice and encouragement throughout our research.

We are very thankful for all his help, guidance, and motivation during our study. Sir has been an exceptional mentor, and his expertise, dedication, and constructive feedback have been invaluable. Our academic goals and ideas have been greatly aided by his support. We have learned a lot, and used our knowledge and insights to challenge ourselves and strive for greatness. We are extremely fortunate to have had the opportunity to work with someone so extraordinary, and we will be eternally grateful to him for his guidance, kindness, and unwavering dedication to our success.

## Keywords

Hybrid Model, Sentiment Analysis, Amazon Data, Vader, RoBERTa, BERT, CNN, LSTM, SVM, TextBlob

Table of Contents

[Hybrid Model Implementation for Sentiment Analysis on Amazon Data i](#_Toc178850631)

[Abstract ii](#_Toc178850632)

[Declaration by author iii](#_Toc178850633)

[Approval 4](#_Toc178850634)

[Contributions by authors to the thesis v](#_Toc178850635)

[Acknowledgments vi](#_Toc178850636)

[Keywords vii](#_Toc178850637)

[List of Figures x](#_Toc178850638)

[List of Abbreviations and Symbols xi](#_Toc178850639)

[Introduction 12](#_Toc178850640)

[1.1 Motivation 12](#_Toc178850641)

[1.2 Objective 13](#_Toc178850642)

[1.3 Research Questions 13](#_Toc178850643)

[1.4 Document Outline 13](#_Toc178850644)

[Literature review 14](#_Toc178850645)

[2.1 Introduction 14](#_Toc178850646)

[2.1 VADER 14](#_Toc178850647)

[2.2 TextBlob 14](#_Toc178850648)

[2.3 BERT 15](#_Toc178850649)

[2.4 RoBERTa 15](#_Toc178850650)

[2.5 CNN 15](#_Toc178850651)

[2.6 LSTM Networks 15](#_Toc178850652)

[2.7 SVM 15](#_Toc178850653)

[2.8 TF-IDF 16](#_Toc178850654)

[2.9 Hybrid Models 16](#_Toc178850655)

[2.10 Related Works 16](#_Toc178850656)

[Methods 19](#_Toc178850657)

[3.1 Programming Environment and Technologies 20](#_Toc178850658)

[3.2 Data Collection 20](#_Toc178850659)

[3.3 Data Preprocessing 21](#_Toc178850660)

[3.3.1 Import Datasets 21](#_Toc178850661)

[3.3.2 Setup Preprocess 22](#_Toc178850662)

[3.3.3 Apply Preprocess and Store Data 23](#_Toc178850663)

[3.4 Splitting Data 23](#_Toc178850664)

[3.5 Model Building 23](#_Toc178850665)

[3.5.1 Transformer Models (RoBERTa, BERT) 24](#_Toc178850666)

[3.5.2 Lexicon Base Models (VADER, TextBlob) 26](#_Toc178850667)

[3.5.3 Machine Learning Model (SVM with TF-IDF vectorization) 27](#_Toc178850668)

[3.5.4 Deep Learning Models: (LSTM, CNN) 29](#_Toc178850669)

[3.5.5 Hybrid Model 30](#_Toc178850670)

[Results or findings 34](#_Toc178850671)

[4.1 Result Analysis 34](#_Toc178850672)

[4.1.1 Accuracy 34](#_Toc178850673)

[4.1.2 Precision (Macro) 35](#_Toc178850674)

[4.1.3 Precision (Micro) 36](#_Toc178850675)

[4.1.4 Precision (weighted) 36](#_Toc178850676)

[4.1.5 Recall 37](#_Toc178850677)

[4.1.6 F1-Score 38](#_Toc178850678)

[4.2 Comparison of Results: 38](#_Toc178850679)

[Discussion 39](#_Toc178850680)

[5.1 Introduction 39](#_Toc178850681)

[5.2 Limitations 39](#_Toc178850682)

[Conclusion 41](#_Toc178850683)

[6.1 Future Work 41](#_Toc178850684)

[6.2 Conclusion 41](#_Toc178850685)

[Bibliography 42](#_Toc178850686)

# List of Figures

[Figure-01:](#Figure_1) [Methodology 19](#Figure_1)

[Figure-02: Sample Dataset 21](#Figure_2)

[Figure-03: Import datasets 21](#Figure_3)

[Figure-04: Setup Pre-processed 22](#Figure_4)

[Figure-05: Apply Pre-processed and store 23](#Figure_5)

[Figure-06: Spliting Data 23](#Figure_6)

[Figure-07: Preparing transformer Models 24](#Figure_7)

[Figure-08: Bert model build 25](#Figure_8)

[Figure-09: roberta model build 25](#_Toc76225928)

[Figure-10: Vader Model Build 26](#Figure_10)

[Figure-11: Textblob model build 27](#Figure_11)

[Figure-12: SVM Model Build with TF-IDF Vectorization 28](#Figure_12)

[Figure-13: Prepare Deep Learning Model 29](#Figure_13)

[Figure-14: LSTM model build 29](#Figure_14)

[Figure-15: Cnn model build 30](#Figure_15)

[Figure-16: Hybrid model setup 31](#Figure_16)

[Figure-17: Hybrid model Build 32](#Figure_17)

[Figure-18: Final Implementation 33](#Figure_18)

[Figure-19: Accuracy Graph 35](#Figure_19)

[Figure-20: Precision (macro) Graph 36](#Figure_20)

[Figure-21: Precision (micro) Graph 36](#Figure_21)

[Figure-22: precision (weighted) graph 37](#Figure_22)

[Figure-23: Recall Graph 37](#Figure_23)

[Figure-24: Fi-score graph 38](#Figure_24)

# List of Abbreviations and Symbols

|  |  |
| --- | --- |
| Abbreviations | |
| VADER | Valence Aware Dictionary and sEntiment Reasoner |
| RoBERTa | Robustly Optimized BERT Pretraining Approach |
| BERT | Bidirectional Encoder Representations from Transformers |
| LSTM | Long Short-Term Memory |
| SVM | Support Vector Machine |
| CNN | Convolutional Neural Network |
| POS | Part of Speech (tagging) |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| ML | Machine Learning |
| DL | Deep Learning |

**Chapter 1**

# Introduction

It's now common and popular practice to read online customer reviews and ratings before making a purchase. Based on feedback, consumers are more likely to purchase a product. Because online marketplaces have grown in popularity over the past few decades, many online retailers and sellers ask their customers to leave reviews of the goods they have purchased. Millions of reviews are generated every day throughout the Internet regarding various goods, services, and locations. Due to this, the internet is the most reliable resource for ideas and reviews regarding a good or service. Product reviews offer valuable insights into a product's features, quality, and recommendations, helping potential buyers gain a comprehensive understanding of it. These reviews benefit not only consumers but also sellers, particularly those who manufacture their own products, by providing a better understanding of consumer preferences and needs. However, it is getting harder for a customer to decide whether or not to purchase a product as more reviews become available for it. Customers are more confused when trying to make the right decision due to conflicting opinions about the same product and unclear reviews. In this case, it appears that all e-commerce enterprises must analyze this content.

## 1.1 Motivation

The motivation for developing a hybrid sentiment analysis model stems from the need to enhance the accuracy, robustness, and scalability of existing sentiment analysis approaches. Individual models like VADER, RoBERTa, or BERT perform well in specific scenarios but struggle with diverse text formats, such as short tweets versus lengthy product reviews. The hybrid model leverages multiple models to provide a more comprehensive sentiment analysis, resulting in deeper insights, especially for business decision-making. By combining rule-based, machine learning, and deep learning methods, this model aims to overcome the limitations of standalone approaches and improve sentiment prediction in domains like e-commerce and product reviews.

## 1.2 Objective

The primary objective of this research is to develop a hybrid sentiment analysis model that integrates various sentiment analysis techniques (VADER, RoBERTa, BERT, SVM, CNN, LSTM, TextBlob) to:

* Increase sentiment prediction accuracy across diverse datasets.
* Provide actionable insights for businesses, particularly in the context of customer feedback and product improvement.
* Enhance decision-making processes by producing a robust system that can generalize well across various text types and domains (e.g., Amazon product reviews, social media).

## 1.3 Research Questions

When compared to individual models, how can a hybrid sentiment analysis model increase the overall accuracy of sentiment prediction?

Which model combinations work best for distinguishing between formal and informal language, short and long text, and other types of data in order to capture sentiment nuances?

How can sentiment analysis using this hybrid model help make better business decisions, especially when it comes to enhancing products and services?

