Sentiment Analysis of Electric Vehicle Discourse in Malaysia Using BERT-Based Language Model

CHANG ZI YIN

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA
DECLARATION OF Choose an item.

Author's full name	:		
Student's Matric No.	:	Academic Session	:
Date of Birth	:	UTM Email	:
Choose an item. Title	:	TITLE IN CAPITAL LETTERS TITLE IN CAPITAL LETTERS TITLE IN CAPITAL LETTERS	

I declare that this Choose an item. is classified as:

☒ **OPEN ACCESS** — I agree that my report to be published as a hard copy or made available through online open access.

☐ **RESTRICTED** — Contains restricted information as specified by the organization/institution where research was done. *(The library will block access for up to three (3) years)*

☐ **CONFIDENTIAL** — Contains confidential information as specified in the Official Secret Act 1972)

(If none of the options are selected, the first option will be chosen by default)

I acknowledged the intellectual property in the Choose an item. belongs to Universiti Teknologi Malaysia, and I agree to allow this to be placed in the library under the following terms :

1. This is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library of Universiti Teknologi Malaysia is allowed to make copies of this Choose an item. for academic exchange.

Signature of Student:

Signature :

Full Name
Date :

Approved by Supervisor(s)

Signature of Supervisor I:	Signature of Supervisor II
Full Name of Supervisor I NOOR HAZARINA HASHIM	Full Name of Supervisor II MOHD ZULI JAAFAR
Date :	Date :

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

Note: This page is only applicable for theses classified as Restricted/Confidential. Please delete this page from the template if your thesis is not classified as Restricted/Confidential. This letter should be written by a supervisor and addressed to Perpustakaan UTM. A copy of this letter should be attached to the thesis.

Date:

Librarian

Jabatan Perpustakaan UTM,

Universiti Teknologi Malaysia,

Johor Bahru, Johor

Sir,

CLASSIFICATION OF THESIS AS RESTRICTED/CONFIDENTIAL

TITLE: Click or tap here to enter text.

AUTHOR'S FULL NAME: Click or tap here to enter text.

Please be informed that the above-mentioned thesis titled _____ should be classified as RESTRICTED/CONFIDENTIAL for a period of three (3) years from the date of this letter. The reasons for this classification are

(i)

(ii)

(iii)

Thank you.

Yours sincerely,

SIGNATURE:

NAME:

ADDRESS OF SUPERVISOR:

"Choose an item. hereby declare that Choose an item. have read this Choose an item. and in Choose an item. opinion this Choose an item. is sufficient in term of scope and quality for the award of the degree of Choose an item."

Signature	:	_____
Name of Supervisor I	:	KHAIRUR RIJAL JAMALUDIN
Date	:	9 MAY 2017

Signature	:	_____
Name of Supervisor II	:	NOOR HAZARINA HASHIM
Date	:	9 MAY 2017

Signature	:	_____
Name of Supervisor III	:	MOHD ZULI JAAFAR
Date	:	9 MAY 2017

Declaration of Cooperation

This is to confirm that this research has been conducted through a collaboration Click or tap here to enter text. and Click or tap here to enter text.

Certified by:

Signature :
Name :
Position :
Official Stamp
Date

* This section is to be filled up for theses with industrial collaboration

Pengesahan Peperiksaan

Tesis ini telah diperiksa dan diakui oleh:

Nama dan Alamat Pemeriksa Luar **:**

Nama dan Alamat Pemeriksa Dalam **:**

Nama Penyelia Lain (jika ada) **:**

Disahkan oleh Timbalan Pendaftar di Fakulti:

Tandatangan :

Nama :

Tarikh :

ON-LINE RECOGNITION OF DEVELOPING CONTROL CHART PATTERNS

TITLE

TITLE

TITLE

WAN ZUKI AZMAN WAN MUHAMAD

A Choose an item. submitted in Choose an item. of the
requirements for the award of the degree of
Choose an item.

School of Education
Faculty of Social Sciences and Humanities
Universiti Teknologi Malaysia

OCTOBER 2024

DECLARATION

I declare that this Choose an item. entitled *"title of the thesis"* is the result of my own research except as cited in the references. The Choose an item. has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	.....
Name	:	
Date	:	10 NOVEMBER 2016

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Professor Dr. Mohd Shariff Nabi Baksh, for encouragement, guidance, critics and friendship. I am also very thankful to my co-supervisor Professor Dr Awaluddin Mohd Shaharoun and Associate Professor Dr. Hishamuddin Jamaluddin for their guidance, advices and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Ph.D study. Librarians at UTM, Cardiff University of Wales and the National University of Singapore also deserve special thanks for their assistance in supplying the relevant literatures.

My fellow postgraduate student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family member.

ABSTRACT

The purpose of this study is to investigate the sentiment analsysis of electric vehicle discourse in Malaysia. The research was based on 3 primary objective. (a) to identify, preprocess, and explore EV-related textual data from online social platforms by implementing robust data cleaning and pattern discovery techniques; (b) to implement and compare the performance of pre-trained BERT-based models, specifically BERT-base uncased and RoBERTa-base, for structured sentiment analysis; and (c) to analyze the sentiment insights of Malaysians toward electric vehicles. Using VADER for initial sentiment labeling, both models were fine-tuned and evaluated. BERT-base uncased achieved a higher classification accuracy (93%) compared to RoBERTa-base (92%), particularly excelling in positive sentiment detection. The findings highlight the effectiveness of leveraging pre-trained language models for analyzing public opinion in EV-related discourse and suggest avenues for improvement through enhanced labeling strategies and experimentation with other transformer architectures.

ABSTRAK

Kajian ini dilakukan bertujuan mengkaji penggunaan algoritma genetik (GA) dalam pemodelan sistem dinamik linear dan tak linear dan membangunkan kaedah alternatif bagi pcmilihan struktur model menggunakan GA. Algorithma kuasa dua terkecil ortogon (OLS), satu kaedah penurunan kecerunan digunakan sebagai bandingan bagi kaedah yang dicadangkan. Pcmilihan struktur model mengunakan kaedah algoritma genetik yang diubahsuai (MGA) dicadangkan dalam kajian ini bagi mengurangkan masalah konvergens pramatang dalam algoritma genetik mudah (SGA). Kesan penggunaan gabungan operator MGA yang berbeza ke atas prestasi model yang terbentuk dikaji dan keberkesanan serta kekurangan MGA diu t arakan. Kajian simulasi dilakukan untuk membanding SGA, MGA dan OLS. Dengan meggunakan bilangan parametcr dinamik yang setara kajian ini mendapati, dalam kebanyakan kes, prestasi MGA adalah lebih baik daripada SGA dalam mencari penyelesaian yang berpotensi dan lebih berkebolehan daripada OLS dalam menentukan bilangan sebutan yang dipilih dan ketcpatan ramalan. Di samping itu, penggunaan carian tcmpatan dalam MGA untuk menambah baik algorithma tersebut dicadang dan dikaji, dinamai sebagai algoritma mcmetic (MA). Hasil simulasi menunjukkan, dalam kebanyakan kes, MA berkeupayaan menghasilkan model yang bersesuaian dan parsimoni dan mcmenuhi ujian pengsahihan model di samping mcmperolehi beberapa kelebihan dibandingkan dengan kaedah OLS, SGA dan MGA. Tambahan pula, kajian kes untuk sistcm berbilang pcmbolehubah menggunakan data eksperimental sebenar daripada dua sistem iaitu sistem pengulang-alik turbo dan reaktor teraduk berterusan menunjukkan algoritma ini boleh digunakan sebagai alternatif untuk mcmperolehi model termudah yang memadai bagi sistcm tersebut.

(Note: Students are allowed to use either single or one-and-a-half spacing for the abstract, as long as it fits within one page. The chosen spacing style must be consistent across both the English and Malay sections.)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
GA	-	Genetic Algorithm
PSO	-	Particle Swarm Optimization
MTS	-	Mahalanobis Taguchi System
MD	-	Mahalanobis Distance
TM	-	Taguchi Method
UTM	-	Universiti Teknologi Malaysia
XML	-	Extensible Markup Language
ANN	-	Artificial Neural Network
GA	-	Genetic Algorithm
PSO	-	Particle Swarm Optimization

LIST OF SYMBOLS

δ	-	Minimal error
D, d	-	Diameter
F	-	Force
v	-	Velocity
p	-	Pressure
I	-	Moment of Inersia
r	-	Radius
Re	-	Reynold Number

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this age of information, web user is able to access vast amount of data through the advancement of web technology. Users use the web's available resources, and participate in providing comments, which resulting in the generation of additional data. Most of the text information available in the web such as social media is unstructured data. Even thought those review, post and comment are unstructured data but they contain massive amount of information that could provide valuable insight. Especially for marketing business analysis and future improvement guideline for an organization. Hence, organising, analysing, and exploration of web user feedback and idea in an efficient manner is important and crucial to provide an effective decision-making. However, text mining and sentiment Analysis (SA) on web text using natural language processing (NLP) techniques is required (Ferdous, Syed, Bin, & Uddin, 2024).

Bidirectional Encoder representations from Transformer (BERT) is one of the most well-known deep learning models for natural language processing (Gardazi et al., 2025). It is a pre-trained model based on transformer architecture. During pre-trained, BERT's mask language modelling is trained to enable it predict masked word based on surrounding token content. Hence, BERT can learn and understand the contextual information and relationship between token of word causing it effective for language processing task. There is a research show that BERT is applicable in mining nuance information of the language which is beneficial for sentiment analysis by capturing the content of word (Sornalakshmi et al., 2024). Sentiment analysis is typically classifying text based on sentiment expression of positive, negative and neutral. Hence, BERT is suitable to be implement in mining contextual information from a social media text. Research by Bhola et.al. (Bhola et al., 2022). conducted

research on text mining utilizing BERT for sentiment analysis. Sentiment analysis for monitoring customer feedback on social media to improve service and product.

A study analysing on 36000 Facebook posts, identified that there is cluster of discussion on topic related to economic, politic and legal aspect of electric vehicle (Debnath, Ronita Bardhan, Reiner, & Miller, 2021). There another study on cross-multiple social media platforms of Facebook, X and Instagram found out that electric vehicle had high user engagement and towards positive sentiment (Hafize et.al., 2024). This research also exploring on how social media would influence electric vehicle adoption by the consumer. Research by Zhao et.al. (Zhao et.al.,2024) found out that media would influence the market diffusion of electric vehicle. Thereby, this research project aims to assist in providing nuanced information and evaluation by using structured sentiment analysis in gaining real time user's feedback information on strength, limitation and future prospective of the electric vehicle. This would help fostering the market growth of EV industry and ensure wider market acceptance for consumer in making wiser decision.

1.2 Problem Background

The transport sector contributes significant to the greenhouse gas (GHG) emission. Statistical result from climate change scientific research shows that transportation is the largest contribution to the United State GHG emissions in 2022 with a total percentage of 28% ("Climate Change 2022: Mitigation of Climate Change," 2022). While a statistical report by International Energy Agency (IEA) also shows that carbon dioxide (CO2) emission from fossil fuel transportation is about 22% (Malaysia - Countries & Regions - IEA, 2025). Thereby many countries aim for carbon neutrality and intend to achieve net-zero for carbon dioxide emission by the year of 2050 while ensuring affordable and stable renewable energy (Fam & Fam, 2024). Hence, this crisis had triggered a worldwide action in boosting the usage of electric vehicle and minimize the burning of fossil fuel energy vehicles. As the electric vehicle has lesser carbon emission reduction (Littlejohn & Stef Proost, 2022). Many efforts such as improve the infrastructure and tax intensive policy had been implemented for the adaption of electric vehicle (Wibowo & Dovi Septiari, 2023).

Nevertheless, Malaysia also follows closely the global climate issue with commitment and collaboration of Malaysia regarding the global climate issue is also important as one of the major oil and gas producers. Based on the Paris agreement, Malaysia targeted to reduce GHG emission by 35% for the gross domestic product (GDP) by 2030 (Fernandez et. al., 2024). Thereby, Malaysia had a strategies plan in energy transition effort from non-renewable energy to renewable energy resource. (Majekodunmi et al., 2023). One of the strategies plans is targeting 15% of the total industry volume (TIV) by year of 2030 and 80% by 2050 (Siew et. al., 2024). A transition strategy guideline by the Natural Resources and Environmental Sustainability Ministry (NRES) shows that the government intend to transition 50% of government vehicle to electric vehicle ("Portal Rasmi Kementerian Sumber Asli Dan Kelestarian Alam," 2025). Besides, several initiatives had been implemented by the Malaysia government in full exemption of import and excise duties to encourage manufactured of electric vehicle and tax relief for the electric vehicle owner ("Tax Reliefs | Lembaga Hasil Dalam Negeri Malaysia," 2025).

With the effort from the Malaysia government, the Electric Vehicle in Malaysia automotive market is experiencing significant growth, with yearly sales increasing substantially. As the statistic from JPJ Malaysia shows that at the first quarter of year 2025 there were 6827 electric vehicles registered as compared to the first 3 months of 2024 with a record of 4689 electric vehicles registered, which marks an increase of 45.6% (Government of Malaysia, 2025). However, according to Ernst & Young's (EY) fifth annual EY Global Mobility Consumer Index on year 2024 shows that only 25% of Malaysian consumers intent on buying an electric vehicle (EV). Nearly 40 % of Malaysians would intend to buy an internal combustion engine, and 20% would purchase a hybrid car (EY, 2024). Despite implementation of several incentive and more choices of electric vehicle brands, most of the automotive buyer in Malaysia still remain elusive in purchasing the electric vehicle.

The research by Higueras-Castillo et al. (Higueras-Castillo et.al., 2019) proposed that sentiment of customer in electric vehicle would influence the purchase intention. Moreover, research by Kutabish et. Al. (Kutabish, Soares, & Casais, 2023) indicated that consumer nowadays influence by and rely on online reviews to assist

their decision before purchasing. Online reviewer platforms such as YouTube, social media, automotive news and community. Recently, the electric vehicle revolution of Malaysia automotive market had led to surge of online discussion. Online platform such as social media, forums or video review had become key venue for consumer to share experience, discussing and debating about their concern (Ruan & Qin Lv, 2023). Hence, to leverage those reviews on online platform, research on sentiment analysis of electric vehicle has gained significant traction in recent years as text analysis is important for business and organisation to have a insight and understand customer perception. There various recent studies on text mining regarding electric vehicle related topic mostly on conducted on global, with limited focus on Malaysia's unique market dynamics.

There a study implemented deep learning model such as ERNIE combined with deep CNNs to improve sentiment classification accuracy (Wang et.al., 2023). Even though it achieves a high accuracy, there persisting gap lies in the limited sentiment diversity which model only classify sentient into basic categories: positive, neutral and negative. As noted by Wang et. Al. such simplistic classification cannot capture the full spectrum of aspect-based sentiment analysis in public discourse.

There advance natural language processing model, which is bidirectional encoder representations from transformer (BERT) which had significant contribution for sentiment analysis and text mining for text-related data as compared to traditional method. Research by Guven et.al. (Guven, 2021) shows that BERT outperformed Logistic Regression in classifying sentiment analysis of tweet dataset with accuracy of 98.75%. Besides, there is another research shows that BERT model also outperformed model such as Naïve Bayse and Support Vector Machine (SVM) with a accuracy of 95% in analysing customer feedback (Rahman & Maryani, 2024). In addition, there research by Gaurav et.al. (Gaurav, Gupta, & Chui, 2024) proposed that BERT is capable and outperformed traditional model in capturing contextual information and nuanced sentiment classification.

There is research leveraging Large Language Model which is BERT-based model in electric vehicle have demonstrating promising result. However, there is

limited data diversity in this research as it solely relies on small dataset from single platform. This narrow data source may result in potential bias. As proposed by Sharma et. Al. (Sharma, Din, & Ogunleye, 2024) future research can be done on wider dataset to include user generated content from other platform is crucial for more unbiased sentiment analysis. Research done by Wu et.al. (Wu et.al., 2023) also proposed to expand the research study on multiple social media platform and other foreign or domestic platform as the sentiment of public would be different in different countries.

1.3 Problem Statement

Lack of localized research hinder on electric vehicle discourse in Malaysia. There is previous research conducted on other countries and global such as China electric vehicle (Liang, Li, & Chen, 2023) and global YouTube comment on Tesla Motor and Lucid Motor (Sharma, Din, & Ogunleye, 2024) but lack of text mining research done on topic regarding electric vehicle in Malaysia. The second problem statement is the lack of diverse data sources will result in potential bias of the analysis result and incomplete insights into EV-related discussions. Although there is research applied for sentiment analysis, however existing approach rely on single source dataset which might lead to bias data insight (Sharma, Din, & Ogunleye, 2024). The third problem statement is simplistic sentiment classification fails to capture the full spectrum of aspect-based sentiment analysis in public discourse about EVs. There is some of the current research on sentiment classification for electric vehicle using traditional model is simplified, cause failing in capturing nuance consumer discourse information. As proposed by Jena et.al (Jena, 2020) state that there is neglection of public sentiment analysis on topic related to new electric vehicle. Hence, there gap remains in current sentiment analysis of EV domain

1.4 Research Question

The research question is specified as below:

(a) What is the sentiment analysis of the electric vehicle discourse in Malaysia?

(b) How can BERT outperform traditional models and which BERT-based model is better in text mining and sentiment analysis?

(c) How does sentiment and public prospective will vary across platforms?

1.5 Research Objectives

This research aims to mining public discourse on topic related to electric vehicle with the following objective:

(a) To identify, preprocess and explore electric vehicle related text data from multiple online social media and implementing preprocessing procedure to clean data and discover underlying patterns.

(b) To Implement and compare pre-trained Bert-based model in determining structured sentiment analysis.

(c) To analyse the sentiment insight of Malaysian about electric vehicle.

1.6 Research Scope

The following scope are included in this research:

(a) This study will focus exclusively on analysis of discourse within Malaysian social media websites.

(b) The web crawler will be use to crawl on text data published between year 2021 until year 2025.

(c) The analysis only coverage English-language electric vehicle related textual data.

(d) BERT-based model will be implemented for aspect-based analysis, which analysis not just solely on sentiment classification.