Is it possible to apply a hybrid model for sentiment analysis to non-e-commerce domains like market research or social media monitoring?

## 1.4 Document Outline

The rest of the document is as follows. Chapter 2 provides a literature analysis of the sentiment analysis as well as relevant research, journals, and articles reviewed for the idea. Chapter 3 explains justification and explanation of the methodological approach. The results from the experiments are gathered in Chapter 4 and discussed in Chapter 5. Finally, Chapter 6 concludes the study.

**Chapter 2**

# Literature review

## Introduction

The Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that aims to identify and extract subjective information from text, such as opinions, emotions, and attitudes. Hybrid models, which combine multiple techniques or models, have emerged as a promising approach to improve sentiment analysis performance. This literature review explores recent research on hybrid models for sentiment analysis, focusing on the integration of semantic knowledgebases, machine learning algorithms, and deep learning techniques. The enumerated summaries of various papers within their field of study are provided below.

## VADER

VADER is a lexicon-based sentiment analysis tool specifically designed for social media text. It uses a predefined list of words and their associated sentiment scores, providing a simple yet effective way to gauge sentiment polarity (positive, negative, neutral). VADER's ability to capture sentiment in short, informal texts makes it particularly useful in real-time applications like social media monitoring and customer feedback analysis. Its effectiveness has been demonstrated in various studies, showcasing its role in hybrid models that enhance sentiment classification by providing initial sentiment scores before further analysis (Alfrjani et al., 2019) [1].

## TextBlob

TextBlob is another lexicon-based model that extends the capabilities of VADER by offering a broader range of text processing tools. It simplifies the task of sentiment analysis by providing an API that enables users to extract sentiment scores easily. TextBlob is particularly advantageous in applications requiring rapid sentiment analysis, as it efficiently handles basic sentiment classification tasks. In hybrid models, TextBlob can serve as a preprocessing step, providing foundational sentiment scores to enhance more sophisticated algorithms (Horvat et al., 2024) [8]

## BERT

BERT is a state-of-the-art transformer-based language model that captures the context of words by considering both left and right surrounding words in a sentence. This bidirectional approach allows BERT to understand the nuanced meaning of language, making it highly effective for tasks like sentiment analysis. In hybrid models, BERT is often combined with traditional machine learning classifiers or other deep learning architectures to improve sentiment classification accuracy, particularly in complex, domain-specific texts (Kaur et al., 2023) [12].

## RoBERTa

RoBERTa is an optimized variant of BERT that focuses solely on the pre-training phase, removing certain tasks like next sentence prediction and enhancing the model's performance on text classification tasks. It leverages a larger training dataset and longer training time, resulting in improved sentiment classification capabilities. In hybrid architectures, RoBERTa's robust language modeling is utilized alongside other models to achieve high accuracy in tasks like aspect-based sentiment analysis (Gupta, 2023) [7].

## CNN

CNNs are primarily known for their applications in image processing but have also proven effective in text analysis. In sentiment analysis, CNNs can capture local patterns in the text, such as phrases or n-grams, by applying convolutional filters. This capability allows CNNs to extract significant features from textual data, which can then be classified using various algorithms. Hybrid models combining CNNs with other techniques, such as LSTMs or SVMs, enhance sentiment classification by leveraging both feature extraction and classification strengths (Dang et al., 2021) [6]

## LSTM Networks

LSTMs are a type of recurrent neural network (RNN) designed to model sequential dependencies in data. They are particularly effective in capturing long-range dependencies in text, making them well-suited for sentiment analysis tasks involving longer documents or conversations. When integrated into hybrid models, LSTMs can work alongside CNNs or transformers to effectively analyze sentiment by considering both local and global context in the text (Rehman et al., 2019) [4]

## SVM

SVMs are traditional machine learning classifiers that excel in handling high-dimensional data and are particularly effective in binary classification tasks. In the context of sentiment analysis, SVMs can be used to classify extracted features from textual data. Hybrid models that combine SVMs with deep learning architectures, such as CNNs or transformers, leverage the feature extraction capabilities of the latter while maintaining the robustness of SVMs for classification, leading to improved sentiment classification outcomes (Elleuch et al., 2016) [3].

## TF-IDF

TF-IDF is a feature extraction method that transforms text into a numerical format, emphasizing important terms based on their frequency within a document relative to their occurrence across a corpus. This technique is often integrated into hybrid models to create feature vectors that can be analyzed by machine learning classifiers. By combining TF-IDF with advanced models like BERT or CNNs, hybrid sentiment analysis models can benefit from both shallow and deep textual features, significantly enhancing performance in applications like product reviews (GR Narayanaswamy et al., 2021) [9].

## Hybrid Models

Hybrid NLP models combine various machine learning and deep learning techniques to create more effective sentiment analysis systems. These models often integrate lexicon-based approaches with advanced algorithms, allowing them to handle complex, nuanced texts more effectively. For instance, a hybrid model might combine BERT with domain-specific knowledge to analyze public sentiment during crises, showcasing the flexibility and adaptability of hybrid frameworks in tackling various sentiment analysis challenges (Gupta, 2023) [7].

## Related Works

Several studies have explored hybrid models that integrate semantic knowledge bases with machine learning techniques. Alfrjani et al. (2019) [1] introduced a *Hybrid Semantic Knowledgebase-Machine Learning Approach*, which integrates structured semantic knowledge with machine learning algorithms. This combination improves opinion mining by leveraging external knowledge bases to handle ambiguous or domain-specific sentiments more effectively. By incorporating semantic understanding into machine learning, this model mitigates issues related to ambiguous language and improves classification accuracy.

Elshakankery (2019) [2] developed *HILATSA*, a hybrid incremental learning approach designed for analyzing Arabic tweets. This model adapts dynamically as it processes new data, making it ideal for real-time, large-scale sentiment analysis tasks. The hybrid structure combines incremental learning techniques with traditional sentiment analysis algorithms to enhance adaptability and accuracy.

Elleuch et al. (2016) [3] introduced a *CNN-SVM Hybrid Model* for offline Arabic handwritten recognition, which uses CNN for feature extraction and SVM for classification. While this model was primarily designed for text recognition, its architecture can be adapted to sentiment analysis by utilizing CNN for extracting features from text and SVM for sentiment classification.

Rehman (2019) [4] proposed a *Hybrid CNN-LSTM Model* for movie review sentiment analysis. In this model, CNN is used for spatial feature extraction, while LSTM captures sequential dependencies in the text. This approach effectively combines the strengths of both models, leading to improved accuracy in sentiment classification by leveraging both spatial and temporal features.

Shobayo et al. (2024) [5] conducted a comparative study focusing on customer sentiment analysis using Google PaLM, demonstrating how hybrid architectures that combine pretrained models like Google PaLM with traditional machine learning techniques result in superior performance for product review analysis.

Dang et al. (2021) [6] and Gupta et al. (2023) [7] explored hybrid models that combine CNN, LSTM, and transformers to handle sentiment analysis more effectively. These models leverage CNN’s feature extraction capabilities, LSTM’s strength in sequential data, and transformers’ contextual understanding to improve the classification of complex text datasets.

Horvat (2024) [8] introduced a *Hybrid Natural Language Processing (NLP) Model* for analyzing public sentiments during natural crises. This model integrates BERT with domain-specific knowledge bases to assess public sentiment more accurately. The hybrid model captures the complexities of crisis-related texts, showing significant improvements in context-specific sentiment analysis.

Narayanaswamy et al. (2021) [9] explored the combination of BERT and RoBERTa for aspect-based sentiment analysis (ABSA). The hybrid model leverages BERT’s contextual language understanding and RoBERTa’s robust fine-tuning capabilities to achieve high accuracy in identifying sentiments associated with specific aspects of products or services.

Semary et al. (2023) [10] emphasized the importance of feature extraction in hybrid models for sentiment analysis. By combining traditional feature extraction techniques such as TF-IDF with deep learning architectures like CNN and transformers, they demonstrated how hybrid models can achieve better performance on sentiment classification tasks.

Semary et al. (2024) [11] and Kaur et al. (2023) [12] applied hybrid deep learning models for consumer sentiment analysis, combining CNN, LSTM, and transformer models. These models efficiently capture sentiment variations in consumer product reviews by integrating advanced feature extraction techniques, improving accuracy and performance.

Kaur et al. (2023) [13] proposed a *RoBERTa-based hybrid model* for improving sentiment classification. This hybrid model combines RoBERTa’s deep contextual embeddings with traditional machine learning classifiers, leading to improved performance in sentiment analysis tasks, particularly when handling complex textual data.

Yürütücü et al. (2021) [14] introduced a hybrid model that integrates sentiment scores with readability metrics for filtering spam reviews. Their research highlighted how hybrid models combining sentiment analysis with linguistic features can outperform traditional spam filtering techniques, especially in the context of product reviews.

Kanmani (2023) [15] proposed a multi-channel *LSTM-CNN Hybrid Model* for Vietnamese sentiment analysis. The model uses CNN for feature extraction and LSTM for processing sequential data, resulting in more accurate sentiment classifications. The combination of spatial and sequential processing proved beneficial in handling complex Vietnamese texts.

Aggarwal, et al. (2023) [16] and QH Vo et al. (2027) [17] explored the use of pretrained language models like BERT, GPT, and RoBERTa for sentiment analysis. Their studies demonstrated that combining pretrained models with traditional feature extraction techniques leads to significant improvements in classification accuracy, particularly when handling sentiment nuances in large-scale datasets.

Keith et al. (2017) [18] presented a comprehensive analysis of mobile product reviews using hybrid sentiment analysis models. By integrating deep learning models with aspect-based analysis techniques, their approach yielded superior performance in sentiment classification and aspect identification.

**Chapter 3**

# Methods

This chapter gives the reader an overview of the different tools and procedures used to accomplish the Hybrid Model Implementation for Sentiment Analysis on Amazon Data research. The reader will also become familiar with each technology's basic overview and the design choices made to achieve it. In the first section, the programming environments will be discussed. The second section will cover about the data collection procedure. And then in the third section Data Cleaning Procedure will be explain. The fourth section will explain data splitting procedure. After that on fifth section model building and training procedure will be explained. The entire methodology used for this research will be explained in the final section.

Here is a small diagram for illustrated the overall Methodology steps we have done for the research:

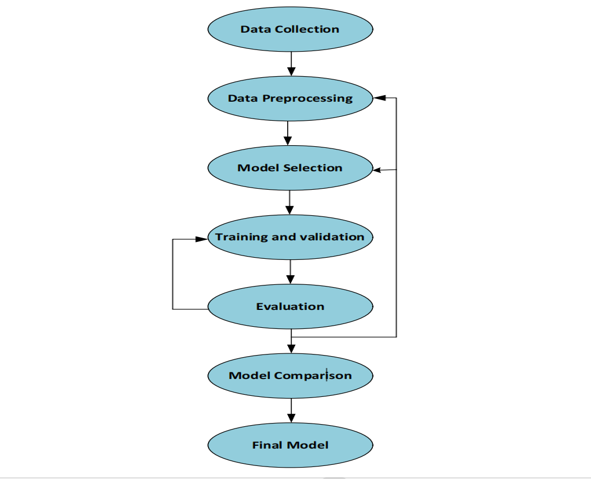


Figure-01: Methodology

## Programming Environment and Technologies

One of the most popular programming languages for data science and machine learning is Python. Numerous machine learning algorithms can be solved using the extensive library of Python. That’s why we have chosen python as the programming language for this research purpose. Python, libraries that we have used in this project are,

* + 1. Pandas,
    2. re (regular expression)
    3. NLTK (Natural Language Toolkit),
    4. BeautifulSoup
    5. Matplotlib,
    6. Seaborn,
    7. TextBlob
    8. TfidfVectorizer
    9. SVC
    10. numpy
    11. Transformers,
    12. Torch,
    13. TensorFlow

## Data Collection

Data was collected from amazon website by scrapping using Python code as Amazon does not have an API like Twitter to download reviews with. Later, the dataset was saved in the Comma Separated Values (CSV) format because Python can handle these kinds of files more easily. Datasets that we had collected is consist of Amazon Smartphone review Data. The fields listed below are included in the dataset:

* + 1. Id: A unique identifier
    2. **product\_title:** Name of the product is stored here
    3. **user\_name:** Name of the reviewer is stored in this field
    4. **rating:** rating of the product
    5. **review:** text of the review
    6. **review\_date:** date of the review

The Dataset contains a total of 4,574 reviews.

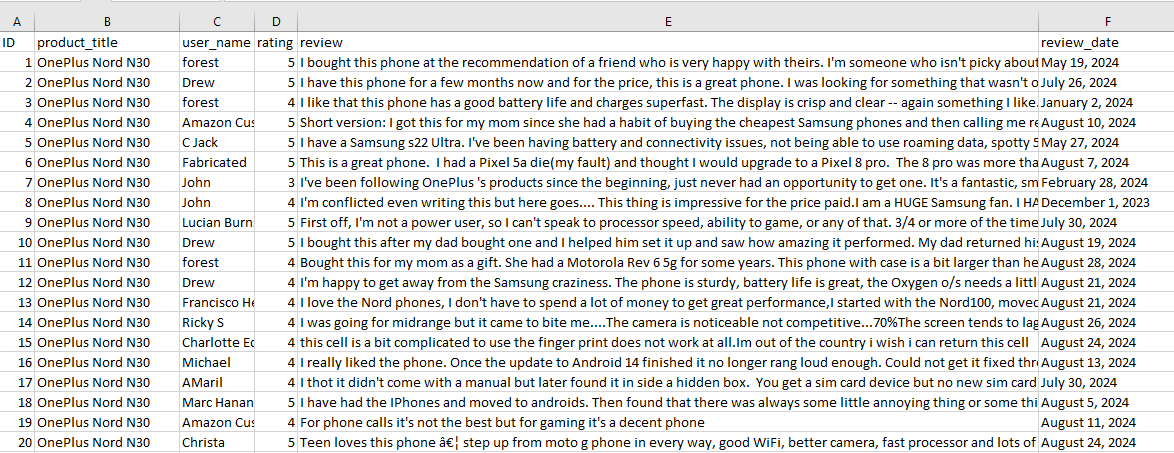


Figure-02: Sample Datasets

## Data Preprocessing

To guarantee that the text data is in a format that can be used to effectively train a neural network model, the data preprocessing steps in this model are essential. For various models, we have pre-processed the data in four different ways. The following is a description of the data pre-processing steps:

### 3.3.1 Import Datasets

First, we have imported the dataset by reading the csv file using pandas library where we have stored the data after collecting by Amazon web-scaping.

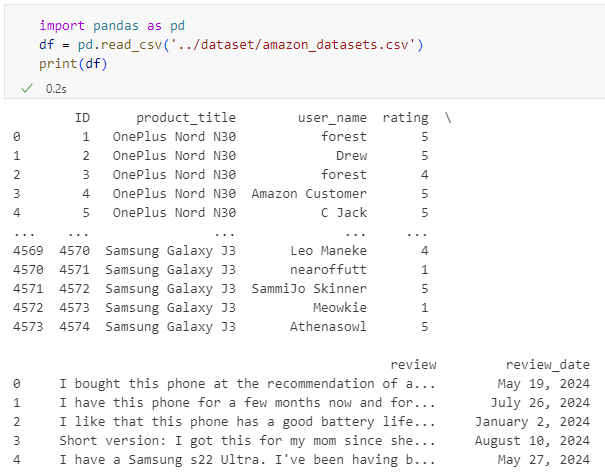


Figure-03: Import Datasets

### 3.3.2 Setup Preprocess

Then we have import re (regular expressions) and stopwords from nltk.corpus. Store the English stopword into a variable stop\_words. After that created a basic preprocess function where we have converted the text into lower case, remove the numbers and extra spaces from the text. Then for VADER we will apply this basic preprocessing directly. For transformers model we have remove the punctuation except the meaningful ones (, ! ?). After that for the Deep Learning models we have removed all the punctuations. And for machine learning model SVM we have removed the punctuations and stopwords. And then create the sentiment based on product rating.



Figure-04: Setup Preprocess

### 3.3.3 Apply Preprocess and Store Data

At last, we have assigned the value in different variables and store them into another CSV file named clean\_datasets.csv.



Figure-05: Apply Preprocess and Store Data

## Splitting Data

In machine learning projects, splitting data into separate subsets for training and testing is an essential preprocessing step. This procedure makes sure the model is tested on a different subset of data to assess its performance on data that hasn't been seen yet and trained on a portion of the data that has. The dataset is typically split into two subsets, testing and training, with ratios like 70-30 or 80-20. The model is trained using the training set, which has the majority of the data. The trained model's final performance is evaluated by running the test set.

By dividing the data, you can be sure that the model's performance is assessed on data that hasn't been seen before, giving you an objective gauge of how well it can generalize. We have split the data into a ratio of 80-20 ratio. 80% data is used for train the models and 20% data is used for testing.