1.7 Significant of the study

The significant of this research is that provide an insight of public concerns and adoption barriers surrounding electric vehicle from a user perspective. This would ensure the government and organization can have a valuable understanding on customer opinion that would ensure better solution and incentive provided in order to attract and boost confident of the consumer. Besides, implementation of Bert-based model would provide more nuance analysis as compared to traditional sentiment analysis. In addition, web scraping of the text dataset would provide a valuable real time research study on electric vehicle related discourse. Can be also the released of preprocessed version of electric vehicle social media dataset.

1.8 Thesis Structure

This research project report consists of a total of six chapters and each chapter entails a systematic way for solving the research question posed on title “Sentiment Analysis of Electric Vehicle Discourse Using BERT-Based Language Model”. Hence, thereby this subsection outlines the scope of those chapter.

Chapter 1 is the introduction to the proposed project. This chapter includes an overview of the chapter, introduction, problem background, problem statement, research question, project objectives, project scope and organization of the chapter.

Chapter 2 is the literature review. This chapter provides a systematic review of current work through published articles. The published literature articles are obtained from Scopus, Web of Science, Science direct and other sources. This chapter will review all the existing research articles such as datasets, ML algorithms, performance metrics evaluation, challenges, and opportunities.

Chapter 3 is about methodology. Each of the model phases, methods, algorithms used, and requirements for this project are described and presented in this chapter. This chapter provides a definite, comprehensive and structural understanding of the project.

Chapter 4 is the implementation of experiments. This chapter presents the implementation of the experiments of this project. Implementation of BERT-based model for text mining is conducted.

Chapter 5 is the result & discussion. This chapter showcase the results of the experiments conducted. Suitable evaluation metric is used to show the accuracy and reliability of the results. The discussion of the experiment results is presented and visualized in this chapter. Besides, analysis of the result is also discussed.

Chapter 6 is the conclusion. This chapter concludes the achievements on the project objectives and future work based on the experiment's implementation and the results.

1.9 Summary

This chapter discuss an overview of electric vehicle and the BERT method used for extracting meaningful and valuable insight from the electric vehicle related text. Besides, provide insight on the problem background of the electric vehicle discourse analysis, in order to identifies the research gap and provides significant mapping to the research question, problem statement and research objective. Research scope is also state in this chapter to clarify the specific boundary of the study. A comprehensive overview for the following chapter in this research is also outlined to have a understanding on each chapter specific scope.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provide an overview in reviewing the most recent literature from year 2020 to years 2025 in natural language processing (NLP) related studies. Reviews on what previous NLP model had been implemented for the NLP tasks such as social media sentiment analysis and text mining and which models performed better for the tasks. NLP model such as machine learning model, classical deep learning model, transformer-based language model and advanced NLP model such Large Language Model (LLM). This chapter is aim to underscore the significance and challenges of current social media discourse analysis especially in domain of electric vehicle, hence, detailed critical analysis of recent advancements and methods in these areas is presented, with a focus on current research trends and existing gaps.

2.2 Machine Learning Approach for Social Media Analysis

The machine learning models performed well in analyzing large amount of data and identifying pattern underlying the data. As it is able to understand nuanced and indirect language such as idioms, sarcasm and context specific meaning by train and learn from patterns in a large text dataset (Du et al., 2023).

2.2.1 Support Vector Machine (SVM)

Support vector machine (SVM) classifier is able to be implemented for sentiment analysis for clarifying text data into positive, negative and neutral sentiment. In research by Hussein et.al. (Hussein & Lakizadeh et.al.,2025), SVM performed rather well with an accuracy of 74.96% in classifying Iraqi text data into 3 different sentiment categories.

In addition, research in classifying another language text data which is Urdu Language was proposed by (Azim et al., 2025). In this research SVM correctly classify the sentiments with an accuracy of 87%.

2.2.2 Random Forest algorithm (RF)

Sentiment analysis research in examining public perception about STEM education and artificial intelligent is conducted by Smith-Mutegi et.al (Smith-Mutegi et.al, 2025) using machine learning based approach. In this study random forest algorithm is implemented in analyzing historical post containing terms of STEM and artificial intelligent obtaining through web crawling. The random forest able to classify the sentiment with an accuracy of 84.28%.

2.2.3 Logistic Regression (LR)

In a comparison of methodology for sentiment analysis, Logistic Regression (LR) effectively utilizing with TF-IDF matrix for determining patterns in the text. It relatively well performed with an accuracy of 74.82% in classify sentiment categories as compared to KNN of 50.78% (Hussein & Lakizadeh et.al.,2025). LR also well performed in sentiment identification for the tweet and IMDB movie review data with a accuracy of 87%. It obtains a better results as compared to deep learning model of LSTM and Bi-LSTM.

2.2.4 K-Nearest Neighbor (KNN)

K-Nearest Neighbors (KNN) had a limitation of high dimensional and sparse feature space causing it obtained a relatively poor performance as compared to other machine learning algorithm of LR and SVM. A relatively lowest accuracy result of 50.78% (Hussein & Lakizadeh et.al.,2025)

Tables 2.1 below show the comparison of traditional machine learning model performance in social media sentiment analysis.

Table 2.1 Machine Learning model table

Author(s)	Dataset	Methodology	Performance

Smith-Mutegi et.al, 2025	33,379 historical posts from X application	Random Forest algorithm	accuracy: 84.28%
Hussein & Lakizadeh et.al.,2025	IRAQIDSAD corpus: - 14,141 annotated comments collected from four common Facebook page site of Iraqi	ML models using TF-IDF matrix: - Support Vector Machine (SVM) - Logistic Regression (LR) - K-Nearest Neighbours (KNN)	- Support Vector Machine (SVM): 74.96% - Logistic Regression (LR): 74.82% - K-Nearest Neighbours (KNN): 50.78%
Azim et al., 2025	Urdu Twitter reviews and IMDB movie reviews datasets which obtained from Kaggle	- Support vector machine (SVM) - Logistic Regression (LR)	Both models obtain accuracy of: 87%
(Xu, Wen, Zhong, & Fang, 2025)	17,720 deepfake-related posts and comments on the Reddit	six machine learning model: Logistic Regression(LR), Random Forest(RF), K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree, and Naive Bayes	Accuracy: - Logistic Regression: 78.92% - Random Forest: 78.49% - K-Nearest Neighbors (KNN): 76.97% - Support Vector Machine (SVM): 78.70% - Decision Tree: 76.00% - Naive Bayes: 76.54%

2.1 Deep Learning Approach for Social Media Sentiment Analysis

This section shows the most recent research that are relevant to sentiment analysis using deep learning techniques. The purpose of this section is to find out the most used deep learning model that can perform better than traditional machine learning model.

2.3.1 Convolutional Neural Network (CNN)

A hybrid model CNN-LSTM is used to reflect contextual subtleties of the text, which could help to identify short term pattern such as n-gram recognition in a text and long-term meaning in a text, which making sentiment analysis more accurate. However, in order to reduce the computational complexity of LSTM, a hybrid model of CNN-GRU is also been implement for comparison as GRU model has a simplified architecture. In the research done by Hussein et.al. (Hussein & Lakizadeh et.al.,2025) show that the result of comparison for both hybrid CNN model of CNN-LSTM and CNN-GRU has an accuracy of 74.34% and 74.21%. CNN-LSTM and CNN-GRU performance is slightly similar.

2.3.2 Bidirectional Long Short-Term Memory (Bi-LSTM)

Bidirectional Long Short-Term Memory (Bi-LSTM) is capable to deal with data which has long term dependencies in the sentences (Wei et.al, 2020). As proposed by Mahadevaswamy et.al. (Mahadevaswamy et.al., 2023) stated that the Bi-LSTM had a memory in the model for it to make better prediction. The proposed model can successfully classify the reviews into positive and negative categories with an accuracy of 91.4%. Besides that, a Bi-LSTM approach was also proposed to classify twitter and IMBD movies review in Urdu language and as a comparison of which classification model performed well. The result shows that the Bi-LSTM approach achieve an accuracy result of 84% (Azim et al., 2025).

In addition, Mao et.al, 2023 (Mao et.al, 2023) also proposed an Information Blocks Bidirectional Long-Short term Memory (IB-BiLSTM) model in capturing the sentiment analysis of animated online education texts from students. The model is design to able to capture temporal correlations and long-range dependencies in text data obtain from online animated education. The model shows a compromising result with an accuracy of 93.92%. Proving that implication of multimodal data for emotion recognition can improve sentiment classification.

2.3.3 Long Short-Term Memory (LSTM)

A standalone LSTM model in research study of Hussein et.al. (Hussein & Lakizadeh et.al.,2025) had a comparative lower accuracy as compared to other hybrid deep learning model in classifying sentiment analysis on Iraqi dialect. It achieves a accuracy of 68.26% showing the necessity of enhancement for the architecture. However, a standalone LSTM model in identify sentiment analysis for Urdu language's tweet data and IMDB movie review data had a rather well accuracy of 84% (Azim et al., 2025).

In addition, a hybrid model combining Recurrent Neural Network (RNN)-based Long Short-Term Memory (LSTM) classifier with Bi-directional Gated Recurrent Units (BiGRU) for feature extraction is proposed by (Atlas et al., 2025) for classifying sentiment of product review in e-commerce domain. The accuracy result of 98.79% show that the proposed model is effective in capturing word level meaning and meaning across long term dependencies.

Tables 2.2 below show the comparison of deep learning model performance in social media sentiment analysis.

Table 2.2 Deep Learning model table

Author(s)	Dataset	Methodology	Performance
Atlas et al., 2025	50,253 fashion products reviews from Amazon website	Recurrent Neural Network (RNN)-based Long Short-Term Memory (LSTM)	Accuracy: 98.79 % Precision: 96.64 Recall: 98.7 F1-score:97.43 Auc:99.2
Mao et.al, 2023	Student feedback text data, emotional text data, writing text data, and verbal expression text data	information blocks Bidirectional Long-Short term Memory (IB-BiLSTM)	Accuracy: 93.92% F1-score: 90.34%

Mahadevaswamy et.al, 2023	Amazon Product Review dataset: 104,975 product review	Bidirectional LSTM network	Accuracy: 91.4%
Hussein & Lakizadeh et.al.,2025	IRAQIDSAD corpus: - 14,141 annotated comments collected from four common Facebook page site of Iraqi	- Convolutional Neural Networks with Long Short-Term Memory (CNN-LSTM) - Convolutional Neural Networks with Gated Recurrent Unit called as (CNN-GRU) and - Long Short-Term Memory (LSTM)	Accuracy for each model: - CNN-LSTM: 74.34% - CNN-GRU: 74.21% - LSTM: 68.26%
Azim et al., 2025	Urdu Twitter reviews and IMDB movie reviews datasets which obtained from Kaggle	- Long Short-Term Memory (LSTM) - Bidirectional Long Short-Term Memory (Bi-LSTM)	Both model Accuracy: 84%

2.2 Ensemble learning Approach for Social Media Sentiment Analysis

This section provides an overview in implementation of ensemble learning towards sentiment analysis. Ensemble learning is a technique which combined multiple classifiers to improve accuracy and handling dynamic data. Azim et al. (Azim et al., 2025) proposed a ensemble model to identify sentiment of Urdu Twitter reviews and IMDB movie reviews. An ensemble classifier model named RRLS which is stacks of machine learning and deep learning model consisting Random Forest (RF), Recurrent Neural Network, Logistic Regression (LR), and Support Vector Machine (SVM). The ensemble model in this study is well performed as compared to the other standalone machine learning classifier of SVM, LR, and deep learning model of LSTM and Bi-

LSTM. It achieved an accuracy of 90%, but when synthetic minority oversampling technique (SMOTE) is implemented the performance of accuracy increased 2.77%. However, ensembled learning required high computational complexity, hence the future work on enhancing RRLS model using lightweight structured model.

Tables 2.3 below show the comparison of ensemble model performance toward social media sentiment analysis.

Table 2.3 Ensemble Learning model table

Author(s)	Dataset	Methodology	Performance
Azim et al., 2025	Urdu Twitter reviews and IMDB movie reviews datasets which obtained from Kaggle	Stacks of Random Forest (RF), Recurrent Neural Network, Logistic Regression (LR), and Support Vector Machine (SVM)	Accuracy without SMOTE: 90% Accuracy with SMOTE: 92.77%

(a)

2.3 Transformer-Based/ Large Language Model Approach for Sentiment Analysis

This section shows the implementation of pre-trained BERT-based model approach in previous research for sentiment analysis. Comparison of the BERT-based model and advance BERT-based model is show in this section.

2.4.1 Bidirectional Encoder Representations from Transformers (BERT) model

BERT-based model is effective for sentiment analysis task as it able to capture global context using bidirectional encoder representation. Research proposed by Xu et.al., (Xu, et.al., 2025) implemented 3 BERT-based model to trained on annotated negative data to classify sentiment of anger, fear and sadness. Negative data on comment post at Reddit regarding deepfake perception. The BERTweet-based-sentiment-analysis performed better in identify the nuanced social media text with accuracy of 87.03%. Whereas the BERT-base-uncased-emotion obtain a accuracy of 84.76%, followed by accuracy of 84.32% for BERT-base-uncased.

In addition, a model named AraBERT that conducted in research studies by Hussein et.al. (Hussein & Lakizadeh, 2025) is a pretrained Arabic language model based on the BERT architecture. It was trained based on standard Arabic language and dialects with

an aim to perform natural language processing as for English language. The model in this study were used to evaluate how well does this model performed for sentiment analysis on common four Iraqi dialects Facebook site pages. The results of the AraBERT outperformed all the others machine learning and deep learning model with an accuracy of 90.18%. This proven that BERT is efficient in language understanding.

2.4.2 Robustly Optimized BERT Approach (RoBERTa)

RoBERTa which is an improved version of BERT it had stronger pretrained capabilities, capable of handle large dataset and has more accurate prediction.

In order to improve the model performance of existing Bidirectional Encoder Representations from Transformers-Bidirectional Long Short-Term Memory (BERT-BiLSTM) model in issue regarding lengthy text and complex sentiment expression (Tiwari et.al, 2020), a hybrid RoBERTa model is proposed by Cao et.al (Cao et.al,2024). A hybrid RoBERTa-CNN-BiLSTM-Transformers (RCBT) model which combination of RoBERTa, CNN and Transformer. As CNN can enhance local feature extraction such as word combination, while Transformer can improve long range dependencies using self-attention without increase computational complex. The proposed model achieves a high accuracy of 93.46% in classifying IMDB movie review sentiment analysis.

The Table 2.4 below shows the comparison of BERT based model used for social media sentiment analysis.

Table 2.4 BERT-based model table

Author(s)	Dataset	Methodology	Performance
Cao et.al,2024	IMDB movie reviews datasets officially provided by Stanford University: 50,000 reviews	- BERT-BiLSTM model - BERT-CNN-BiLSTM -BERT-CNN-BiLSTM-Transformer -RoBERTa-CNN-BiLSTM-Transformers	Accuracy for each model - BERT-BiLSTM model: 91.55% - BERT-CNN-BiLSTM: 92.03% -BERT-CNN-BiLSTM-Transformer: 92.30%

			-RoBERTa-CNN-BiLSTM-Transformers: 93.46%
Hussein & Lakizadeh, 2025	IRAQIDSAD corpus: - 14,141 annotated comments collected from four common Facebook page site of Iraqi	AraBERT	Accuracy: 90.18%
(Xu, Wen, Zhong, & Fang, 2025)	17,720 deepfake-related posts and comments on the Reddit	3 BERT-based models: - BERT-based-uncased - BERT-based-uncased-emotion - BERTweet-based-sentiment-analysis	Accuracy: - BERT-base-uncased: 84.32% - BERT-base-uncased-emotion: 84.76% - BERTweet-based-sentiment-analysis: 87.03%

(b)

2.4 Recent Studies on Electric Vehicle Sentiment Analysis Approach

Sharma et.al. (Sharma, Din, et.al.,2024) investigated sentiment analysis on electric vehicles (EVs) using advanced pre-trained transformer models of BERT, XLNet and RoBERTa. XLNet is choosen as improvement of BERT's masked language model which combining autoregressive and autoencoding modelling (Song, Tan, Qin, Lu, & Liu, 2020). Based on the sentiment on YouTube comments related to Tesla and Lucid Motors. RoBERTa emerged as the best-performing model, achieving an accuracy of 92.33% for Lucid datasets while BERT performed well with an accuracy of 93.63%

for Tesla datase. While the study demonstrates the efficacy of large language models (LLMs) in EV discourse analysis, its reliance on small, platform-specific datasets introduces potential bias. The authors acknowledged this limitation and proposed expanding the research to other platforms like Twitter while enhancing data quality through expert annotations. This study sets a foundational benchmark in applying transformer-based models in EV sentiment analysis and encourages future research to adopt diversified datasets and platforms for broader generalization.

Similarly focusing on sentiment detection, Wang et al. (2023) combined ERNIE, a knowledge-enhanced language model, with a convolutional neural network (CNN) to analyze online comments about new energy vehicles (NEVs). Their hybrid approach achieved a notably high accuracy of 97.39%. However, the study's sentiment classification was constrained to basic categories, limiting its depth in sentiment granularity. Their future direction includes the integration of more nuanced model fusion methods and the simplification of model parameters, offering a pathway toward more efficient and expressive sentiment detection.

Wu et al. (Wu et.al.,2023) conducted on research by applying Latent Dirichlet Allocation (LDA) for topic modeling and sentiment analysis through the Natural Language Processing tools of NLPIR-Parser to study public opinions on Sina Weibo, a popular Chinese social media platform. Their findings highlighted specific NEV-related topics such as preferential policies and user sentiment (positive or negative) surrounding them. Despite offering valuable insights into Chinese social discourse on NEVs, the study lacked comparative analysis with international platforms, suggesting the potential for cross-cultural sentiment comparisons and policy benchmarking in future research. This comparative angle could tie in meaningfully with Sharma et al. (2024), who also emphasized dataset diversity and platform inclusiveness.