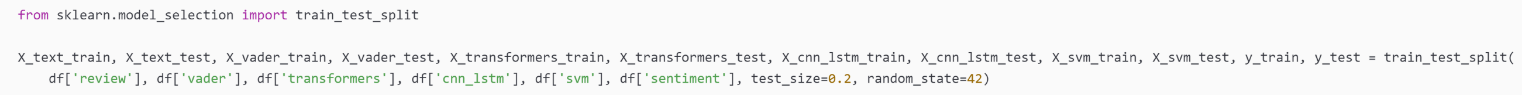


Figure-06: Splitting Data

## Model Building

After completing preprocessing of data and splitting the data we have built the models that we have used for implementing our hybrid model and train them for the final hybrid model implementation. We have used total 7 models for implementing the hybrid model that we have built. These models are VADER (Valence Aware Dictionary and sEntiment Reasoner), TextBlob, RoBERTa (Robustly Optimized BERT Pretraining Approach), BERT (Bidirectional Encoder Representations from Transformers), LSTM (Long Short-Term Memory), SVM (Support Vector Machine), CNN (Convolutional Neural Network). Here we will explain them one by one.

### 3.5.1 Transformer Models (RoBERTa, BERT)

Vaswani et al. (2017) introduced transformer models, a deep learning architecture mainly utilized for natural language processing (NLP) applications. Because the model can focus on multiple segments of the sequence at once, it is incredibly effective at managing long-range dependencies. This is because it processes input data using an attention mechanism. Regardless of distance, the model captures relationships between words in the sequence in order to comprehend the context of each individual word. The BERT and RoBERTa transformer models were utilized in this study.

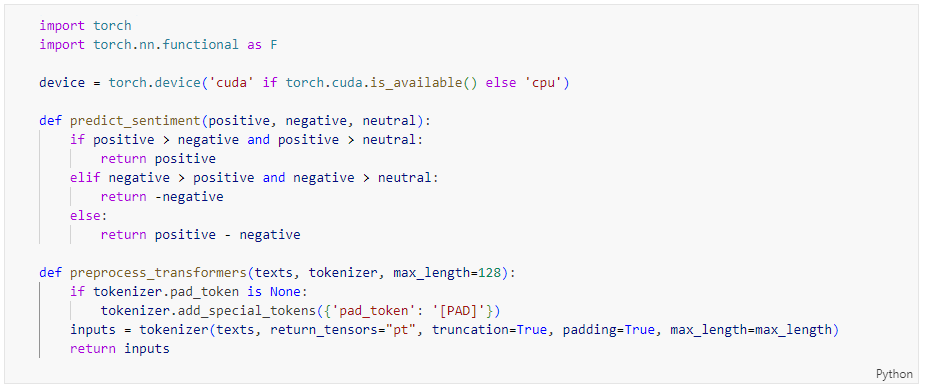


Figure-07: Preparing Transformer Models

First of all, we have initialized sentiment prediction function which will return the sentiment value on the basis of their probabilities. And also define a common function for pre-process the transformer models.

#### 3.5.1.1 BERT (Bidirectional Encoder Representations from Transformers)

In sentiment analysis we commonly use BERT where understanding the full context of a sentence is important for accurate predictions. BERT uses the encoder portion of the Transformer architecture as its foundation. BERT reads text bidirectionally, which means that it considers both the left and right sides of a word to understand its context, in contrast to traditional models that read text sequentially. Words in a sentence are randomly hidden, and the model is trained to predict the hidden word by analyzing the context. Next, by predicting whether two sentences will follow one another, the model is trained to comprehend sentence relationships. When it comes to natural language comprehension tasks like named entity recognition, classification, and question answering, BERT works incredibly well.

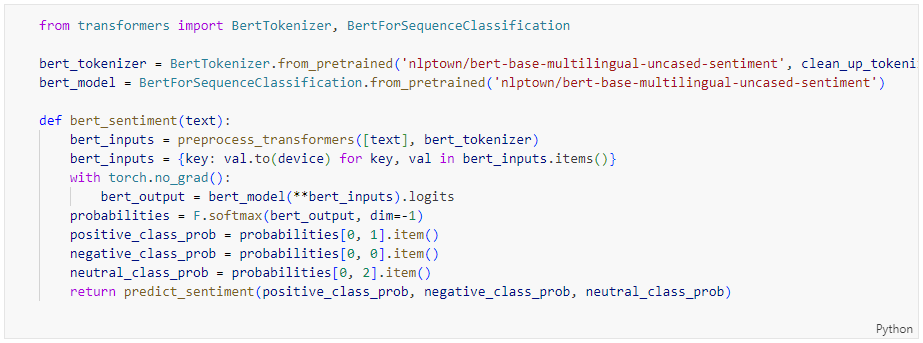


Figure-08: BERT Model Build

For Building BERT model, we have import BertTokenizer and BertForSequenceClassification from transformers libraries of python. Then we have created the tokenizer and build the BERT model and then we create the BERT sentiment prediction function. Where we have return the sentiment value directly on the basis of their probabilities.

#### 3.5.1.2 RoBERTa (Robustly Optimized BERT Pretraining Approach)

Basically, RoBERTa is a better BERT variant that has been optimized during training. While it employs the same Transformer-based encoder architecture as BERT, it has a few changes to enhance performance. The NSP task is eliminated by RoBERTa since tests revealed no discernible improvement in performance with it. Compared to BERT, RoBERTa has been trained on a far larger volume of data and for a longer period of time. In order to maximize the model's performance and make better use of the hardware, RoBERTa increases the batch size and learning rate. Unlike BERT, which uses static masking, where the masked tokens are selected once during preprocessing, RoBERTa modifies the masked tokens dynamically for each epoch. For NLP tasks like text classification, sentiment analysis, and question answering with improved performance, RoBERTa is frequently utilized.

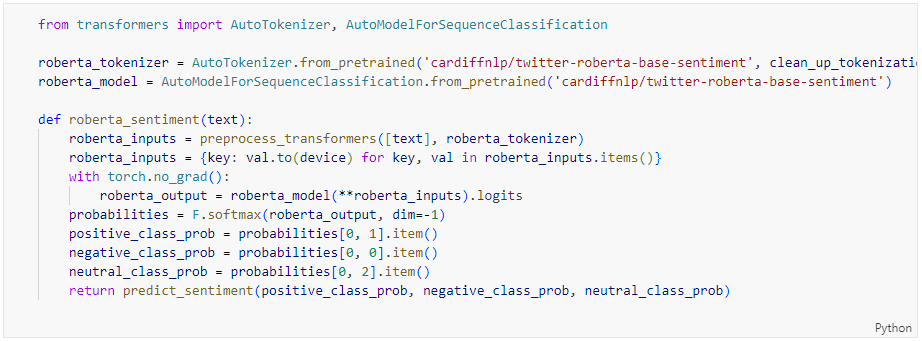


Figure-09: RoBERTa Model Build

For Building RoBERTa model, we have import AutoTokenizer and AutoModelForSequenceClassification from transformers libraries of python. Then we have created the tokenizer and build the RoBERTa model and then we create the RoBERTa sentiment prediction function. Where we have returned the sentiment value directly on the basis of their probabilities.

### 3.5.2 Lexicon Base Models (VADER, TextBlob)

Using lexicons or predefined word dictionaries, the lexicon-based approach to sentiment analysis is a rule-based methodology. In these lexicons, every word or phrase is assigned a sentiment score that categorizes it as either positive, negative, or neutral. The sentiment score of each word in a text can be added up or averaged to determine the text's overall sentiment. Unlike machine learning models, this method does not require a training phase. However, it might be limited by the predefined lexicon and struggle to understand sarcasm, context, or complicated sentences.

#### 3.5.2.1 VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a lexicon and rule-based sentiment analysis tool designed specifically for analyzing the sentiment of social media content, which tends to be short, informal, and contain emoticons, punctuation, and slang. VADER uses a dictionary of words where each word is assigned a valence score, which indicates its sentiment intensity. VADER outputs a compound score, a single value ranging from -1 (extremely negative) to +1 (extremely positive). This compound score is a normalized, weighted composite score of the sentiment across the text.



Figure-10: VADER Model Build

For building VADER model at first, we have import SentimentIntensityAnalyzer. Then, create Vader sentiment prediction function where we have simply returned the compound value of VADER polarity score.

#### 3.5.2.2 TextBlob

Another lexicon-based tool for Natural Language Processing (NLP) that offers a straightforward sentiment analysis API is TextBlob. Based on the words in the text, it assigns a sentiment polarity score that ranges from -1 (negative) to +1 (positive). TextBlob computes subjectivity in addition to polarity, indicating how subjective or opinionated a text is. Scores range from 0 (very objective) to 1 (very subjective). TextBlob doesn’t handle intensifiers (like "very" or "extremely") as VADER does, and it doesn’t account for emojis or social media slang. It works better with simpler sentiment analysis tasks or with texts that are more formal.

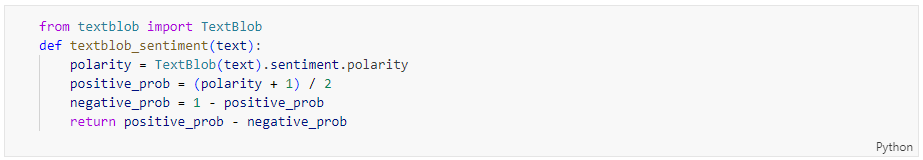


Figure-11: TextBlob Model Build

For building TextBlob model at first, we have import TextBlob library. Then, create TextBlob sentiment prediction function where we have simply returned the polarity score of TextBlob.

### 3.5.3 Machine Learning Model (SVM with TF-IDF vectorization)

A Machine Learning (ML) model refers to a mathematical model that is trained to recognize patterns in data and make predictions or decisions based on that data. Models are typically trained using datasets, which consist of input features and corresponding outputs or labels. Once trained, these models can be applied to new data to make predictions. ML models can be categorized into different types based on the type of problem they solve, such as classification, regression, clustering, etc. For our hybrid model we have used only one ML model that is SVM.

#### 3.5.3.1 TF-IDF

TF-IDF is a statistical measure used in text processing to evaluate the importance of a word in a document relative to a collection of documents. It is often used in Natural Language Processing (NLP) tasks, especially in text classification or sentiment analysis. TF-IDF is widely used for converting text data into numerical values for use in machine learning models. It helps identify words that are more representative of the content of a specific document while down-weighting common words that occur across many documents, like "the" or "is.". TF-IDF is composed of two components:

* **Term Frequency (TF)**: Measures how frequently a term (word) appears in a document relative to the total number of words in the document.
* **Inverse Document Frequency (IDF)**: Measures the importance of a term by considering how frequently it appears across all documents in the corpus. The idea is that terms appearing in many documents are less informative.

The TF-IDF score is calculated by multiply TF and IDF:

#### 3.5.3.2 SVM (Support Vector Machine)

The Support Vector Machine (SVM) is a classification method designed to identify the most effective hyperplane for separating distinct classes in each feature space. The primary objective is to optimize the margin, or absolute distance, between the hyperplane and the closest data points belonging to each class, referred to as support vectors.

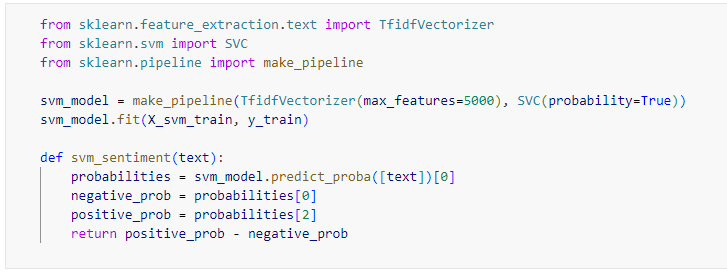


Figure-12: SVM Model Build with TF-IDF Vectorization

First of all, we have utilized a pipeline that combines two critical components: TF-IDF Vectorizer and Support Vector Machine (SVM). The aim is to process text data and classify it into predefined sentiment categories (e.g., negative, neutral, positive). For utilizing the pipeline, we have used TfidfVectorizer(max\_features=5000) where ‘max\_features=5000’ is used for limit the number of features (words) to the top 5,000 based on their TF-IDF scores. This reduces dimensionality and focuses on the most significant terms, enhancing the model's performance. Then we have initialized the SVM classifier using “SVC (probability=True)”, where the “probability=True” parameter allows the model to output class probabilities, which is useful for understanding the confidence of predictions.

After that we have defined the SVM sentiment prediction function where “svm\_model.predict\_proba([text])” returns an array of probabilities for each sentiment class (e.g., negative, neutral, positive). The function calculates the difference between the probability of the positive sentiment and the negative sentiment (positive\_prob - negative\_prob), providing a measure of sentiment strength. A positive value indicates a stronger positive sentiment, while a negative value suggests a stronger negative sentiment.

### 3.5.4 Deep Learning Models: (LSTM, CNN)

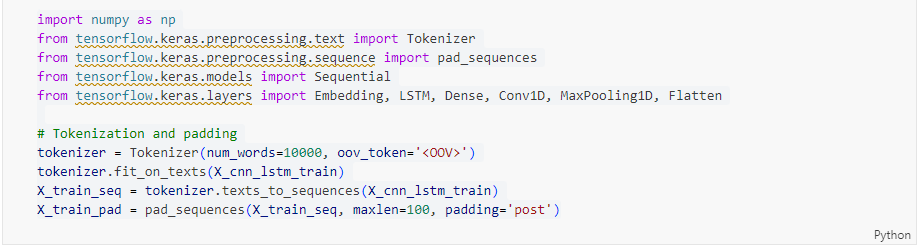
A class of machine learning techniques called deep learning models makes use of multi-layered neural networks to find patterns and representations in data. Here, sentiment analysis is used to categorize the sentiment of text data using CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory).

Figure-13: Prepare Deep Learning Model

For Building deep learning models at first, we have done the tokenization and padding.

#### 3.5.4.1 LSTM (Long Short-Term Memory)

Recurrent neural networks (RNNs) of the long-term dependency model (LSTM) are made to handle sequential data by identifying long-term dependencies in the input sequence. It is especially helpful for text or time-series data where the current state is influenced by previous elements in the sequence.



Figure-14: LSTM Model Build

For building the LSTM model we have applied an LSTM layer with 128 units to capture sequential dependencies after using an embedding layer to learn word representations. Finally, a dense layer with a sigmoid activation function outputs a probability between 0 and 1, representing the positive sentiment. For tasks where comprehending word order is crucial, this model is perfect. It predicts sentiment in this situation by identifying patterns in text sequences.

#### 3.5.4.2 CNN (Convolutional Neural Network)

Although CNNs are mainly used for image data, they can also be used to classify text by treating the text as a one-dimensional sequence. CNNs extract features from specific areas of the input data by applying convolutional filters.



Figure-15: CNN Model Build

We have created dense word vectors using an embedding layer in order to construct the CNN model. In order to identify local patterns (such as phrases) in the sequence, a 1D convolutional layer with 128 filters and a kernel size of 5 slides over the text. Max pooling is used to minimize the size of the feature maps following convolution. Using the sigmoid activation function, the final dense layer outputs a probability for positive sentiment.

## 3.5.5 Hybrid Model

After successfully building all the models. We have built the proposed hybrid model of this research. This hybrid sentiment analysis model combines the strengths of multiple models to generate a comprehensive sentiment prediction. Each individual model processes text data differently, and the final prediction is based on the average score from all models. The hybrid approach improves robustness and generalization by leveraging various algorithms and techniques for sentiment prediction.

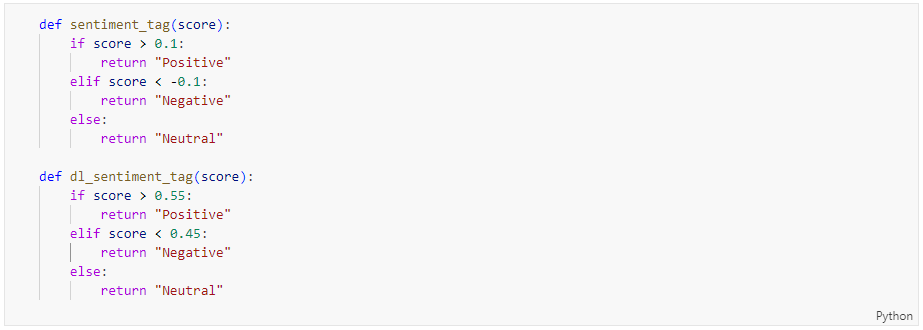


Figure-16: Hybrid Model Setup

We have declared the sentiment tagging for the models first. Scores above a positive threshold (> 0.1 for VADER & TextBlob & Transformers & SVM and > 0.55 for CNN & LSTM) are tagged as "Positive". Scores below a negative threshold (< -0.1 for VADER & TextBlob & Transformers & SVM and < 0.45 for CNN & LSTM) are tagged as "Negative". Scores in between are considered as "Neutral".