In addition, another research was also focusing on policy sentiment. Research by Wibowo et.al. (Wibowo et.al., 2023) explored Indonesian public reactions towards electric vehicle tax incentives through tweets data. An Indonesian RoBERTa-based sentiment classifier is proposed in this study for public reaction sentiment classification. The proposed model achieved a accuracy of 71.81%, highlighting there is still room for methodological improvement. The study's narrow geographic and platform focus limited its broader applicability. Future work intends to expand to other social media platforms, offering an opportunity for more comparative or regional

sentiment analyses, especially in line with studies like (Wu et.al.,2023) and (Sharma, Din, et.al.,2024) that proposed in the future work for cross-platform exploration.

Özkara et al. (Özkara et al., 2025) examined social media discourse on electric vehicle by leveraging LSTM and LDA models on English and Turkish tweets from Platform X application. The LSTM model obtain with a high accuracy rates of 96.7% for Turkish and 92.1% for English, the study provides evidence for LSTM's robustness in sentiment detection across multilingual data. However, the research was constrained by only crawling for one-month tweets data window and there were also potential demographic biases due to platform user characteristics were simplicity. Hence, in future work, the authors recommended incorporating BERT-based models for enhance sentiment identification in multilingual contexts and extending analyses across longer timelines and additional platforms such as Reddit and Instagram. This aligns with (Sharma, Din, et.al.,2024) and (Cui et al., 2025). who highlighted the value of incorporating broader social media data to enrich sentiment insights.

Cui et al. (Cui et al., 2025) contributed a novel perspective by analyzing 2818 short video interviews on TikTok with 2,101 of electric vehicle users in Beijing. The Latent Dirichlet Allocation (LDA) Model is used for topic modelling, while Baidu's Large Language Model (LLM), and Random Forest models facilitated the extraction of sentiment for different themes. The LLM achieving 95% accuracy for automobile sentiment evaluation of positive, negative and neutral. Whereas the random forest model is best fit in this study with R-squared (R^2) of 0.8668 for female and 0.8706 for male, whereas Mean Absolute Error (MAE) of 0.1384 for female and 0.1256 for male. Despite its innovation, the study's limitations included geographic and demographic constraints, potential bias from non-random sampling, and challenges in nuanced sentiment recognition for sarcasm. The authors proposed incorporating more platforms and multimodal data to fill current analytic gaps. The study's focus on short video content and multimodal analysis potential aligns conceptually with (Liu et al., 2025), who also advocate for richer data formats beyond text.

Liu et al. (Liu et al., 2025) proposed quantitative economic research by forecasting EV sales using a hybrid BERT-Bi-LSTM model. Drawing from monthly sales data, forum posts, and gasoline prices, the model achieved an accuracy of 94%, showing strong predictive power. However, limitations arise from inadequate factor selection, incomplete data decomposition, and the lack of multimodal data integration. Future directions include incorporating videos, images, and performing outlier analysis to

capture sudden market shifts. This study complements (Cui et al., 2025) in its call for multimodal data analysis and bridges sentiment analysis with predictive economic modeling—broadening the scope of EV-related research.

While not directly EV-related, the study by Pascal et.al. (Pascal et.al., 2025) introduced a deep learning approach using EEG data to predict user behavior in electronic markets. The proposed SIEPTNet model used CNNs to predict all five personality traits from EEG data, showing improved accuracy performance with Gaussian filtering. Despite not being situated within the EV domain, the methodology's potential in capturing cognitive or emotional user responses to EV-related marketing or interfaces could complement sentiment studies like those by (Cui et al., 2025) or (Liu et al., 2025), especially if integrated into multimodal frameworks.

Intercorrelations among these studies highlight a trend toward multimodal data integration (Cui et al., Liu et al., 2025), cross-platform social media analysis (Sharma et al., Wu et al., Özkara et al.), and transformer model dominance in sentiment extraction (Sharma et al., Wang et al., Wibowo et.al.). There's a consistent emphasis on moving beyond textual data and basic sentiment classification, pushing toward nuanced, scalable, and contextually rich sentiment modeling frameworks. These efforts collectively contribute to a more comprehensive understanding of EV discourse and user perception, crucial for both industry decision-making and policy formulation. The Table 2.5 below shows the summary of recent study in electric vehicle towards sentiment analysis.

Table 2.1 Table summary of electric vehicle sentiment research

Author(s)	Dataset	Methodology	Performance	Limitation	Future Work
Sharma, Din, et.al.,2024	Lucid Motors and Tesla Motors-related YouTube data	BERT, XLNet, and RoBERTa pre-trained transformer models	Accuracy for Tesla dataset: - BERT without Fine Tuning: 9.75% BERT with Fine Tuning: 93.63% - RoBERTa without Fine Tuning: 5.34	- Bias of the dataset due to limited data resource from one social media platform - adopt human annotation by experts to	- applying this study methods on other social media datasets such as Twitter

			RoBERTA with Fine Tuning: 92.12% - XLNet without Fine Tuning: 42.26% XLNet with Fine Tuning: 90.10% Accuracy for Lucid dataset: - BERT without Fine Tuning: 37.06% BERT with Fine Tuning: 90.33% - RoBERTa without Fine Tuning: 17.30% RoBERTA with Fine Tuning: 92.33% - XLNet without Fine Tuning: 43.88% XLNet with Fine Tuning: 90.90%	improve on data quality - comparing other transformser model using diverse EV's companies dataset	
Wang et al, 2023	dataset of new energy vehicle comments collected from multiple automotive	Hybrid model of Enhanced Representation through kNowledge IntEgration (ERNIE) and	accuracy rate of 97.39%	Limited sentiment diversity which model only classify sentient into basic categories	Explore advanced model fusion methods at a deeper level and simplify model

	social media platform	a deep (Convolutional Neural Network) CNN			parameters to reduce complexity
Wu et.al., 2023	Sina Weibo	- LDA topic modelling - Sentiment analysis is performed in the NLPIR-Parser platform	- positive, negative sentiment - on which NEV topic - preferential policies	- only focuses solely on public posts of China's social media platform - do not implement posts on other foreign social media platforms such as Twitter, Facebook, and other foreign social media platforms - public's attitudes and sentiments towards NEVs vary in different countries	Analyse and compare the public's opinion on social media platform of domestic and foreign - Integrate the policies of other nation to provide policy recommendation for the adoption of NEVs in China.
Wibowo et.al., 2023	Twitter data	Indonesian RoBERTa-Based Sentiment Classifier	accuracy: 71.81%	only focus on Indonesia tweet data	extend this study to use broader social media data
Özkara et al., 2025	X social media platform - consists of	- Long short-term memory (LSTM)	Accuracy: - 96.7% for Turkish tweets	- Not capturing long term trends with limited to	- exploring deep learning models such

	6000 English and 891 Turkish tweets	model - Latent Dirichlet Allocation (LDA)	- 92.1% for English tweets	one month of tweets data - data bias as the platform demographics consist mostly of younger and tech-oriented users - Language-specific preprocessing and model accuracy could also affect comparability across Turkish and English tweets.	as BERT or transformer-based architectures to enhance sentiment detection, especially in multilingual contexts data - Lengthen the data collection period - have a boarder picture of public discourse by including other social platforms such as Reddit or Instagram
Cui et al., 2025	2818 short videos about EV experience form Tiktok User named EV USERS UNION: street interviews by a TikTok user in	Text Modelling: - Latent Dirichlet Allocation (LDA) Model sentiment analysis: - LLM developed by	- LLM: 95 % accuracy - Random Forest: For Female R^2: 0.8668 MAE: 0.1384 For male R^2: 0.8706 MAE: 0.1256	- limited geographic and demographic diversity - sampling bias with reliance on single platform and use of non-random sampling	- integrating additional analytical methods to obtain and analyse user's personal information

	Beijing which contain 41 hours length of video interviews with 2101 electric vehicle (EV) owners	Baidu - Random Forest		- Limitation of LLM in detecting nuanced emotional shift such as sarcasm due to general domain pre-training of the foundation models - Manual interpretation is needed to assign final topic label in LDA analysis	- incorporating more additional platforms to obtain relevant data sources - focusing on less developed region, to understand EV consumption in that regions
Liu et al., 2025	forum text from Auto Home: Car owners homes and East Money Information to collect monthly data on electric vehicle sales and gasoline prices.	Bidirectional Encoder Representations from Transformers -Bidirectional long short-term memory (BERT-Bi-LSTM)	Accuracy: 94%	- Limited influencing factors: Current research lacks comprehensive selection of factors affecting EV sales. - Incomplete data decomposition: The model does not fully address decomposition of complex nonlinear data.	- Integrate multimodal data (video, audio, images) to capture richer consumer sentiment. - Utilize large AI models for better multimodal data extraction and fusion. - Conduct

				- Insufficient feature extraction: Prior methods struggle with capturing nonlinear feature information and contextual text understanding. - Lack of model robustness: Existing frameworks are not stable or generalizable enough for complex data. - No use of large AI models: The study does not explore large models for multimodal data (e.g., video, audio, images). - No multimodal data analysis: Only textual data is analysed, missing insights	outlier analysis to handle unexpected events like policy changes and reduce prediction distortion. - Enhance data decomposition techniques for improved handling of nonlinear features and seasonal effects.

				from other formats like EV ads or user videos. - No outlier analysis: Sudden changes from policy or market fluctuations are not accounted for, which could affect prediction accuracy.	
Pascal Penava & Buettner, 2025	LEMON Data which consist of electroencephalographic data - publicly available resting-state EEG data from 203 participants whose data was tagged using 62 digitized EEG channels.	CNN architecture called Subject-Independent EEG-based Personality Trait Network (SIEPTNet)	Average evaluation metrics over 10 folds without Gaussian filtering across all five personality traits - Openness: 0.6069 - Conscientiousness: 0.6400 - Extraversion: 0.6394 - Agreeableness: 0.6779 - Neuroticism: 0.6208 Average evaluation metrics over 10 folds with Gaussian filtering across all five personality traits - Openness: 0.6403 - Conscientiousness:	- Assumption of personality traits are stable, which may reduce adaptability if EEG-based traits change over time - Only explored 1D deep learning architecture, limiting comparison with potentially better 2D or 3D CNN models. - No real-world testing, so	- Test personality trait stability over time using EEG data from the same individuals across different periods - Explore alternative model architectures like 2D and 3D CNNs for better

			0.6398 - Extraversion: 0.6818 - Agreeableness: 0.6874 - Neuroticism: 0.6539	practical utility and robustness remain unverified.	performance comparison. - Apply and evaluate the engine in real-world settings, especially in electronic markets, to assess practical effectiveness.

2.5 Auto Sentiment Labelling Modelling

This subsection will discuss on method for automated sentiment analysis used by previous researcher for sentiment analysis on user generated content or reviews on the social media, news and forum platforms. Research done by Borg & Boldt, 2020 et. al. (Borg & Boldt, 2020) used Valence Aware Dictionary for Sentiment Reasoning (VADER) in compliment with Swedish sentiment lexicon to predict initial sentiment labelling for the customer email responses. This label was then used to train the Support Vector Machine model for sentiment classification and future sentiment prediction for threaded email prediction. While, there were also another research implemented VADER for automatic sentiment labelling for social media post related to celebrity endorsements. The labelled data was also then used to train SVM classifier, yielding satisfactory performance in terms of F1-score and accuracy. This shows the practical utility of initial annotation in marketing related sentiment analysis using VADER (Syahputra et al., 2024).

In addition, the comparison of four automatic sentiment pretrained model which are VADER, TextBlob, Flair and FinBERT are implement in research by Alonso etl.al (Alonso & Sicilia, 2024). This research used human labelled data by three human annotators per sample and consensus voting on CrytoLin dataset as a benchmark to see which pretrained auto labelling model performed best. It shows that the FinBERT

performed the best. Khalid et.al. (Khalid et al., 2024) proposed a sentiment majority voting classifier (SMVC) to automatically labelling the tweet dataset related to deepfake. SMVC select the sentiment based on the majority voting among three lexicon based model which are TextBlob, AFINN and VADER for a more robust automatic labelling process. The labelled data was then used with transfer learning features in LSTM, decision tree and logistic regression achieving a high accuracy.

There was also other automated sentiment approach which using zero-shot classification for initial sentiment label assignment based on polarity score. Then using a pretrained BERT model for contextual and semantic feature extraction. The labelled sentiment dataset was then input into a Convolutional Bi-directional Recurrent Neural Network (CBRNN) for sentiment classification and this model shows a strong performance across various dataset (Sayyida Tabinda Kokab et al., 2022). Research by Ullah et.al.(Ullah et al., 2023) develop a sentiment classification model called QLeBERT for targeting product quality sentiment evaluation. The model automatically labelled customer review data by integrating a custom build product quality lexicon, N-gram and BERT derived from review content. This labelled features was then input into a BiLSTM model for classification and resulting in a strong classification performance, especially in binary quality assessments.

Table 2.6 Table summary of auto sentiment labelling modelling

Author(s)	Dataset	Model used	Analysis	Labelling Method	Contribution Analysis
(Borg & Boldt, 2020)	168010 of Swedish telecom company customer support emails grouped into 69900 threads	LinearSVM	F1-score: 0.834 AUC: 0.896	VADER-based sentiment labelling and Swedish sentiment lexicon	Demonstrated effective sentiment extraction and prediction for future email sentiment

(Syahputra et al., 2024)	Celebrity endorsement from Twitter, Facebook and Instagram	SVM classifier	Accuracy:76.92 F1-score:87%	VADER-based sentiment labelling	Combined VADER labelling and SVM for analysing public sentiment toward endorsement
(Alonso & Sicilia, 2024)	Cryptocurrency news	Benchmarked with VADER, TextBlob, Flair, FinBERT	FinBert performed the best with human labelled as benchmark	Manual annotation by diverse annotators with consensus voting	Provided a high quality manually labelled corpus on CryptoLin for sentiment model evaluation
(Khalid et al., 2024)	Deepfake related Twitter tweets	- LSTM - Decision Tree - Logistic Regression	Logistic Regression performed better with accuracy: 98.9%	Sentiment Majority Voting: Combined of TextBlob, VADER, AFINN	Implemented a SMVC labelling and hybrid fusion to enhance classification accuracy.
(Sayyida Tabinda Kokab et al., 2022)	US airline reviews	BERT-based CBRNN model	Accuracy of CBVRNN:0.90	Zero shot classification based on polarity score then automated labelling using pretrained BERT model	Address noisy data and OOV using semantic embedding and zero-shot labelling

(Ullah et al., 2023)	Customer product review	BiLSTM model	F1-score: 0.91	Custom build product quality lexicon, N-gram and BERT	QLeBERT and combination of domain specific lexicon and embedding is proposed for product quality sentiment.

2.6 Depth and Analysis

The literature from Sections 2.2 to 2.6 reveals a progressive evolution in sentiment analysis techniques applied to electric vehicle (EV) discourse, encompassing a range of traditional, ensemble, deep learning, and transformer-based methods. Early sentiment analysis approaches employed classical machine learning algorithms such as Naive Bayes and Support Vector Machines (SVM), with limitations in handling complex linguistic contexts. These traditional methods, while foundational, often lacked the semantic depth needed to capture nuanced sentiment, particularly in large and diverse datasets. Ensemble learning approaches then sought to mitigate these limitations by integrating multiple algorithms to improve accuracy and robustness. Studies employing voting classifiers, Random Forests, and hybrid frameworks like CNN-LSTM and Bi-LSTM-Attention demonstrated significant gains in performance, particularly in capturing contextual dependencies. However, they still struggled with language ambiguity and required extensive feature engineering, which constrained scalability across different datasets and platforms.

The shift towards deep learning and especially transformer-based models marks a major turning point in the field. BERT and its variants (RoBERTa, XLNet, ERNIE) consistently outperform earlier methods by leveraging contextual embeddings and deep bidirectional understanding of language. Transformer models, as shown in the studies by Sharma et al., Wang et al., and Wibowo et al., not only improve sentiment classification accuracy but also facilitate cross-lingual and domain-specific adaptation. The integration of domain knowledge, as in ERNIE's case, and the inclusion of platform-specific nuances—such as Twitter, YouTube, and Sina Weibo—underscore

the importance of tailoring sentiment analysis tools to the characteristics of both the data and the user base. Yet, challenges remain, particularly in ensuring dataset diversity, platform representativeness, and mitigating biases arising from small, localized, or temporally limited datasets.

Emerging trends from recent studies further emphasize the importance of multimodal and cross-platform sentiment analysis. The application of Latent Dirichlet Allocation (LDA) for topic modeling, as used by Wu et al. and Cui et al., reflects an effort to contextualize sentiment within broader discourse themes such as policy, environmental awareness, or economic concerns. These topic models provide a thematic lens through which sentiments are better understood in relation to public concerns. Additionally, the integration of models like Random Forests and even EEG-based CNN frameworks in the study by Pascal et al. signals a move toward interdisciplinary approaches, combining behavioral insights with computational methods to enrich sentiment interpretation. Furthermore, the studies highlight a growing interest in video-based and multimodal data—such as those from TikTok and sales forecasts—where sentiment is inferred not just from text but also visual, behavioral, and contextual cues.

Overall, there is a clear trajectory toward more holistic, context-aware, and technologically sophisticated sentiment analysis approaches in the EV domain. Transformer-based architectures dominate current methodologies due to their scalability and performance, yet there is a consistent call across studies for enhanced generalizability, multimodal integration, and real-time applicability. The field is steadily moving beyond surface-level polarity classification toward deeper understanding of consumer perception, emotional resonance, and market dynamics—all of which are critical for informing EV-related policy decisions, marketing strategies, and user-centered technological development.

In addition, the automatic sentiment literature review on past researcher paper shows that automatic sentiment labelling method vary in complexity and approach. Ranging from rule-based lexicon methods which consist of VADER, TextBlob and FINN. Voting based hybrid labelling strategies to improve robustness and zero shot classification for label inference without supervised training. There was also methos of custom lexicon construction and contextual modelling for domain specific sentiment detection. Some studies focusing on highlighting the importance of high quality human manual annotation as model benchmarking while there is also other

research that focus on building scalable automatic labelling pipeline for real word sentiment analysis. Thereby, these research shows the essential of automatic labelling in modern sentiment analysis pipeline and is typically enhanced by combining lexicon-based, machine learning and pretrained language model approach toward the application domains and data availability.