Once the sentiment tagging setup is complete, we can begin constructing the hybrid model. Five distinct preprocessed version of input text for the various models are accepted by the hybrid\_model function. Where TextBlob uses text, the Vader model uses vader\_text, transformer models RoBERTa and BERT uses transformer\_text, CNN and LSTM use cnn\_lstm\_test and the SVM model uses svm\_text.

Each model is called sequentially and their sentiment scores are calculated and store into different variables. And the hybrid model score is the mean of all the models score. This averaging provides a balanced sentiment prediction by integrating various perspectives from rule-based, machine learning, and deep learning models. The results from all models, including individual sentiment scores and sentiment tags, are stored in a dictionary (comparison), which is returned for further analysis. This is how we have build our hybrid model.



Figure-17: Hybrid Model Build

* 1. **Implementing Hybrid Model and Store the Result**

After building the hybrid model we have implemented it. The variables texts, vader\_texts, transformers\_texts, cnn\_lstm\_texts, and svm\_texts represent the preprocessed test data for different models (VADER, transformers, CNN/LSTM, and SVM). A loop is used to iterate over the dataset. For each review (text), the hybrid\_model function is called. This function takes in the review and its various representations for different models. The function returns a comparison dictionary that contains sentiment scores and tags from multiple models (VADER, BERT, RoBERTa, CNN, LSTM, SVM) as well as the hybrid sentiment. Each comparison result is appended to a list (results\_list), storing the sentiment analysis results for all reviews. The list of results is converted into a DataFrame. The DataFrame is saved as a CSV file (sentiment\_results.csv) without including the index.

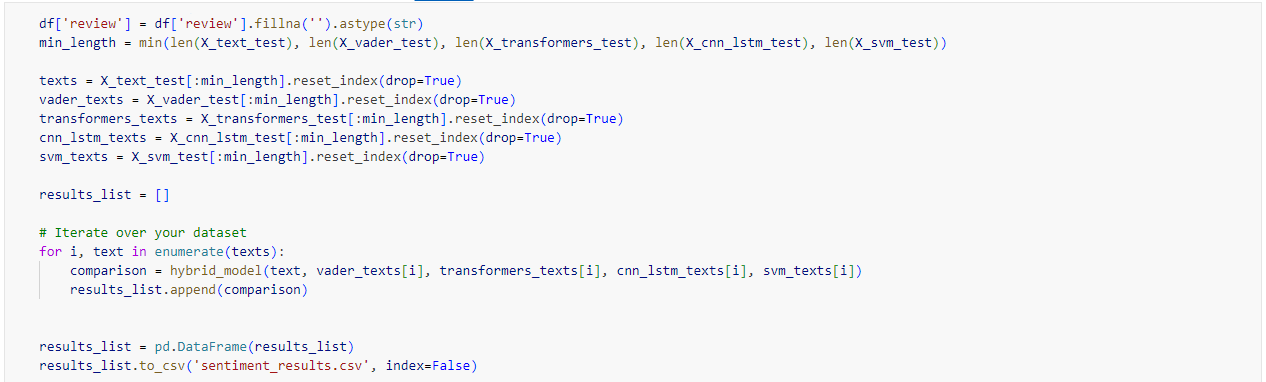


Figure-18: Final Model Implementation

**Chapter 4**

# Results or findings

## Result Analysis

We have analysis our result on 2 different ways. First, we have compared the sentiment scores for each model. Where our hybrid model performs better than the other models in some points. By analyzing sentiment scores, we have found out the in terms of short text our model is perform better than the other models except Vader. But in terms of long text our models perform better.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision (Macro) | Precision (Micro) | Precision (Weighted) | Recall | F1-Measure |
| **VADER** | 0.67 | 0.3319 | 0.67 | 0.6334 | 0.3949 | 0.3541 |
| **RoBERTa** | 0.73 | 0.3663 | 0.73 | 0.6655 | 0.4452 | 0.3973 |
| **BERT** | 0.67 | 0.3205 | 0.67 | 0.6307 | 0.3716 | 0.3397 |
| **TextBlob** | 0.70 | 0.3294 | 0.70 | 0.6073 | 0.3851 | 0.3550 |
| **SVM** | 0.76 | 0.3544 | 0.76 | 0.6539 | 0.4122 | 0.3811 |
| **CNN** | 0.74 | 0.2467 | 0.74 | 0.5476 | 0.3333 | 0.2835 |
| **LSTM** | 0.74 | 0.2492 | 0.74 | 0.5531 | 0.3333 | 0.2852 |
| **Hybrid** | 0.77 | 0.3741 | 0.77 | 0.6444 | 0.4167 | 0.3917 |

### 4.1.1 Accuracy

Accuracy reflects the overall correctness of the model means it calculates how often the model's predictions are correct. In our research Hybrid model achieves the highest accuracy 77%, followed by SVM 76% and RoBERTa 73%. VADER 67%, BERT 67%, and TextBlob 70% perform moderately. CNN 74% and LSTM 74% are quite close in performance.

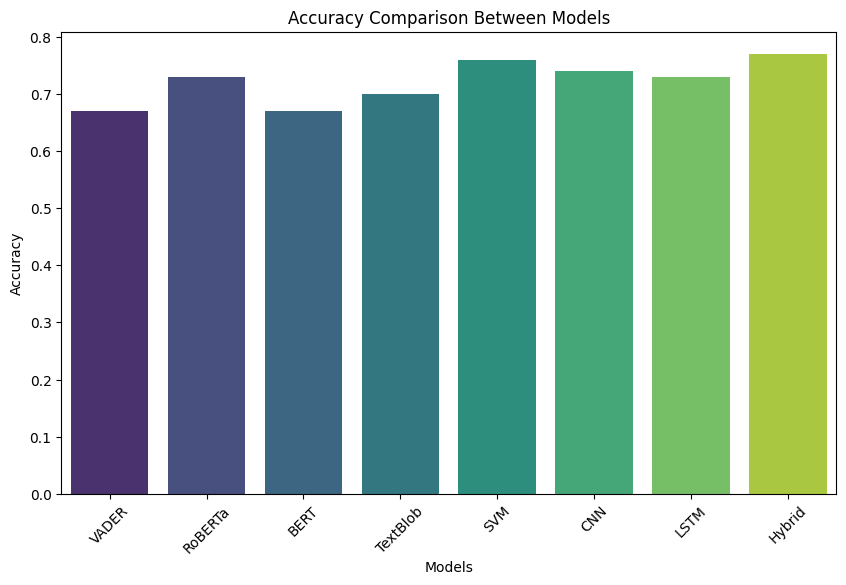


Figure-19: Accuracy Graph

### 4.1.2 Precision (Macro)

The average precision for all classes is called precision (macro). All classes are treated equally, regardless of size. The highest macro precision is found in hybrid (0.3741), which is followed by RoBERTa (0.3663) and SVM (0.3544). Similar lower precision is shown by BERT (0.3205), TextBlob (0.3294), and VADER (0.3319). The models with the lowest precision (macro) are CNN (0.2467) and LSTM (0.2492), suggesting that they may not be able to effectively handle imbalanced classes.

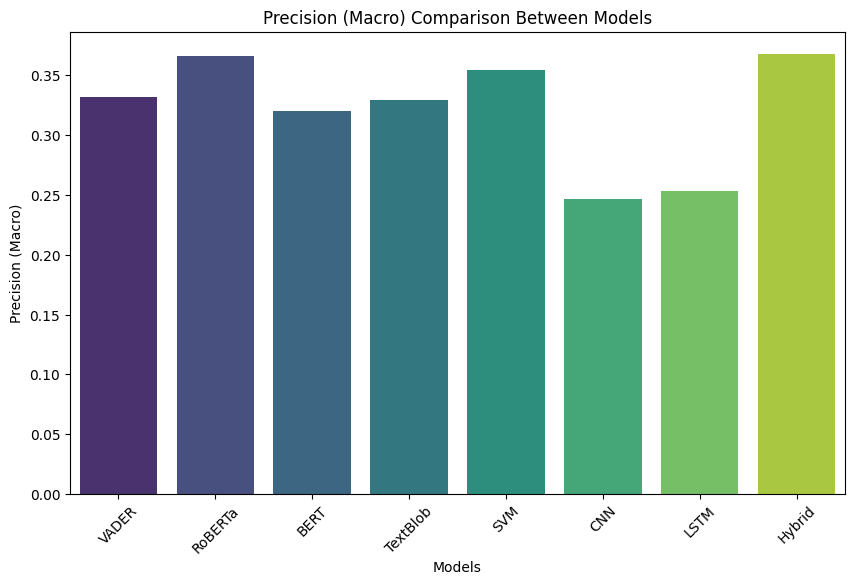


Figure-20: Precision (macro) Graph

### 4.1.3 Precision (Micro)

Precision (Micro) counts the total number of true positives and false positives to calculate the precision globally. This metric is affected by the class that occurs most frequently. The models that perform best in precision (micro) are hybrid 77%, SVM 76%, and RoBERTa 73%, indicating that they effectively handle the majority class. Additionally competitive are LSTM 74% and CNN 74%. In comparison to the other models, BERT 67% and VADER 67% have somewhat lower values, suggesting less ability to deal with the majority class.