2.7 Conclusion

In conclusion, the literature reviewed demonstrates a clear progression in sentiment analysis techniques applied to electric vehicle (EV) discourse, moving from traditional machine learning methods to advanced deep learning and transformer-based models. While early approaches provided foundational insights, recent studies have emphasized the superior performance and contextual understanding offered by models like BERT, RoBERTa, and XLNet. Additionally, there is a notable shift toward incorporating multimodal data, cross-platform analysis, and topic modeling to capture the complexity of public sentiment. Despite significant advancements, challenges such as data bias, limited platform diversity, and the need for more nuanced sentiment classification remain. These gaps highlight opportunities for future research to explore more inclusive, scalable, and context-rich frameworks, ultimately contributing to a more comprehensive understanding of EV-related public opinion and user behavior. In addition, there is also past research studied reveal approach such as combining rule based, statistical and deep learning method had been implemented to improve the labelling accuracy and model performance. Automatic labelling is not essential for scaling sentiment analysis across large dataset but also plays an important role in bootstrapping supervised leaning models when labelled data is scarce.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter will provide a comprehensive explanation of techniques and process that will be implement in this research project. Subsequently, the library and packages used will be listed. This project framework consists of phases starting from identification of problem to data collection, data preprocessing, sentiment analysis model development, model hyperparameter tuning, until model evaluation measure and verification phase. However, the phase 1 research gap will not need be discussed in this methodology as it had been discussed in detailed in chapter 2 literature review. The detailed flow of the research work phase starting from data collection until model evaluation will be shown to ensure systematic workflow and resulting in effective completion of the research.

3.2 Phase 2: Data Collection

The second phase of this project is to collect a primary dataset in field regarding electric vehicle discourse. The process in this stage is to achieve objective 1. For the dataset collection, the data for this study were sources from Reddit. It was a global platform with diverse individuals from different social and economic backgrounds. Besides, the social media environment is anonymous and real-time updated which provide a platform for more objective public opinions. To collect the project data, web scraping will be implemented to crawl posts and comments containing the keyword related to ‘electric vehicle’ and specific to region of ‘Malaysia’. In order to web scrap the data from Reddit website, a Reddit’s official API is access through the PRAW (Python Reddit API Wrapper) library. The metadata collected is as shown in table 3.1. The architecture flowchart of web scarping is shown in figure 3.1.

Table 3.1 Description of Web Scrap Data

Variable	Descriptions
Post_id	Unique ID of the original post submission
title	Title of the Reddit post (for context in sentiment analysis).
Self_text	Body text of the Reddit post (it would be empty or NAN if it is link or image)
Post_score	Total upvote minus downvotes for the post
Upvote_ratio	Ratio of upvote to total votes for the post (ranging from 0 to 1)
Num_comments	Total number of comments under the posts
Post_created_utc	UTC timestamp when the post was created (in Unix epoch format)
Comment_id	Unique Reddit id of the comment
Comment_body	Unique Reddit id for the comment
Comment_score	Total upvote minus downvote for the comments
Comment_awards	Number of Reddit awards the comment received
Comment_created_utc	UTC timestamp when the comment was created (in Unix epoch format)

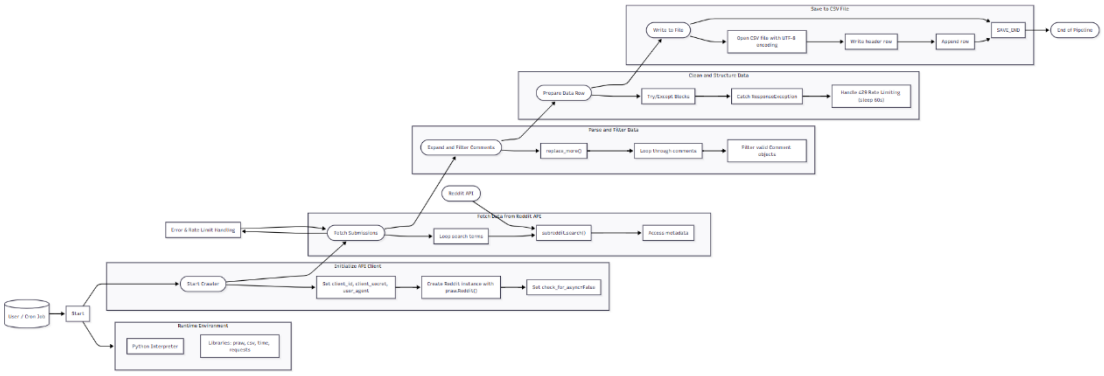


Figure 3.1 Architecture of Web Crawler

3.3 Phase 3: Data Preprocessing

In this phase, the collected data will be pre-processed and the objective 1 will also be fulfilled. The process of data preprocessing holds great significance in the field of natural language processing (NLP) by having a clean and structured data to ensure only relevant information is considered. Text data is often messy, noisy, and unstructured, which can affect the quality and accuracy of your NLP models learning algorithms. In this phase, the data cleaning techniques that required for text cleaning will be shown.

3.3.1 Remove Unwanted Character

Remove unwanted character such as punctuation, numbers, symbols, HTML tags, excessive whitespace, mentions and emojis. The characters will cause noise and ambiguity to the data, and may not relevant for the sentiment task. Regular expression can be used to remove the non-alphabetical by import the re module to use the ‘re.sub()’ function. However, emoji and emoticon contain rich sentiment indicators but it requires the conversion of these to text for Bert to process. BERT tokenizer will split

the emojis which converted to text into sub words as a sentiment clues. Hence, the retain or removal of some unwanted character is considerable but the irrelevant one will be prior to be remove.

3.3.2 Language Filtering

As the data were web scrap from platform which can be access globally, language filtering is required to only cater preferred language for this project. The preferred language for this project is scope to be English only. However, as Malaysia is a multilingual country with diverse population, the discourse in Malaysia sometimes includes a mix of Malay and English or Chinese with English. Hence, language detection is applied to either remove unmeaningful text that is not English language or retain meaningful sentiment value by translating non-English text to English language. The language detection library in Python 'langdetect' which is port of Google's language detection library can be install to identify the language of a given text. The detect() function of 'langdetect' library is capable in identification of 55 languages. Besides, there is also another open-source python library for language detection, which is 'lingua_py'. It can detect a total of 75 languages. Next, the detected non-English text can be translated by using translation library such as 'googletrans' or 'DeepL'. Google translation is optimal for general translation while DeepL is suitable for technical translation. There is research shows that DeepL translation library accuracy is more highly accurate and nuanced translation as compared to Google translation library.

3.3.3 Text Normalizations

Normalization is process in converting text data into standardized and consistent format. It helps reduce the complexity and variability of the text. The normalization process involves stemming, lemmatization, stop word remover and lowercase. Stemming and lemmatization can help group words with similar meanings thus reduce the size and complexity of the vocabulary. Stop word remover can reduce

the redundancy of the text by removing the most frequently uninformative word such as articles, prepositions, conjunctions and pronouns. Besides, lowercase can reduce vocabulary size as different algorithms will classifies uppercase letter and lowercase letter such as 'Hello' and 'hello' as 2 different words. Those normalization process can be done through import of Natural Language Toolkit (NLTK) or spaCy library; which is advance Natural Language Processing in Python.

However, that text normalisation is needed before implementation of model training using approach such as traditional machine learning model or deep learning model. Whereas, for Bert-based model some text normalisation is not needed. The implementation and method for BERT in text normalization is shown as below:
1. For Word lemmatization and stemming:

In BERT, lemmatization and stemming is not needed as BERT implemented a tokenization method named WordPiece Tokenization. This is particularly useful for handling unseen words and deal effective with compound or morphologically complex words which could maintain sematic details than stemming. Breaking down the words into subword units rather than relying on whole words. Handling subwords by initially splitting words into characters with a ## prefix for all.

For example the word 'hugging' is token into 'h', '##u', '##g', '##g', '##i', '##n', '##g'. Then it merged subword pairs iteratively and used the formula:

$$score = \frac{(freq\_of\_pair)}{(freq\_of\_first\_element \times freq\_of\_second\_element)} \quad (3.1)$$

If given a corpus with ('hug', 10), ('pug ', 5), ('pun ', 12), ('bun', 4) and ('hugs', 5), word piece start with the token such as 'h', '##u', '##g' and learn merge such as ('##u', '##s') → '##gs', then ('h', '##u') → 'hu', followed by ('hu', '##g') → 'hug', until it form full sub words like hugging through optimized merges.

2. For Lowercasing or Uppercasing:

During preprocessing for lowercasing and uppercasing, need to encounter whether or not to retain the casing before training using Bert-based model. As some Bert-based model such as BERT-based-cased, BERT-based-uncased and BERT-based-multilingual-cased had different implementation requirement on capitalization.

For BERT-based-cased, it will train on task with retaining its capitalization either uppercase or lowercase. It ensures better performance for named entity recognition. While for BERT-based-uncased, it will convert the text to lowercase itself. For BERT-based-multilingual-cased, lowercasing of word is not applied.

3. For Stopword Remover:

During preprocessing for informative stop word remover is not beneficial especially for sentiment analysis using BER-based model. As Bert is a deep contextual language model, which capable in understanding the meaning of a word based on its surrounding context. Stopword such as is or the often had syntactic or grammatical importance that helps BERT to capture the sentence structure. Besides, stop word is semantically for sentiment classification. For example, removing of the word 'not' in the sentence would influence the semantic meaning of the sentence. Nevertheless, BERT's WordPiece tokenizer can effectively handle the stopword as it often short and frequent.

3.4 Phase 4: Automatic Sentiment Labelling

Vader (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon rule-based sentiment analysis pretrained model which use to identifying and categorising the opinion expressed in a sentences or text. Determining whether it is positive, negative or neutral sentiment. It particularly applied for social media content. Vader assigned a polarity score from -1 which is most negative to +1 which is most positive for lexicon of words. In order to assign the labelling using VADER, a compound score can be assigned. If the compound score more than and equal to 0.05 it is positive, while

if the compound score is less than and equal to 0.05 then it is label as negative and otherwise it would be neutral. The compound score is computed using the formula shown below.

$$compound\ score = \frac{sum\ of\ all\ valence\ score}{normalised\ within\ a\ range\ of [-1,1]} \tag{3.1}$$

3.5 Phase 5: Train-test Split

The sample data that had been labelled is then undergoes train test split to get a training and testing dataset. This is important to ensure no overlap between train and test data to avoid bias. The training data is for the BERT classifier to learn the rules or pattern. Resulting a classification model. While the testing dataset, which is new data that unseen by the classifier will be used to test the classification model for the performance. Stratified sampling is applied for the data splitting to preserve the proportion of classes to prevent bias and handling imbalance of the labelled sentiment. In scikit learn train test split, stratified sampling is used by stating stratify parameter in the train_test split() method.

$$n_k = n \times \frac{N_k}{N} \tag{3.2}$$

Where:

N: Total number samples in dataset

n_k: Number of samples to take from class k

n: Desired sample size / size of the test set

N_k: Total sample in class k

3.6 Phase 6: Feature Transformation and Preparation

In text classification and prediction, Natural Language Processing (NLP) is one of the fields in computer science essential to use for linguistic machine learning study. Vectorization or text representation is one of the ways used to extract features from text and hence undergo text vectorization which converts text into a vectorized numerical value and is used as input for model training of the features this method is usually called feature extraction.

3.6.1 BERT-based feature extraction

The feature extraction process in BERT starts from sequence encoding, which the raw text is transformed into structured numerical inputs. After tokenization of the sentence using WordPiece tokenization, the input is represented as input_ids. Input_ids is the input sentence in the form of sequence of integer indices. Special tokens such as CLS (classification), SEP (separator), and PAD (padding) are embedded within these indices to mark sequence boundaries and fill shorter inputs. CLS are at beginning of the sentence while SEP are at the separate segment of sentence. To differentiate real tokens from padding, an attention_mask is generated, assigning 1 to valid tokens and 0 to PAD placeholders. This is to ensure the model ignores padded positions during self-attention calculations. In addition, for tasks involving paired sequences such as question-answering, token_type_ids are introduced: 0 for tokens in the first sentence and 1 for the token in the second sentence. This would enable the model to distinguish between it.

These sequence which are input_ids, attention_mask and token_type_ids is then been processed into token embeddings. It are summed with positional embeddings which are learned absolute position indicators and segmentation embeddings which are sentence identifiers. This combined input representation is the forward into BERT's based which contain 12 encoder layers where multi-head self-attention dynamically weights contextual relationships. For example, linking the token 'love' to 'Tesla' in 'I love Tesla'. While the feed-forward networks refine local interactions.

The final state which is particularly the CLS token's representation, it encode task-specific features and ready for downstream classification. Hence, this structured encoding ensures the model captures nuanced semantics while handling variable-length inputs efficiently.

3.7 Phase 7: Build Bert-based Electric Vehicle Sentiment Analysis Model

In this stage, the model in classifying the sentiment analysis of electric vehicle discourse will be build using pre-trained BERT-based model. This phase is implemented to achieve objective 2 by comparison of which BERT-based model is better in performance of sentiment classification for topic regarding electric vehicle discourse.

3.7.1 BERT-based uncased

The BERT-based uncased model consist of 12 transformer encoder layers that processed the lowercase input sentence using token, position and segmentation embedding. Each layer apply the multi-head self-aatttention and a feed-forward network to produce contextual embedding. The final CLS token representing the whole sentence. The CLS vector is then passed through a dense layer with softmax activation and optimised using cross entropy loss for sentiment classification. The architecture flow of BERT-based uncased model is shown as figure below.

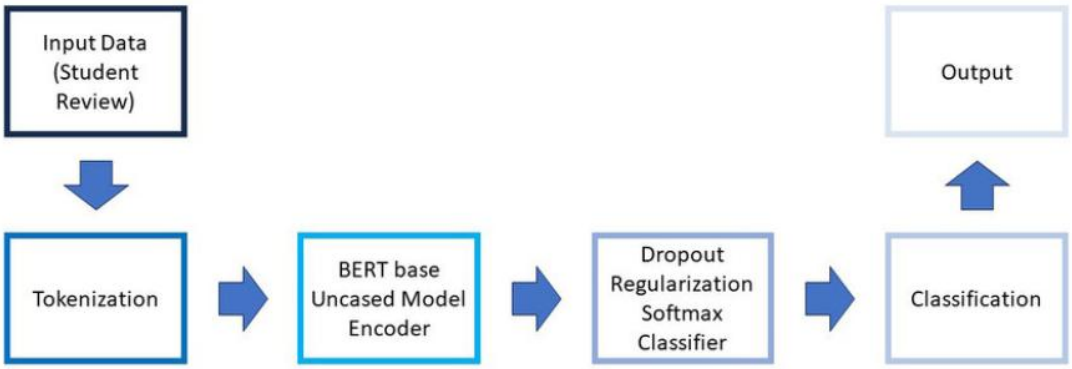


Figure 3.2 Example of BERT-based uncased model

3.7.2 RoBERTa-based model

RoBERTa is optimised and robust version of the BERT as it was build based on architecture of BERT. RoBERTa is pretrained on larger dataset than BERT with bigger batch size and longer sentences. Trained on dataset such as Book corpus, CCNews, OpenWebText and stories.

The input representation for each token is represented as the sum of three embeddings. Below show the key component and equation for the RoBERTa.

1. Input Representation

$$Input(x_i) = Token\ embedding(x_i) + sengment\ embedding(s_i) + position\ embedding\ (p_i) \quad (3.3)$$

Where

x_i: the i-token in the input sequence

s_i: segment embedding of sentence

p_i: position embedding

2. Transformer Encoded Layer: Each layer applies multi head self attention followed by a feed forward network.

3. Masked Language Modelling where the token are randomly masked and the model tried to predict it.

4. Dynamic Masking is when training the masking pattern will changes every time a sequence had been seen. This approach make it more robust as compared to BERT that performed static masking.

3.8 Phase 8: Experiment on Hyperparameter

In this phase will show the hyperparameter tuning that can be implemented for increased the performance of the model.

3.8.1 Epoch

Epoch is the tuning of the number of complete passes through the training dataset. This is to act as a control how many timed the model can learn from the entire dataset. It cannot be set too low to avoid underfitting, or either too many epochs which would lead to overfitting.

3.8.2 Learning Rate

Learning rate is the size of step at each iteration to update the model weights. It is used to determine how quickly the model converges. It is typicaaly in range of 2e-5 to 5e-5 for BERT. Besides, learning rate scheduler such as linear decay and warm up can also be implement. The learning rate formula is as shown below.

$$\theta = \theta - \eta \cdot \nabla\theta L(\theta) \tag{3.4}$$

Where θ = model parameter, η = learning rate, $\nabla\theta L$ = gradient of loss

3.8.3 Optimizer

An optimizer such as BERTAdam (AdamW) optimizer can be implemented. The formula is shown below.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \widehat{m_t} &= \frac{m_t}{1-\beta_1^t}, \widehat{v_t} = \frac{v_t}{1-\beta_1^t} \\ \theta_t &= \theta_{t-1} - \eta \times \frac{\widehat{m_t}}{\sqrt{\widehat{v_t} + \in}} \end{aligned} \tag{3.5}$$

Where:

- θt: Parameters (weights) at iteration t
- gt: Gradient of the loss function L(θt)
- η: Learning rate (step size)
- $\beta 1$, $\beta 2$: Exponential decay rates for first and second moments

- ϵ: Small constant to avoid division by zero (numerical stability)
- λ: Weight decay coefficient (L2 regularization)

3.9 Phase 9: Model Evaluation and Performance Metrics Evaluation

In this phase, the evaluation measurement will be used to evaluated the experiment output from phase 6 and 7. The evaluated output of each model comparison and hyperparameter tuning of each model will be recorded and further analysis for each respective performance based on performance metric will be conducted. This phase is to achieve objective 3, where the analysis on the sentiment insight of Malaysian discourse about the electric vehicle can be evaluated. Evaluate how well the BERT model performs in identifying positive, negative and neutral sentiment.

The common evaluation metric for sentiment analysis classifier is accuracy, precision, recall and F1 score, confusion matrix and Matthew's correlation coefficient. The definition and formula for the evaluation metric is show as below:

(a) Accuracy:

Accuracy indicates the proportion of correctly predicted instance over the total number of predictions. It is mathematically defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.6}$$

(b) Precision:

Precision metric is employed to address the inadequacies of Accuracy. It indicates how accurate the percentage of positive predictions. High precision indicates a low false positive rate. The formula for calculating Precision of binary and multi-class classification is as below:

Binary Classification Equation:

$$Precision = \frac{TP}{TP + FP} \tag{3.7}$$

Multi-Class Precision (Averaged over all classes):

$$Precision_{macro} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \tag{3.8}$$

(c) Recall

Recall is the ratio of how many correctly predicted objects to the total number of predicted objects. This metric is utilized to evaluate the model's capacity to forecast positive outcomes.

Binary Classification Equation:

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

Multi-class Recall Equation:

$$Recall_{macro} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i}\frac{TP}{TP + FN} \tag{3.10}$$

(a) F1-score

F1-score is the average of precision and recall, considering both false positive and false negative. It is more useful than accuracy when dealing with imbalance class distribution. Accuracy is only ideal when the false positive and false negative are similar.

Binary Classification Equation:

Multiclass F1-score Equation:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.11}$$

$$F1 - score_{macro} = \frac{1}{N}\sum_{i=1}^{N} 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.12}$$

(b) Matthew's Correlation Coefficient (MCC)

Matthew's correlation coefficient (MCC) or phi coefficient is the performance metrics and model evaluation measurement of the correlation of the true classes with the predicted labels. This metric is used in the machine learning field as binary and multiclass classification quality measurement. The MCC

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.13}$$

measurement is distributed unevenly when the datasets are imbalanced (Zhu, 2020) and for the unequal prediction class frequencies. (Jones & Ward, 2003)

(c) Confusion Matrix

Confusion matrix is a two-dimensional matrix which is used to evaluate and inspect errors in classification cases and is also used as the tuning parameters for threshold detection. Confusion matrices can lead to multi-class calculation with $n \times n$ matrices. The misclassified items for each pair of the class elements are encoded using a confusion matrix. From Figure there are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) together to form a matrix configuration. The TP and TN show that the model correctly predicts and the actual conditions are positive and negative respectively. However, the FP refer to the models that provide wrong predictions of the negative class (predicted condition as a positive result while actual condition as a negative result) while the FN refer to the models that provide wrong predictions of the positive class (predicted condition as a negative result while actual condition as a positive result).

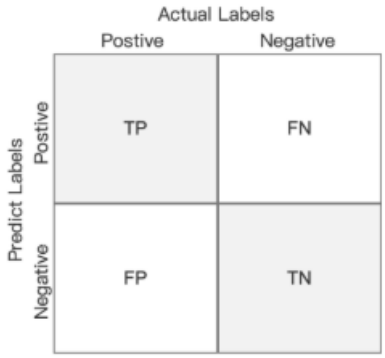


Figure 3.3 Confusion Matrix

3.10 Phase 10: Final Report and Project Wrap Up

Lastly, the whole project report will be wrapped up. Future work and limitation of this project will also be discussed.

3.11 Conclusion

This chapter provided a detailed flow of each phases in the methodology for building sentiment classifier od electric vehicle using BERT-based model. The flowchart and project framework of the methodology is shown in figure 3.4 and figure 3.5 below.

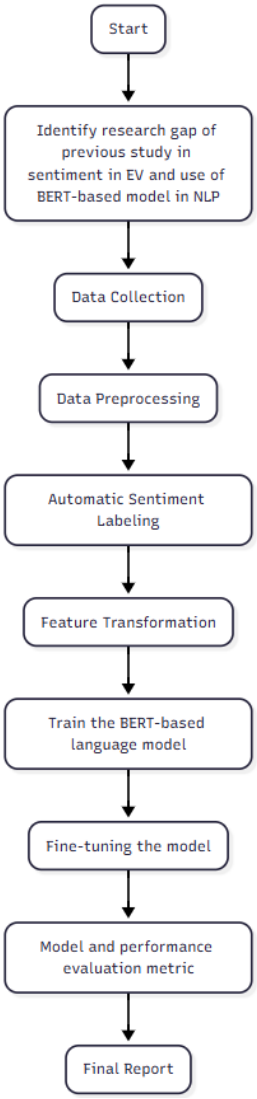


Figure 3.4 Flowchart of Methodology Phases

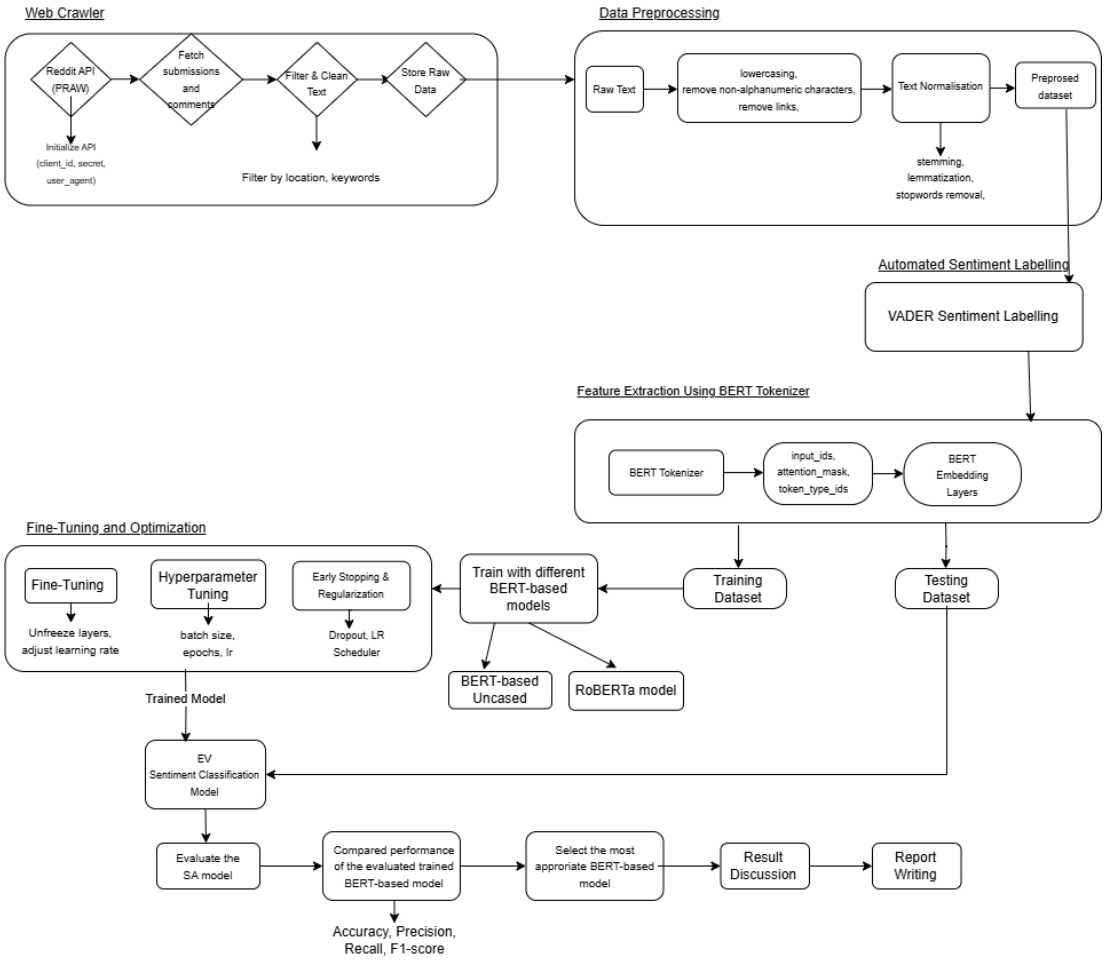


Figure 3.5 Project Framework

CHAPTER 4

INITIAL RESULTS

4.1 Introductions

This section is describing the implementation and experiment part of this research project based on the objective of the proposed research. The code snippet is provided for each of the experiment's implementation, from data collection, data preprocessing, model implementation to model evaluation metric.

4.2 Data Collection

4.2.1 Crawling Method

In this project, primary data collection was done using web crawling method to crawl discourse data on electric vehicle related topic. Web crawling was performed to collect data from the Reddit by using Python Reddit API Wrapper (PRAW) library to interacted with Reddit's API. In the Reddit scraping pipeline, content related to electric vehicles (EVs) from the r/Malaysia subreddit is extracted.

Figure 4.1 shows the code execution for the web scraping process. It loops through a curated list of EV-related search terms and retrieve a maximum of 50 posts per query. For each submission metadata such as 'submission.title', 'submission.selftext', 'submission.score' and 'submission.created_utc' is extracted. Comments of the post are expanded using 'submission.comments.replace_more(limit=None)' to ensures all nested comments thread are fully resolved. Besides, other relevant attributes such as 'comment.body', 'comment.score', and 'comment.total_awards_received' are written to a structured CSV format using Python built in 'csv.writer'.

In addition, there are strategies employed to ensure efficient and robust data collection as showed below.

(a) Exception handling for HTTP 429 (too many requests):

If it exceeds Reddit's rate limits, the API responds with HTTP 429. It captures this through 'ResponseException' and inspect the 'status_code'. If it facing too many requests, it will enter a cooldown period by waiting for 60 seconds before resuming. This is to ensure compliance and avoiding IP bans or account throttling.

(b) General HTTP and Request Exception Handling:

Using 'requests.exceptions.HTTPError' to caught and handle with logging and fallback sleep behavior. For the pipeline resilient to temporary API and network issues.

(c) Deliberate delays between search queries:

This is to avoid triggering Reddit's rate limiter. A 'time.sleep(60)' is performed after processing each search term to ensures that if there are multiple high-volume queries, the script is below Reddit's request thresholds.

(d) No asynchronous processing is used:

The parameter 'check_for_async = False' is set during the initialization of the PRAW Reddit client. This is to suppress warning related to asynchronous event loops, particularly in Python environments that may already use an asynchronous event loop it the background. This enforces synchronous execution of blocking API calls, ensuring a predictable, sequential flow of data extraction and simplifying the debugging and error handling.

```python
# Reddit API credentials
REDDIT_CLIENT_ID = 'y3h5X9SacEDwC-uL8sQFwQ'
REDDIT_CLIENT_SECRET = 'jEpcfFVGCmjm29a5Z-I4045P1kIT5A'
REDDIT_USER_AGENT = 'EVSentimentAnalysisBot by /u/EzNameGG'

# Initialize Reddit API with async warning suppressed
reddit = praw.Reddit(
    client_id=REDDIT_CLIENT_ID,
    client_secret=REDDIT_CLIENT_SECRET,
    user_agent=REDDIT_USER_AGENT,
    check_for_async=False
)

# Search terms
search_terms = [
    'ev', 'electric vehicle', 'electric car', 'tesla', 'charging station',
    'byd', 'ora good cat', 'EV infrastructure', 'EV subsidy', 'ev experience', 'ev issues'
]
subreddit = reddit.subreddit('Malaysia')

# Prepare CSV file
csv_file = 'ev_sentiment_data_new.csv'
with open(csv_file, 'w', newline='', encoding='utf-8') as f:
    writer = csv.writer(f)
    writer.writerow([
        'post_id', 'title', 'selftext', 'post_score', 'upvote_ratio',
        'num_comments', 'post_created_utc',
        'comment_id', 'comment_body', 'comment_score',
        'comment_awards', 'comment_created_utc'
    ])
```

```python
#  Loop through search terms
for i, term in enumerate(search_terms):
    print(f"\n🔍 Searching for: {term}")
    try:
        for submission in subreddit.search(term, limit=100):
            print(f" Processing post: {submission.title[:60]}...")

            try:
                submission.comments.replace_more(limit=None)
            except Exception as e:
                print(f" Error expanding comments: {e}")
                continue

            for comment in submission.comments.list():
                if not isinstance(comment, Comment):
                    continue

                writer.writerow([
                    submission.id,
                    submission.title,
                    submission.selftext,
                    submission.score,
                    submission.upvote_ratio,
                    submission.num_comments,
                    submission.created_utc,
                    comment.id,
                    comment.body.replace('\n', ' ').strip(),
                    comment.score,
                    comment.total_awards_received,
                    comment.created_utc
                ])
```

```python
        except ResponseException as e:
            if hasattr(e, 'response') and e.response.status_code == 429:
                print(" Hit rate limit (HTTP 429). Sleeping for 60 seconds.")
                time.sleep(60)
            else:
                print(f" Unexpected error: {e}")
                continue
        except requests.exceptions.HTTPError as e:
            print(f" HTTP error: {e}. Sleeping 60 seconds.")
            time.sleep(60)
        except Exception as e:
            print(f" Other error: {e}")
            continue

        #  Sleep between terms to avoid search rate limit
        print("⏸ Waiting 60 seconds before next search term...")
        time.sleep(60)

print(f"\n Done! Data saved to {csv_file}")
```

Figure 4.1 Code Snippet of Reddit Web Scraping

4.2.2 System Design and Flow of Web Scraping

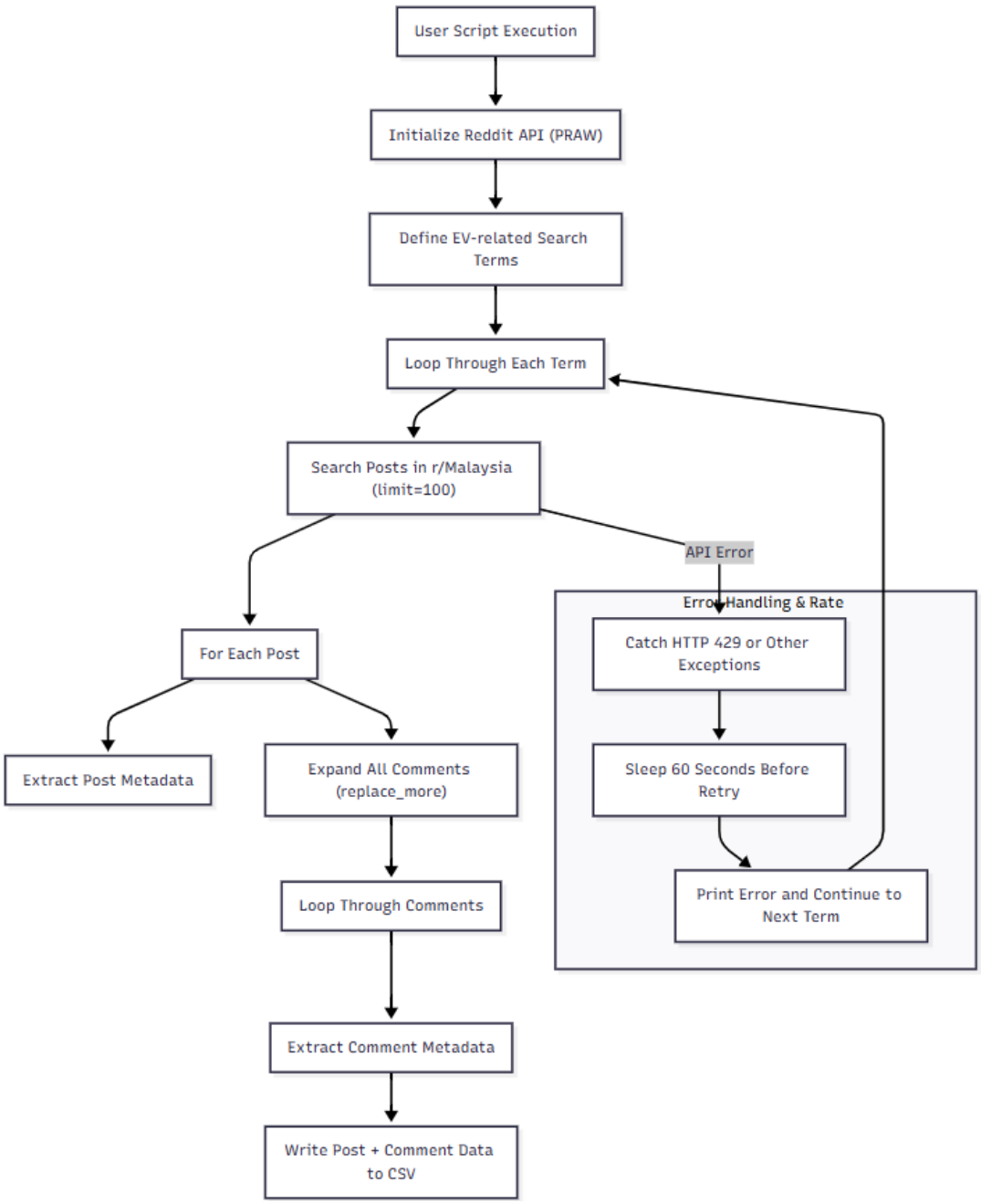


Figure 4.2 Reddit web scraping system design and flow

4.2.3 Number of records collected

The crawler successfully collected 43657 samples with 12 features as shown in Figure 4.4.

```
dir = '/content/drive/MyDrive/Colab_Notebooks/Research_data/ev_sentiment_data_new2.csv'
ev_new_data2 = pd.read_csv(dir)
ev_new_data2
```

Figure 4.3 Snippet of importing the collected data

	post_id	title	selftext	post_score	upvote_ratio	num_comments	post_created_utc	comment_id	comment_body	comment_score	comment_awards	comment_created_utc
0	113tako	What are your thoughts on EV owners using publ...	NaN	785	0.97	365	1.676562e+09	j8se6pf	EV owners when they find an unattended electri...	546	0	1.676566e+09
1	113tako	What are your thoughts on EV owners using publ...	NaN	785	0.97	365	1.676562e+09	j8tvbf7	Apparently some electric cars have battery cap...	211	0	1.676587e+09
2	113tako	What are your thoughts on EV owners using publ...	NaN	785	0.97	365	1.676562e+09	j8uhyyh	Imagine if I set up a bitcoin mining rig here....	127	0	1.676596e+09
3	113tako	What are your thoughts on EV owners using publ...	NaN	785	0.97	365	1.676562e+09	j8s4nfc	That's gonna be one really slow charge. Probab...	383	0	1.676563e+09
4	113tako	What are your thoughts on EV owners using publ...	NaN	785	0.97	365	1.676562e+09	j8sn4cg	It's amazing how many people here feel that it...	253	0	1.676570e+09
...	...	...	...	...	...	...	...	...	...	...	...	...
43652	17pmdxp	New structure to lower taxes for EVs, says Loke	NaN	4	0.83	5	1.699332e+09	k86uwol	Yesssssssssss make EV affordable	3	0	1.699346e+09
43653	17pmdxp	New structure to lower taxes for EVs, says Loke	NaN	4	0.83	5	1.699332e+09	k88l2nx	i just had to spent loads just to buy batterie...	1	0	1.699378e+09
43654	17pmdxp	New structure to lower taxes for EVs, says Loke	NaN	4	0.83	5	1.699332e+09	k8mheji8	I highly doubt it will be affordable for anyon...	1	0	1.699607e+09
43655	17pmdxp	New structure to lower taxes for EVs, says Loke	NaN	4	0.83	5	1.699332e+09	k8b6goz	Hi there, I doubt your blackouts followed by p...	0	0	1.699414e+09
43656	17pmdxp	New structure to lower taxes for EVs, says Loke	NaN	4	0.83	5	1.699332e+09	k8bi4wf	nope, breakers were not tripped and other hous...	1	0	1.699420e+09

43657 rows × 12 columns

Figure 4.4 Snippet of collected data dataset frame

4.3 Data Preprocessing

After web scraping, the raw CSV file is extracted from the google drive data storage to undergoes data preprocessing process. The data preprocessing process is as below.