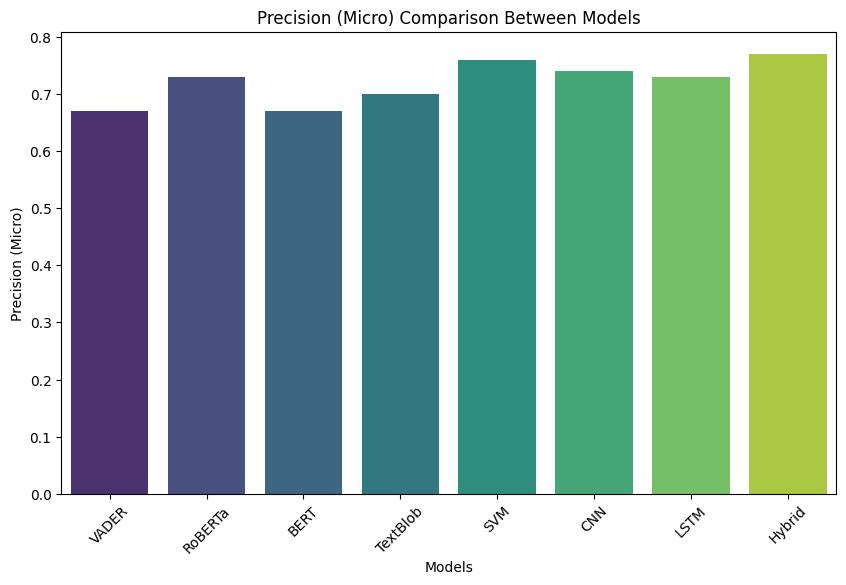


Figure-21: Precision (micro) Graph

### 4.1.4 Precision (weighted)

Precision (Weighted) determines each label's precision and weights it according to the number of actual instances of each class. When it comes to addressing class disparities, this measure is helpful. Here, Hybrid model 64% and RoBERTa 67% perform best, demonstrating their ability to manage class imbalance and produce dependable predictions for both majority and minority classes.  
Results from VADER 64% and SVM 65% are reasonable. CNN 55% and LSTM 55% underperform compared to BERT 63% and TextBlob 60%.

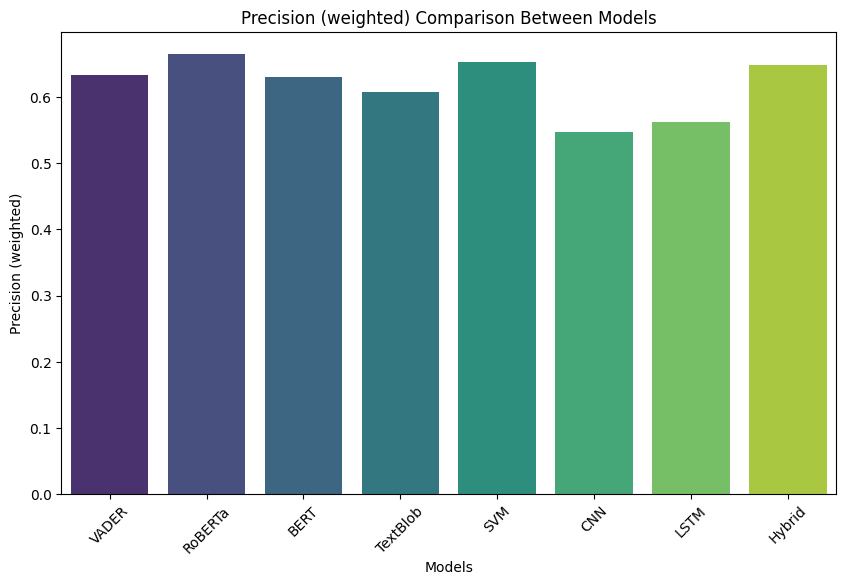


Figure-22: Precision (weighted) Graph

### 4.1.5 Recall

The number of correctly identified actual positives is measured by recall. Greater recall denotes more comprehensive coverage of the favorable cases. Recall is comparatively good for hybrid (0.4167), SVM (0.4122), and RoBERTa (0.4452), which means they can detect more true positives. Recall is a little lower for TextBlob (0.3851) and VADER (0.3949). BERT is even lower (0.3716). The lowest recall, shared by both CNN and LSTM (0.3333), suggests that they miss a sizable portion of true positives.

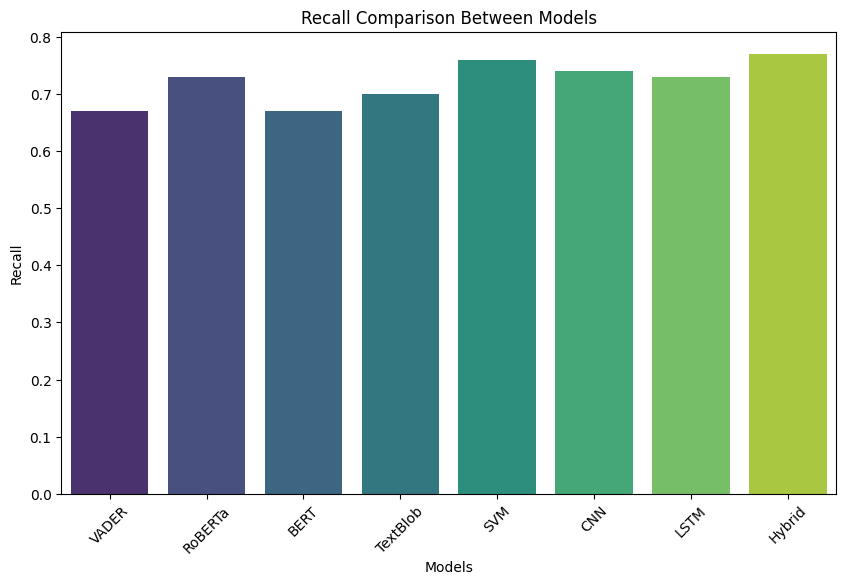


Figure-23: Recall Graph

### 4.1.6 F1-Score

The harmonic mean of Precision and Recall is known as the F1-Measure. It balances the two metrics and is useful when precision and recall are at odds. RoBERTa (0.3973) and Hybrid (0.3917) exhibit the best balance between precision and recall. Next, and with a good overall performance, is SVM (0.3811). Although less, VADER (0.3541) and TextBlob (0.3550) are still largely in balance. BERT (0.3397) is not as fast as the rest. The two models with the lowest F1 scores—CNN (0.2835) and LSTM (0.2852)—indicate a poor trade-off between precision and recall.

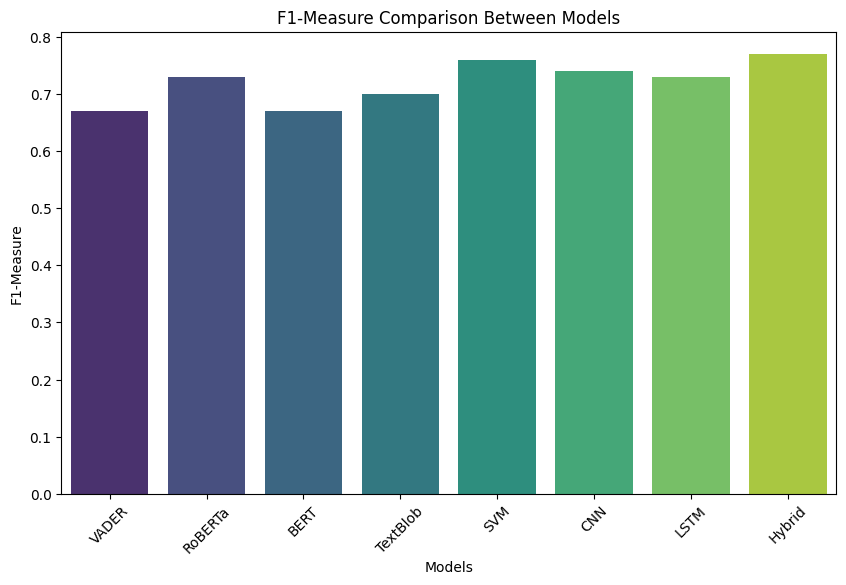


Figure-24: Precision (macro) Graph

## 4.2 Comparison of Results:

According to this study, the hybrid model outperformed the others in terms of accuracy (0.77), precision, recall, and F1 score. Additionally demonstrating strong performance, SVM and RoBERTa are dependable models for sentiment analysis. While CNN and LSTM underperformed, particularly in recall and F1 scores, indicating they missed more true positives, VADER, BERT, and TextBlob performed moderately. In general, the best methods for attaining balanced precision and recall are hybrid models (combining multiple models) or SVM and RoBERTa.