(a) Data Transformation

Transformed the 'post_created_utc' and 'comment_created_utc' timestamps column to readable Malaysia time (UTC +8) using code as shown in Figure 4.5 to achieve result as shown in Figure 4.6. As the previous 'post_created_utc' and 'comment_created_utc' are in Unix timestamps format.

```
import pytz

# Define Malaysia timezone
malaysia_tz = pytz.timezone('Asia/Kuala_Lumpur')

# Convert and localize timestamps to Malaysia time
ev_data['post_created_utc'] = pd.to_datetime(ev_data['post_created_utc'], unit='s').dt.tz_localize('UTC').dt.tz_convert(malaysia_tz)
ev_data['comment_created_utc'] = pd.to_datetime(ev_data['comment_created_utc'], unit='s').dt.tz_localize('UTC').dt.tz_convert(malaysia_tz)

# Save to Google Drive
cleaned_path = '/content/drive/MyDrive/Colab_Notebooks/Research_data/ev_sentiment_data_cleaned.csv'
ev_data.to_csv(cleaned_path, index=False, encoding='utf-8')

print(f"Cleaned data saved to: {cleaned_path}")
```

Figure 4.5 Code Snippet for data transformation

post_created_utc	comment_created_utc
2023-09-25 10:25:34+08:00	2023-09-26 07:37:41+08:00
2023-09-25 10:25:34+08:00	2023-09-25 10:26:04+08:00
2023-09-25 10:25:34+08:00	2023-09-25 11:02:29+08:00
2025-05-11 18:34:04+08:00	2025-05-11 21:15:19+08:00
2025-05-11 18:34:04+08:00	2025-05-12 06:17:10+08:00

Figure 4.5 Snippet for data transformation

(b) Data Cleaning: Handling Duplicates

Checking of the duplicated rows in the dataset is conducted as some of the posts appeared many times because they were associated with more than one tags or categorizations. Removed the duplicates so that each post title would only appear as a single post to avoid redundant data.

Figure 4.6 shows the code snippet for checking the duplicated data. It shows in Figure 4.7 that's there is 440 duplicated data. Then, 'cleaned_ev_data = cleaned_ev_data.drop_duplicates()' is used to remove the duplicated data as shown in figure 4.7. After removed duplicated data, only 43217 samples left.

```
# check duplicate
duplicate_rows = cleaned_ev_data.duplicated().sum()
print(f"Number of fully duplicated rows: {duplicate_rows}")

# Preview some duplicate rows
if duplicate_rows > 0:
    print("\nSample duplicate rows:")
    display(cleaned_ev_data[cleaned_ev_data.duplicated()].head())
```

Figure 4.6 Code Snippet checking duplicates

```
Number of fully duplicated rows: 440

Sample duplicate rows:
```

	post_id	title	selftext	post_score	upvote_ratio	num_comments	post_created_utc	comment_id	comment_body	comment_score	comment_awards	comment_created_utc
3048	1b6u6ch	Electric Vehicle ownership in Malaysia have be...	NaN	30	0.92	42	2024-03-05 10:42:17+08:00	kdj1bzn	While EVs are necessary for a step towards gre...	5	0	2024-03-06 07:20:27+08:00
3052	1b6u6ch	Electric Vehicle ownership in Malaysia have be...	NaN	30	0.92	42	2024-03-05 10:42:17+08:00	ktguk5a	When fuel is cheaper than Electricity in this ...	2	0	2024-03-06 00:09:17+08:00
3056	1b6u6ch	Electric Vehicle ownership in Malaysia have be...	NaN	30	0.92	42	2024-03-05 10:42:17+08:00	ktf2a5c	Tapi tesla bro, auto pilot bro, bukak pintu mc...	5	0	2024-03-05 14:37:24+08:00
3060	1b6u6ch	Electric Vehicle ownership in Malaysia have be...	NaN	30	0.92	42	2024-03-05 10:42:17+08:00	ktfin0n	Hydrogen fueled cars are a technological dead ...	10	0	2024-03-05 17:53:06+08:00
3061	1b6u6ch	Electric Vehicle ownership in Malaysia have be...	NaN	30	0.92	42	2024-03-05 10:42:17+08:00	ktf0zga	Kinda is, because it fast to refill unlike ev ...	1	0	2024-03-05 14:23:51+08:00

Figure 4.7 Snippet for duplicated data found

```
cleaned_ev_data = cleaned_ev_data.drop_duplicates()
cleaned_ev_data
```

Figure 4.8 Code Snippet for remove duplicate

43217 rows × 12 columns

Figure 4.9 Snippet of data after removed duplicate

(c) Data Cleaning: Handling Missing data

Before performing methods for handling missing values or text data in the columns, an inspection of missing values in which column is conducted first using 'isnull().sum()' as show in Figure 4.10.

```
cleaned_ev_data.isnull().sum()
```

Figure 4.10 Code Snippet for checking missing values

Based on Figure 4.11, since only the column 'selftext' has missing values and this project is planning doing sentiment analysis on comments, this column is non-essential. Hence, no missing value handling is needed.

	0
post_id	0
title	0
selftext	11568
post_score	0
upvote_ratio	0
num_comments	0
post_created_utc	0
comment_id	0
comment_body	0
comment_score	0
comment_awards	0
comment_created_utc	0

Figure 4.11 Snippet of Inspect Missing Values

(d) Filter Non Ev Related posts

In this subsection is to filter out scrap post that not related to electric vehicle. The post is scrap by defining the ev related keyword to see whether post title is related to ev or not. There are total of 4708 rows of data that post title is not related to ev.

```
ev_keywords = [
    'ev', 'electric vehicle', 'electric car', 'charging station', 'tesla', 'byd',
    'ora good cat', 'mg4', 'battery capacity', 'range anxiety', 'charge point',
    'hybrid', 'plug-in hybrid', 'neta v', 'volvo ev', 'perodua ev',
    'ev infrastructure', 'ev subsidy', 'ev charging', 'renewable energy',
    'ev experience', 'ev issues'
]
```

```
def contains_ev_keywords(text):
    text = text.lower()
    return any(keyword in text for keyword in ev_keywords)

# Apply to title and selftext separately
title_match = cleaned_ev_data['title'].apply(contains_ev_keywords)
selftext_match = cleaned_ev_data['selftext'].apply(contains_ev_keywords)

# Filter non-EV-related posts
non_ev_posts = cleaned_ev_data[(~title_match) & (~selftext_match)]

# Print total number of non-EV-related posts
print(f"Total number of non-EV-related posts: {len(non_ev_posts)}")

# View non-EV-related posts
display(non_ev_posts[['title', 'selftext']].iloc[500:])
```

Figure 4.12 Code Snippet of Filtering the Non Ev Post

```
Total number of non-EV-related posts: 5208
```

	title	selftext
6345	Like toys car ads	This article looks like toys car ads, battery ...
6346	Like toys car ads	This article looks like toys car ads, battery ...
6347	Like toys car ads	This article looks like toys car ads, battery ...
6348	Like toys car ads	This article looks like toys car ads, battery ...
6349	Like toys car ads	This article looks like toys car ads, battery ...
...	...	...
40812	No wonder habit like this, thats why when coop...	
40813	No wonder habit like this, thats why when coop...	
40814	No wonder habit like this, thats why when coop...	
40815	No wonder habit like this, thats why when coop...	
40816	No wonder habit like this, thats why when coop...	

4708 rows × 2 columns

Figure 4.13 Snippet of Inspect Missing Values

(e) Data Cleaning: Remove Text Noise

Focus on cleaning the comments stores in the ‘title’, ‘selftext’ and ‘comment_body’ column only, as this is the text that will be used for sentiment analysis implementation. Before data cleaning the unnecessary text noise in the comments is inspect first before removing it as shown in Figure 4.14. To view

whether there is non-alphabetical character, containing digits, containing multiple spaces and URLs. Figure 4.15 shows parts of the unnecessary contains found in the comment texts. This unnecessary text noise need to be remove to reduce the dimension of the data.

```
import re

def inspect_text_noise(text):

  full_text = ' '.join(text.astype(str))

  non_alpha = sorted(set(re.findall(r'[^a-zA-Z\s]', full_text)))
  has_digits = any(char.isdigit() for char in full_text)
  has_multiple_spaces = bool(re.search(r'\s{2,}', full_text))
  urls = re.findall(r'http[s]?://\S+', full_text)


  print("Non-alphabetic characters:", non_alpha)
  print("Contains digits:", has_digits)
  print("Contains multiple spaces:", has_multiple_spaces)
  print("Sample URLs found:", urls[:5])


inspect_text_noise(cleaned_ev_data['comment_body'])
```

Figure 4.14 Code Snippet of Inspect Text Noise

```
Non-alphabetic characters: ['\x05', '!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', '0', '1', '2', '3', '4', '5',
Contains digits: True
Contains multiple spaces: True
Sample URLs found: ['https://preview.redd.it/lin8sp7ogmia1.jpeg?width=256&format=pjpg&auto=webp&s=82949a8b74c6197d07c63a9d644fb0b51bffd7f2',
```

Figure 4.15 Snippet of Founded Text Noise

Convergent of emoji to text is applied in this data cleaning step as the emoji might contain sufficient information for the sentiment. It would help in sentiment classification.

```
# Function to convert emojis to text
def convert_emojis(text):
    if isinstance(text, str):
        return emoji.demojize(text, language='en')
    return text

# Apply to relevant text columns
cleaned_ev_data.loc[:, 'title'] = cleaned_ev_data['title'].apply(convert_emojis)
cleaned_ev_data.loc[:, 'selftext'] = cleaned_ev_data['selftext'].apply(convert_emojis)
cleaned_ev_data.loc[:, 'comment_body'] = cleaned_ev_data['comment_body'].apply(convert_emojis)
```

Figure 4.16 Code Snippet of Converting Emoji to Text

The steps used to clean the data as shown in Figure 4.17 and 4.18 are removing the URL, hashtags, digit, keeping only English letter, remove multiple space, remove space at beginning and end of string, remove stretchy words and lowercase the text. Figure 4.19 shows the snippet of the text does not containing any text noise after text cleaning.

```
def clean_text(text):

  text = re.sub(r'https?://\S+|www\.\S+', '', text) #remove url
  text = re.sub(r'@\w+', '', text) # remove mentions
  text = re.sub(r'#\w+', '', text) # remove hashtags
  text = re.sub(r'\d+', '', text) # remove digit
  text = re.sub(r'[^a-zA-Z\s]', '', text) # keep only english letter
  text = re.sub(r'\s+', ' ', text) #remove multiple spaces
  text = text.strip() #remove space at beginning and end of string
  #text = text.lower()

  return text

cleaned_ev_data = cleaned_ev_data.copy()

cleaned_ev_data['cleaned_title'] = cleaned_ev_data['title'].astype(str).apply(clean_text)
cleaned_ev_data['cleaned_selftext'] = cleaned_ev_data['selftext'].astype(str).apply(clean_text)
cleaned_ev_data['cleaned_comment'] = cleaned_ev_data['comment_body'].astype(str).apply(clean_text)
```

Figure 4.17 Code Snippet for Text Cleaning

```
def reduce_stretchy_words(text):
    return re.sub(r'(.)\1{2,}', r'\1\1', text)  # "yesssssss" -> "yess"

cleaned_ev_data['cleaned_comment'] = cleaned_ev_data['cleaned_comment'].apply(reduce_stretchy_words)
```

Figure 4.18 Code Snippet for Remove stretchy word

```
Non-alphabetic characters: []
Contains digits: False
Sample URLs found: []
```

Figure 4.19 Snippet for After Text Cleaning

(f) Data Transformation: Data Normalization.

Next is data normalization to convert all the text into standardized format to reduce complexity and variability for text classification tasks later. Text Normalization such as removing stop word and word lemmatization is conducted as shown in Figure 4.20. the output of normalized text is shown in

Figure 4.21 in new ‘normalized_title’, ‘normalized_selftext’ and ‘normalized_comment’ column.

```
def normalize_text(text):
    if not isinstance(text, str):
        return ""

    words = text.split()
    result = [
        lemmatizer.lemmatize(word) for word in words
        if word.lower() not in custom_stopwords
    ]
    return ' '.join(result)

cleaned_ev_data['normalized_title'] = cleaned_ev_data['cleaned_title'].apply(normalize_text)
cleaned_ev_data['normalized_selftext'] = cleaned_ev_data['cleaned_selftext'].apply(normalize_text)
cleaned_ev_data['normalized_comment'] = cleaned_ev_data['cleaned_comment'].apply(normalize_text)
```

Figure 4.20 Code Snippet for Text Normalisation

normalized_title	normalized_selftext	normalized_comment
thought EV owner using public electrical point...	NaN	NaN
Say hello EV plat number	Say hello new Malaysias EV number plate Adapte...	NaN
Say hello EV plat number	Say hello new Malaysias EV number plate Adapte...	NaN
Proton reveals EV subbrand eMAS	NaN	NaN
Proton reveals EV subbrand eMAS	NaN	NaN

Figure 4.21 Snippet of Normalized Text

(g) Data Recleaning

Data cleaning processed is repeatead again to remove the NaN row in all three columns of ‘normalized_title’, ‘normalized_selftext’ and ‘normalized_comment’. As there were missing NAN values after the performance of text preprocessing before head as shown in Figure 4.22. There is total of 239 row contain NAN values. For the normalized column that is NAN values, the content of this column is filled with the combination of ‘normalized_title’ and ‘normalized_selftext’ if both column contains the text. Whereas if either one of the column does not contain the text then only filled ‘normalized_comment’ with either one of the column that is available. This

approach is taken as the ‘normalized_title’ and ‘normalized_selftext’ yet containing information that can be filled for the ‘normalized_comment’ column for the sentiment analysis later.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38009 entries, 0 to 38008
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   post_id              38009 non-null  object
 1   title                38009 non-null  object
 2   selftext             30124 non-null  object
 3   post_score           38009 non-null  int64
 4   upvote_ratio         38009 non-null  float64
 5   num_comments         38009 non-null  int64
 6   post_created_utc     38009 non-null  object
 7   comment_id           38009 non-null  object
 8   comment_body         38009 non-null  object
 9   comment_score        38009 non-null  int64
 10  comment_awards       38009 non-null  int64
 11  comment_created_utc  38009 non-null  object
 12  cleaned_title        38009 non-null  object
 13  cleaned_selftext     30124 non-null  object
 14  cleaned_comment      37869 non-null  object
 15  normalized_title     38009 non-null  object
 16  normalized_selftext  30124 non-null  object
 17  normalized_comment   37770 non-null  object
dtypes: float64(1), int64(4), object(13)
memory usage: 5.2+ MB
```

Figure 4.22 Snippet of Normalized Text