**Chapter 5**

# Discussion

## Introduction

This section should analyze the significance and implications of the results:

Overall, the best models are Hybrid, SVM and RoBERTa, with the hybrid model consistently receiving the highest score across the board. This implies that integrating various models can produce a strong classifier that capitalizes on the advantages of several strategies. RoBERTa consistently perform well in recall and F1-scoring, demonstrating its ability to distinguished true positives and strike a balance and imbalance classes. The lower performance of VADER, BERT and TextBlob across most metrics indicates that these models might not be reliable for this particular task. Specifically, in F1, recall, and precision (macro), CNN and LSTM perform worse than expected. They may be missing a lot of true positives, as suggested by their low recall and F1, which suggests they might have trouble with complex sentiment analysis in this instance.

So, it shows that ensemble approaches can be effective for sentiment analysis, as hybrid models seem to provide the best trade-off between the metrics. SVM and RoBERTa are dependable, comprehensive models, particularly for jobs requiring a high degree of precision and recall. For VADER and TextBlob to perform better, they may require additional fine-tuning or a hybrid strategy. For the CNN and LSTM models to perform better, especially in terms of recall and F1 scores, additional training data, altered hyperparameters, or improved feature engineering may be needed.

## 5.2 Limitations

**Complexity:** Developing, training, and deploying hybrid models can be challenging, particularly when integrating several models with various architectures and parameter sets.

**Computational Cost:** Hybrid model training and deployment can be computationally costly, particularly for complicated and large-scale models.

**Interpretability:** It can be difficult to comprehend how each model contributes to the hybrid system, which makes it difficult to interpret the model's predictions.

**Data Quality:** The quality of the training data can significantly impact the performance of the hybrid model. Predictions may be erroneous or biased as a result of noisy or biased data.

**Chapter 6**

# Conclusion

## 6.1 Future Work

* **Data Augmentation:** Examine methods for adding to the training set, such as random deletions, back translation, and synonym replacement, in order to enhance the generalization of the model.
* **Hybrid Architecture Optimization:** Examine hybrid architectures that are more scalable and efficient, possibly by fusing various models or by applying transfer learning strategies.
* **Interpretability Techniques:** Provide techniques to enhance hybrid models' interpretability so that it is simpler to comprehend how the models arrive at their predictions.
* **Domain Adaptation:** Examine methods for minimizing the need for significant retraining by adapting hybrid models to new languages or domains.
* **Real-time Applications:** Examine how hybrid models can be implemented in real-time applications like chatbots for customer service or social media monitoring.
* **Evaluation Metrics:** Provide more thorough evaluation metrics that go beyond accuracy and F1-score to account for the subtleties of sentiment analysis tasks.

## 6.2 Conclusion

This thesis investigated hybrid models combining semantic knowledge bases, machine learning, and deep learning techniques to enhance sentiment analysis accuracy.

The study demonstrated that hybrid approaches, such as CNN-LSTM, BERT, and RoBERTa-based models, significantly outperformed traditional methods by effectively capturing complex sentiment patterns, especially in domain-specific tasks like customer reviews.

Future research should focus on optimizing these models for broader applicability, efficiency, and real-time sentiment analysis across different languages and contexts.

# Bibliography

1. R Alfrjani, T Osman, G Cosma, “A Hybrid Semantic Knowledgebase-Machine Learning Approach for Opinion Mining” - Data & Knowledge Engineering, Elsevier, Volume 121, May 2019, Pages 88-108, doi: <https://doi.org/10.1016/j.datak.2019.05.002>
2. K Elshakankery, MF Ahmed, “A hybrid Incremental learning approach for Arabic tweets sentiment analysis” - Egyptian Informatics Journal, Elsevier, Volume 20, Issue 3, November 2019, Pages 163-171, doi: <https://doi.org/10.1016/j.eij.2019.03.002>
3. M Elleuch, R Maalej, M Kherallah, “A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition”- Procedia Computer Science, Elsevier, Volume 80, 2016, Pages 1712-1723, doi: [https://doi.org/10.1016/j.procs.2016.05.512](https://doi.org/10.1016/j.procs.2016.05.512%20)
4. AU Rehman, AK Malik, B Raza, W Ali, “A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis” - Multimedia Tools and Applications, pringer, Volume 78, page 26597-26613, (2019), doi: <https://doi.org/10.1007/s11042-019-07788-7>
5. O Shobayo, S Sasikumar, S Makkar, O Okoyeigbo, “Customer Sentiments in Product Reviews: A Comparative Study with GooglePaLM” - Analytics, dpi.com,Volume 2024,3(2),241-254, doi: <https://doi.org/10.3390/analytics3020014>
6. CN Dang, MN Moreno-García, F De la Prieta, “Hybrid Deep Learning Models for Sentiment Analysis” - Complexity, Wiley Online Library, Volume 2021, Issue 1/9986920, doi: <https://doi.org/10.1155/2021/9986920>
7. A Gupta, I Arhatia, D Gautam, R Verma, “Mobile Review Sentiments: An In-Depth Analysis” - researchgate.net
8. M Horvat, G Gledec, F Leontić, “Hybrid Natural Language Processing Model for Sentiment Analysis during Natural Crisis” - Electronics, mdpi.com, Volume Electronics 2024, 13(10), 1991 , doi: <https://doi.org/10.3390/electronics13101991>
9. GR Narayanaswamy, “Exploiting BERT and RoBERTa to Improve Performance for Aspect Based Sentiment Analysis” - arrow.tudublin.ie,Volume Narayanaswamy, G.R. (2021), doi: <https://doi.org/10.21427/3w9n-we77>
10. NA Semary, W Ahmed, K Amin, P Pławiak, “Improving sentiment classification using a RoBERTa-based hybrid model” - Frontiers in Human,frontiersin.org, Volume 17 - 2023, doi: <https://doi.org/10.3389/fnhum.2023.1292010>
11. N A. Semary, W Ahmed, K Amin, P Pławiak, Mohamed Hammad “Enhancing machine learning-based sentiment analysis through feature extraction techniques” - Plos one, journals.plos.org, Volume February 14, 2024, doi: <https://doi.org/10.1371/journal.pone.0294968>
12. G Kaur, A Sharma “HAS: Hybrid Analysis of Sentiments for the perspective of customer review summarization”-Journal of Ambient Intelligence and Humanized, Springer, Volume 14, pages 11971–11984, (2023), doi: <https://doi.org/10.1007/s12652-022-03748-6>
13. G Kaur, A Sharma “A deep learning‑based model using hybrid feature extraction approach for consumer sentiment analysis”-Journal of big data, Springe,Volume 10, article number 5, (2023), doi: <https://doi.org/10.1186/s40537-022-00680-6>
14. ÖY Yürütücü, “THE USE OF PRETRAINED LANGUAGE MODELS IN SENTIMENT ANALYSIS” - search.proquest.com, APRIL 2022,69 Pages, doi:
15. S Kanmani, S Balasubramanian, “Leveraging Readability and Sentiment in Spam Review Filtering Using Transformer Models”- Computer Systems Science & Engineering,cdn.techscience.cn, Vol.45, No.2, 2023 doi: <https://doi.org/10.32604/csse.2023.029953>
16. R Aggarwal, L Gupta, “A Hybrid Approach for Sentiment Analysis using Classification Algorithm”- International Journal of Computer Science and Mobile
17. QH Vo, HT Nguyen, B Le “Multi-channel LSTM-CNN model for Vietnamese sentiment analysis”- 9th international, ieeexplore.ieee.org, Volume 19-21 October 2017, doi: <https://doi.org/10.1109/KSE.2017.8119429>
18. B Keith, E Fuentes, C Meneses, “A Hybrid Approach for Sentiment Analysis using Classification Algorithm” - Proceedings of ACM SIGKDD, blog.sentic.net, Volume August 2017 (KDD’17), 10 pages,

*End quote goes here.*