```
recleaned_ev_data.loc[
    recleaned_ev_data['normalized_comment'].isnull() | (recleaned_ev_data['normalized_comment'].str.strip() == ''),
    ['normalized_title',  'normalized_selftext','comment_body','normalized_comment']
]
```

	normalized_title	normalized_selftext	comment_body	normalized_comment
128	thought EV owner using public electrical point...	NaN	https://preview.redd.it/z9lyghrrmria1.png?widt...	NaN
508	Say hello EV plat number	Say hello new Malaysias EV number plate Adapte...	https://en.m.wikipedia.org/wiki/International_...	NaN
531	Say hello EV plat number	Say hello new Malaysias EV number plate Adapte...	https://en.m.wikipedia.org/wiki/International_...	NaN
695	Proton reveals EV subbrand eMAS	NaN	https://preview.redd.it/7jaf464lkb8d1.png?widt...	NaN
696	Proton reveals EV subbrand eMAS	NaN	https://preview.redd.it/o1cpifd7kb8d1.png?widt...	NaN
...	...	...	...	...
36785	rmalaysia daily random discussion quick questi...	rmalaysias official daily random discussion qu...	):	NaN
36911	Malaysias EV sale stuck slow lane amid high co...	Prime Minister Anwar Ibrahim want per cent veh...	https://preview.redd.it/6m0hq8g8ow1d1.jpeg?wid...	NaN
37370	Malaysias EV sale stuck slow lane amid high co...	Prime Minister Anwar Ibrahim want per cent veh...	https://preview.redd.it/6m0hq8g8ow1d1.jpeg?wid...	NaN
37601	personal opinion year budget	something wish write regarding budget presente...	How?	NaN
37873	personal opinion year budget	something wish write regarding budget presente...	How?	NaN

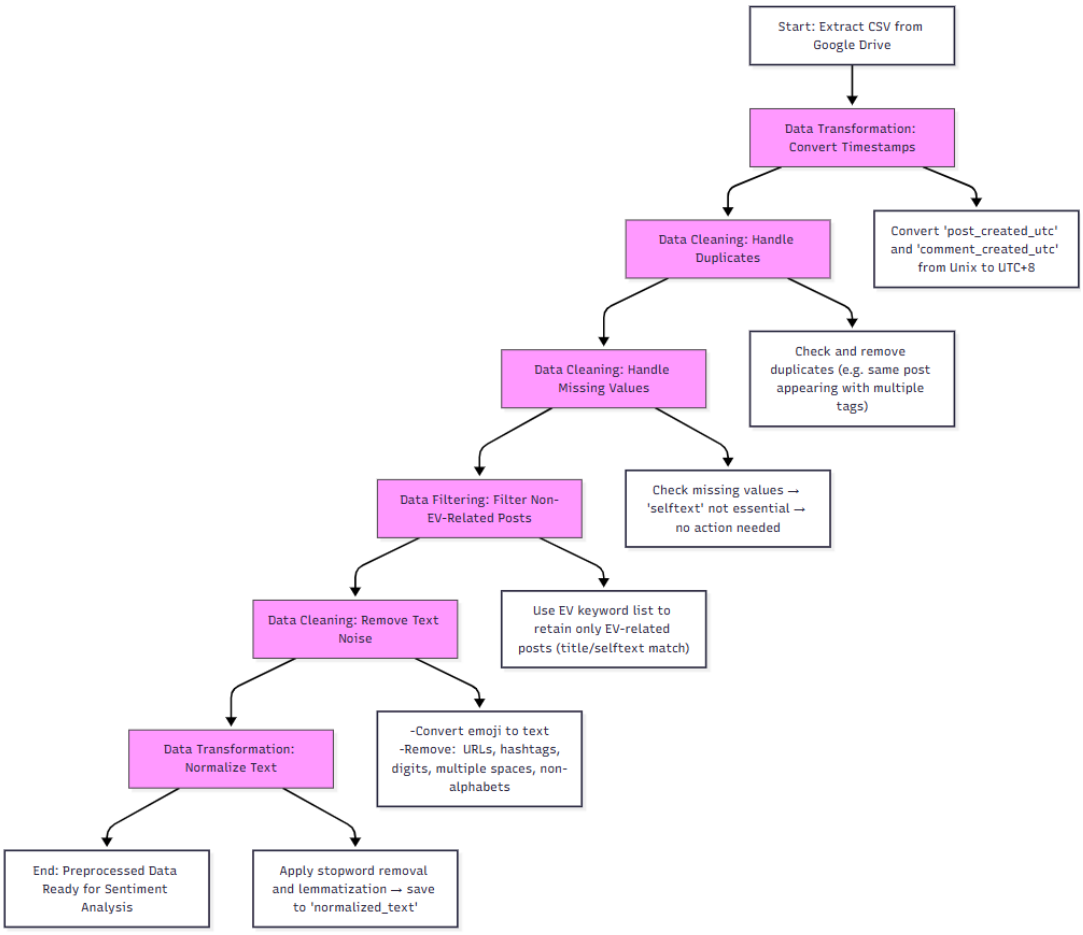
239 rows × 4 columns

Figure 4.23 Snippet of NAN in 3 Column

```python
import numpy as np

def fill_comment(row):
    title = row['normalized_title']
    selftext = row['normalized_selftext']
    comment = row['normalized_comment']

    if pd.notnull(comment) and comment.strip() != '':
        return comment  # Keep existing comment

    if pd.notnull(title) and title.strip() != '' and pd.notnull(selftext) and selftext.strip() != '':
        return title.strip() + " " + selftext.strip()

    if pd.notnull(title) and title.strip() != '':
        return title.strip()

    if pd.notnull(selftext) and selftext.strip() != '':
        return selftext.strip()

    return np.nan  # If all are missing

# Apply the function to fill normalized_comment
recleaned_ev_data['normalized_comment'] = recleaned_ev_data.apply(fill_comment, axis=1)

# Drop rows where normalized_comment is still missing or empty
recleaned_ev_data = recleaned_ev_data[
    recleaned_ev_data['normalized_comment'].notnull() &
    (recleaned_ev_data['normalized_comment'].str.strip() != '')
]

# Reset index (optional)
recleaned_ev_data.reset_index(drop=True, inplace=True)
```

Figure 4.24 Code Snippet too fill NAN Normalized Comment Column

4.3.1 Data Preprocessing Flowchart



4.4 Exploratory Data Analysis (EDA)

After data preprocessing, an EDA is conducted to view the structured of the data in order to have overview and understanding of the underlying data. There is 38009 rows of data sample left after the data preprocessing is implemented as show in Figure 4.25.

Figure 4.26 shows that many the mean post score is higher than median, indicating a right skewed distribution. Many posts have very low post score with minimum value of 0 while a few posts have extreme high score of maximum 1475. The mean of comment is higher than median; this shows a right skewed distribution. Some posts have very few comments with a minimum of 1 while other posts had a maximum of 800 comments. There is significant variability in the number of

comments per post. The minimum score of -58 shows that there is some comments received significant downvotes, while the maximum score of 863 indicates highly upvoted comments. Hence can be conclude that most posts have relatively low scores and few comments, but a small number of post are highly engaged.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38009 entries, 0 to 38008
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   post_id              38009 non-null  object
 1   post_score           38009 non-null  int64
 2   upvote_ratio         38009 non-null  float64
 3   num_comments         38009 non-null  int64
 4   post_created_utc     38009 non-null  object
 5   comment_id           38009 non-null  object
 6   comment_score        38009 non-null  int64
 7   comment_awards       38009 non-null  int64
 8   comment_created_utc  38009 non-null  object
 9   normalized_title     38009 non-null  object
 10  normalized_selftext  30124 non-null  object
 11  normalized_comment   38009 non-null  object
dtypes: float64(1), int64(4), object(7)
memory usage: 3.5+ MB
None
```

Figure 4.25 Snippet of Statistical data description

	post_score	upvote_ratio	num_comments	comment_score	comment_awards
count	38009.000000	38009.000000	38009.000000	38009.000000	38009.0
mean	120.076771	0.882285	304.430345	4.182878	0.0
std	270.057470	0.101636	197.373820	14.375248	0.0
min	0.000000	0.330000	1.000000	-58.000000	0.0
25%	8.000000	0.830000	104.000000	1.000000	0.0
50%	14.000000	0.910000	319.000000	2.000000	0.0
75%	69.000000	0.950000	454.000000	4.000000	0.0
max	1475.000000	1.000000	800.000000	863.000000	0.0

Figure 4.26 Snippet of Statistical data description

Figure 4.27 shows the frequency of posts by year. There is rapid growth of comments from 2018 to 2021, and a subsequent decline from 2022 to 2025.

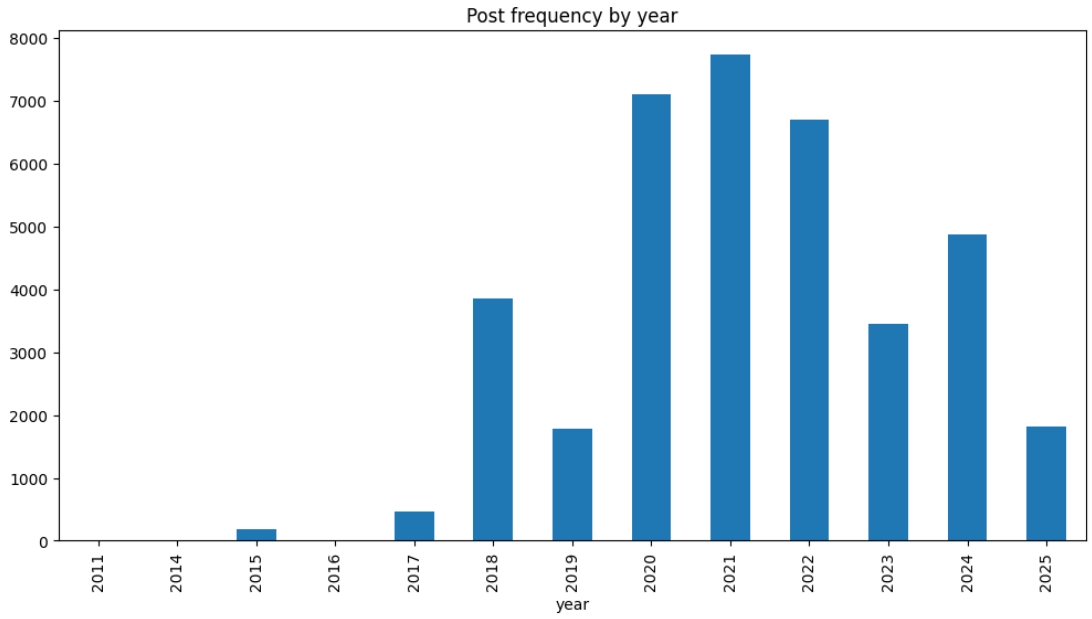


Figure 4.27 Snippet of Post Frequency by Years

Figure 4.28, 4.29, 4.30 shows that most comment are highly skewed. As most of the comment being very short and very few exceeding 200 characters. The delected comment impact the distribution. The distribution is heavily left skewed, with a long tail on the right representing rare occurrence of long comments.

```
 Comment Length Stats:
count       37770
unique      32427
top       deleted
freq         1208
Name: normalized_comment, dtype: object
```

Figure 4.28 Comments Length Statistic

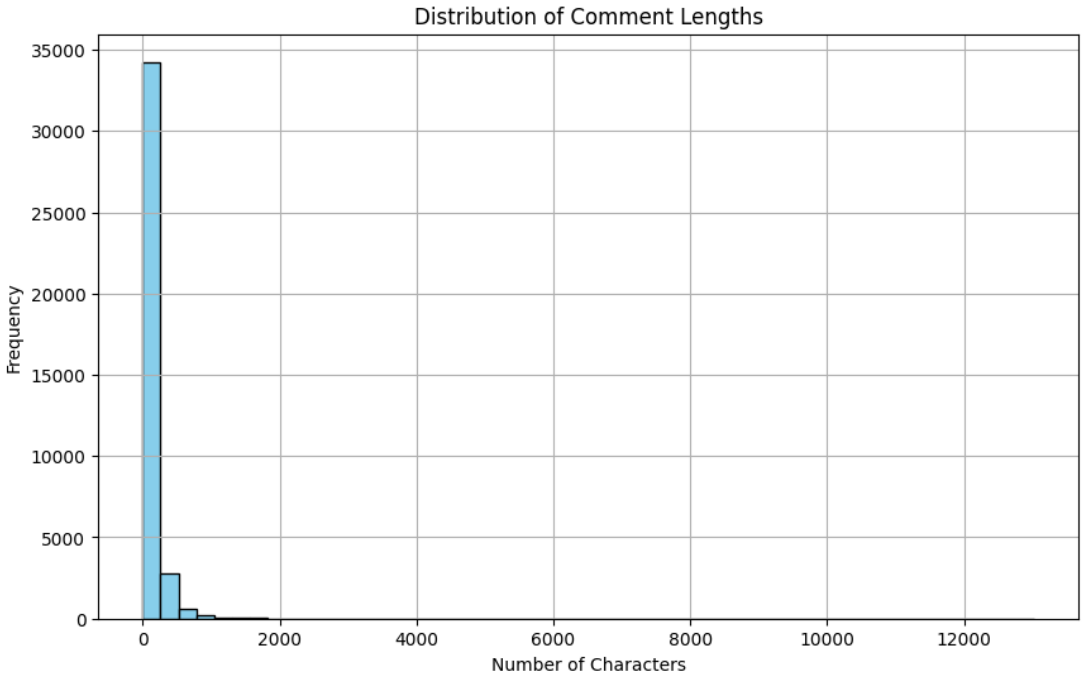


Figure 4.29 Distribution of Comment Lengths

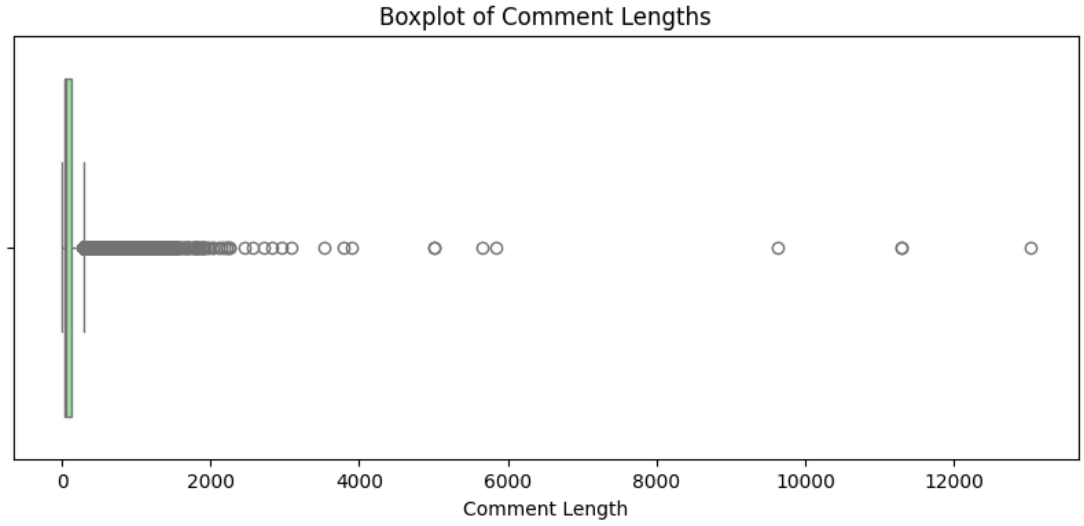


Figure 4.30 Boxplot of Comment Lengths

Both visualizations from Figure 4.31 and Figure 4.32 confirm that the primary topic of discussion is electric vehicles. Positive sentiments are reflected by words like good, like. Negative sentiments or challenges are indicated by words such as problem, issue, expensive, bad, hate and hard. The top word and word cloud shows that the discussions likely revolve around the strong focus on electric vehicle, pricings, user opinion, and challenges with a geographic context related to Malaysia.

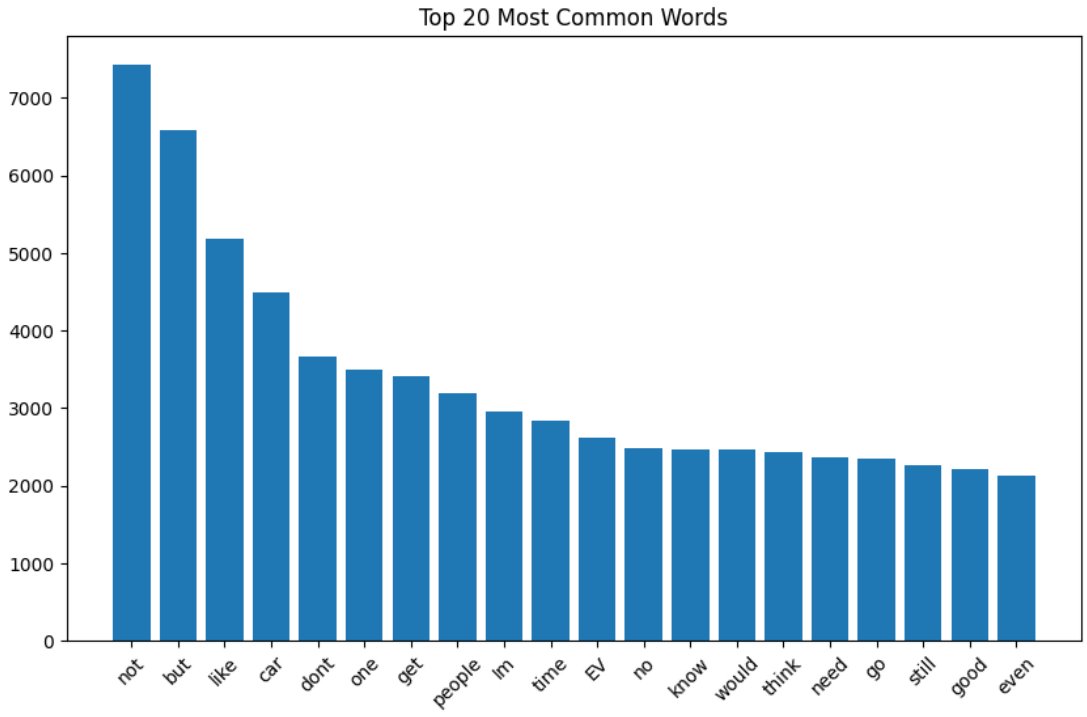


Figure 4.31 Bar Chart of Top 20 Words

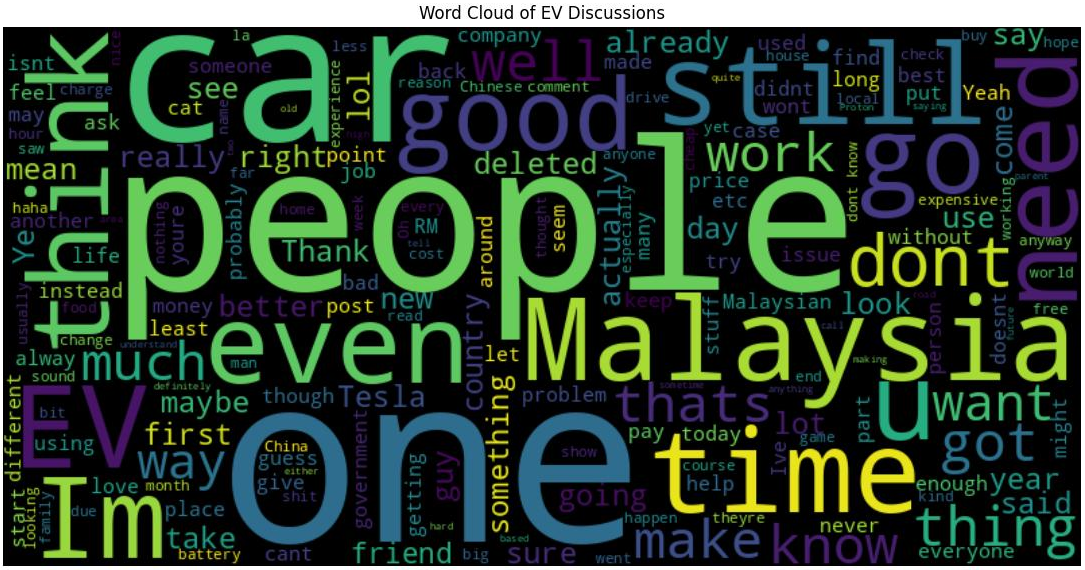


Figure 4.32 Word Cloud of the Text

The correlation matrix heatmap show a strong relationship between words like ‘car’, ‘ev’. Highlighting electric vehicle as a central theme in the discussion. While word such as don’t and like indicate frequent expression od opinion, suggesting mixed or negative sentiment around the topics.

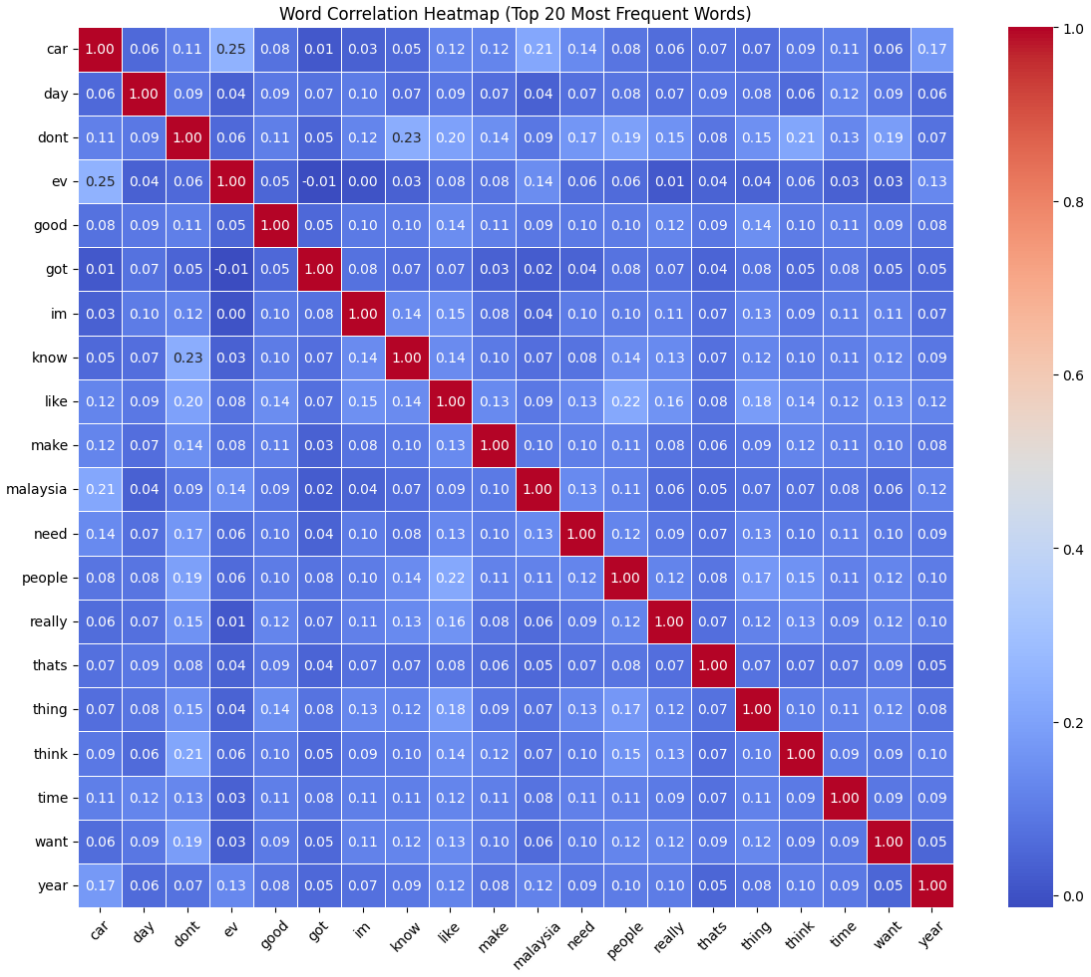


Figure 4.33 Bar Chart of Top 20 Words

4.5 Sentiment Labelling

Valence Aware Dictionary and Sentiment Reasoner (Vader) is used for the automatic sentiment labelling for the normalized text comment data before model training. Vader used predefined sentiment lexicon which is a list of English words and expression had a valance score to indicated how positive or negative is the particular word. Figure 4.34 shows the code implementation for VADER labeling in this section using vader = SentimentIntensityAnalyzer(). Each row in the 'normalized_comment' column is classified based on the compound score computed by the VADER. A compound score of more than and equal to 0.05 is positive sentiment, while a score of

less than and equal to -0.05 is negative sentiment and score between it is neutral sentiment.

```
vader = SentimentIntensityAnalyzer()


def vader_label(text):
    score = vader.polarity_scores(str(text))['compound']
    if score >= 0.05:
        return 'positive'
    elif score <= -0.05:
        return 'negative'
    else:
        return 'neutral'

labelled_data['vader_sentiment'] = labelled_data['normalized_comment'].apply(vader_label)
```

Figure 4.34 Code Snippet of Vader Sentiment Labelling

Figure 4.35 and Figure 4.36 shows the results of automatic labelling using VADER. The auto sentiment labelling shows that there is 17184 positive comments, 9350 negative comments and 11475 neutral comments. This shows an overall higher positive sentiment labelled comment text.

	normalized_comment	vader_sentiment
0	EV owner find unattended electrical outlet	neutral
1	Apparently electric car battery capacity kWh p...	negative
2	Imagine set bitcoin mining rig sudden tapping ...	negative
3	Thats gonna one really slow charge Probably ta...	neutral
4	amazing many people feel totally ok charge car...	positive
...	...	...
38004	Yess make EV affordable	neutral
38005	spent load buy battery UPS overloadings due ex...	negative
38006	highly doubt affordable anyone outside not buy...	negative
38007	Hi doubt blackout followed power surge arent f...	neutral
38008	nope breaker not tripped house experiencing issue	neutral

38009 rows × 2 columns

Figure 4.35 Output of Sentiment labelled for Comment Text

vader_sentiment	count
positive	17184
neutral	11475
negative	9350

dtype: int64

Figure 4.36 Snippet for Sentiment Distribution

4.5.1 Train-test Split

Figure 4.37 shows the code in preparing before the model training. By encoded the sentiment label and then performed train test split. Neutral is encoded to 0, negative is encoded to 1 while positive is encoded to 2. Then it used StratifiedShuffleSplit() to split the dataset into training and testing sets while ensure the original proportion of sentiment label is equally distributed. It split the data 80% for training and 20% for testing. Figure 4.38 shows there is 30407 training data and 7602 testing data.

```
# Prepare and Encode Labels
labelled_data = labelled_data[labelled_data['vader_sentiment'].notna()].copy()
labelled_data['label'] = labelled_data['vader_sentiment'].map({'Negative': 0, 'Neutral': 1, 'Positive': 2})
labelled_data = labelled_data[labelled_data['normalized_comment'].notna() & (labelled_data['normalized_comment'].str.strip() != '')].copy()

# Stratified Train-Test Split
splitter = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_idx, test_idx in splitter.split(labelled_data['normalized_comment'], labelled_data['label']):
    train_df = labelled_data.iloc[train_idx]
    test_df = labelled_data.iloc[test_idx]
```

Figure 4.37 Code Snippet for Splitting and encoding the dataset

```
print("Train shape:", train_df.shape)
print("Test shape:", test_df.shape)
```

```
Train shape: (30407, 3)
Test shape: (7602, 3)
```

Figure 4.38 Code Snippet for Splitting and encoding the dataset

4.6 Features Transformation

Features transformation is implemented by converting raw text into numerical inputs of input_ids and attention masks using the BERT tokenizer. This step is important to turning human readable text into format that a model can process. Then 'Dataset.from_pandas()' wraps training and testing data frame into Hugging face type 'dataset.Dataset' format. This is to helping mapping and formatting the operation efficiently. Then '.map(tokenizer)' is implemented to transform text into token IDS and attention mask. The cast_column("label", Value("int32")) ensure all target variable is in the correct integer format for model training. Followed up by DataCollatorWithPadding() which ensure all sequence in a batch have the same length by dynamically padding it. Lastly, to_tf_dataset() finalized the data pipeline by converting everything into TensorFlow compatible dataset for training and testing.

```
import pyarrow as pa
from datasets import Value

# Tokenization
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

def tokenize(example):
    return tokenizer(example["normalized_comment"], truncation=True, max_length=256)

train_dataset = Dataset.from_pandas(train_df[["normalized_comment", "label"]])
test_dataset = Dataset.from_pandas(test_df[["normalized_comment", "label"]])

train_dataset = train_dataset.map(tokenize)
test_dataset = test_dataset.map(tokenize)

train_dataset = train_dataset.cast_column("label", Value("int32"))
test_dataset = test_dataset.cast_column("label", Value("int32"))

data_collator = DataCollatorWithPadding(tokenizer=tokenizer, return_tensors="tf")
```

```
# Convert to TensorFlow Dataset
tf_train_dataset = train_dataset.to_tf_dataset(
    columns=["input_ids", "attention_mask"],
    label_cols="label",
    shuffle=True,
    batch_size=8,
    collate_fn=data_collator,
)

tf_eval_dataset = test_dataset.to_tf_dataset(
    columns=["input_ids", "attention_mask"],
    label_cols="label",
    shuffle=False,
    batch_size=8,
    collate_fn=data_collator,
)
```

Figure 4.39 Code Snippet for Splitting and encoding the dataset

4.7 Model Training

In this subsection, 2 model will be trained. Which are BERT-based uncased and RoBERTa. The training process of the BERT-based uncased is shown in figure 4.40. The model is initialized from 'TFBertForSequenceClassification', which is a model pretrained on bert-based uncased for 3-class sentiment classification task. An optimizer is created using 'created_optimizer' which included a learning rate schedular with linear decay.The model is compiled with AdamW optimizer using sparse categorical cross entropy function and accuracy as the evaluation metric. To monitor the performance during the training, 2 callback are configured. Setting a model checkpoint to save each epoch and a earlystopping is implemented to prevent overfitting by halting training if validation accuracy does not improve for two consecutive epoch. Then model is then trained on the prepared TensorFlow dataset on 5 epoch. In figure 4.40 shows the training of RoBERTa which is also the similar process as bert-based uncased.

The training result of bert-based uncased dhow in figure 4.41 shows that the training accuracy improve steadily from 83.88% in epoch 1 to 98.91% in epoch 5. This indicate the model learn well from the data. Although validation ccuradcy slightly fluctuated, it remained high across all epoch, showing good generalization without severe overfitting. Whereas for the RoBERTa model, the model begin with a training accuracy of 76.18% in epoch 1 and reached 97.17% by epoch 3. This indicating a strong learning progress. The validation loss decreased consistently, through it is slightly increased in the final epoch. This shows that the model might be approaching its optimal point.

```python
import tensorflow as tf

# Load BERT Model

# Optimizer and Callbacks (Checkpointing & Early Stopping)
batch_size = 8
epochs = 5
steps_per_epoch = len(tf_train_dataset)

optimizer, schedule = create_optimizer(
    init_lr=2e-5,
    num_train_steps=steps_per_epoch * epochs,
    num_warmup_steps=0
)

model.compile(optimizer=optimizer,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

# Directory for saving all checkpoints
checkpoint_dir = "/content/drive/MyDrive/Modified_Templates/bert_checkpoints"
os.makedirs(checkpoint_dir, exist_ok=True)

checkpoint_cb = tf.keras.callbacks.ModelCheckpoint(
    filepath=os.path.join(checkpoint_dir, "bert_epoch_{epoch:02d}_valacc_{val_accuracy:.4f}.keras"),
    save_weights_only=False,
    save_best_only=False,
    monitor='val_accuracy',
    mode='max',
    verbose=1
)

earlystop_cb = tf.keras.callbacks.EarlyStopping(
    monitor='val_accuracy',
    patience=2,
    restore_best_weights=True,
    verbose=1
)
```

```python
# 9. Train the Model
history = model.fit(
    tf_train_dataset,
    validation_data=tf_eval_dataset,
    epochs=epochs,
    callbacks=[checkpoint_cb, earlystop_cb]
)
```

Figure 4.40 Code Snippet for Set up of BERT-based uncased training

```
Epoch 1/5
3801/3801 [==============================] - ETA: 0s - loss: 0.4273 - accuracy: 0.8388
Epoch 1: saving model to /content/drive/MyDrive/Colab_Notebooks/Research_data/bert_checkpoints/bert_epoch_01_valacc_0.9132.keras
/usr/local/lib/python3.11/dist-packages/transformers/generation/tf_utils.py:465: UserWarning: `seed_generator` is deprecated and will be removed in a future version.
  warnings.warn("`seed_generator` is deprecated and will be removed in a future version.", UserWarning)
3801/3801 [==============================] - 709s 176ms/step - loss: 0.4273 - accuracy: 0.8388 - val_loss: 0.2632 - val_accuracy: 0.9132
Epoch 2/5
3801/3801 [==============================] - ETA: 0s - loss: 0.1931 - accuracy: 0.9371
Epoch 2: saving model to /content/drive/MyDrive/Colab_Notebooks/Research_data/bert_checkpoints/bert_epoch_02_valacc_0.9259.keras
3801/3801 [==============================] - 642s 169ms/step - loss: 0.1931 - accuracy: 0.9371 - val_loss: 0.2380 - val_accuracy: 0.9259
Epoch 3/5
3801/3801 [==============================] - ETA: 0s - loss: 0.1089 - accuracy: 0.9663
Epoch 3: saving model to /content/drive/MyDrive/Colab_Notebooks/Research_data/bert_checkpoints/bert_epoch_03_valacc_0.9282.keras
3801/3801 [==============================] - 643s 169ms/step - loss: 0.1089 - accuracy: 0.9663 - val_loss: 0.2472 - val_accuracy: 0.9282
Epoch 4/5
3801/3801 [==============================] - ETA: 0s - loss: 0.0624 - accuracy: 0.9802
Epoch 4: saving model to /content/drive/MyDrive/Colab_Notebooks/Research_data/bert_checkpoints/bert_epoch_04_valacc_0.9238.keras
3801/3801 [==============================] - 644s 169ms/step - loss: 0.0624 - accuracy: 0.9802 - val_loss: 0.2920 - val_accuracy: 0.9238
Epoch 5/5
3801/3801 [==============================] - ETA: 0s - loss: 0.0378 - accuracy: 0.9891
Epoch 5: saving model to /content/drive/MyDrive/Colab_Notebooks/Research_data/bert_checkpoints/bert_epoch_05_valacc_0.9311.keras
3801/3801 [==============================] - 645s 170ms/step - loss: 0.0378 - accuracy: 0.9891 - val_loss: 0.2830 - val_accuracy: 0.9311
Restoring model weights from the end of the best epoch: 5.
```

Figure 4.41 Snippet for training result of BERT-based uncased model

```python
import os
from transformers import (
    TFRobertaForSequenceClassification,
    RobertaTokenizer,
    create_optimizer
)
import tensorflow as tf

# Prepare optimizer and learning rate schedule
epochs = 5
steps_per_epoch = len(tf_train_dataset)

optimizer, schedule = create_optimizer(
    init_lr=2e-5,
    num_train_steps=steps_per_epoch * epochs,
    num_warmup_steps=int(0.1 * steps_per_epoch * epochs)  # optional warmup
)

# Compile model
roberta_model.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy']
)
```

```python
# Prepare callbacks

# Checkpoint weights only
checkpoint_dir = "/content/drive/MyDrive/Modified_Templates/roberta_checkpoints"
os.makedirs(checkpoint_dir, exist_ok=True)

checkpoint_cb = tf.keras.callbacks.ModelCheckpoint(
    filepath=os.path.join(checkpoint_dir, "best_weights.h5"),
    save_weights_only=True,        # <--- Important!
    save_best_only=True,
    monitor='val_accuracy',
    mode='max',
    verbose=1
)

earlystop_cb = tf.keras.callbacks.EarlyStopping(
    monitor='val_accuracy',
    patience=2,
    restore_best_weights=True,
    verbose=1
)

# Train
history = roberta_model.fit(
    tf_train_dataset,
    validation_data=tf_eval_dataset,
    epochs=epochs,
    callbacks=[checkpoint_cb, earlystop_cb]
)
```

Figure 4.42 Code Snippet for Set up of RoBERTa training

```
Epoch 1/5
3801/3801 [==============================] - ETA: 0s - loss: 0.5696 - accuracy: 0.7618
Epoch 1: val_accuracy improved from -inf to 0.87530, saving model to /content/drive/MyDrive/Modified_Templates/roberta_checkpoints/best_weights.h5
3801/3801 [==============================] - 730s 182ms/step - loss: 0.5696 - accuracy: 0.7618 - val_loss: 0.3600 - val_accuracy: 0.8753
Epoch 2/5
3801/3801 [==============================] - ETA: 0s - loss: 0.3006 - accuracy: 0.8979
Epoch 2: val_accuracy improved from 0.87530 to 0.89700, saving model to /content/drive/MyDrive/Modified_Templates/roberta_checkpoints/best_weights.h5
3801/3801 [==============================] - 657s 173ms/step - loss: 0.3006 - accuracy: 0.8979 - val_loss: 0.3039 - val_accuracy: 0.8970
Epoch 3/5
3801/3801 [==============================] - ETA: 0s - loss: 0.2001 - accuracy: 0.9347
Epoch 3: val_accuracy improved from 0.89700 to 0.91239, saving model to /content/drive/MyDrive/Modified_Templates/roberta_checkpoints/best_weights.h5
3801/3801 [==============================] - 652s 172ms/step - loss: 0.2001 - accuracy: 0.9347 - val_loss: 0.2808 - val_accuracy: 0.9124
Epoch 4/5
3801/3801 [==============================] - ETA: 0s - loss: 0.1337 - accuracy: 0.9571
Epoch 4: val_accuracy improved from 0.91239 to 0.92042, saving model to /content/drive/MyDrive/Modified_Templates/roberta_checkpoints/best_weights.h5
3801/3801 [==============================] - 666s 175ms/step - loss: 0.1337 - accuracy: 0.9571 - val_loss: 0.2539 - val_accuracy: 0.9204
Epoch 5/5
3801/3801 [==============================] - ETA: 0s - loss: 0.0882 - accuracy: 0.9717
Epoch 5: val_accuracy improved from 0.92042 to 0.92423, saving model to /content/drive/MyDrive/Modified_Templates/roberta_checkpoints/best_weights.h5
3801/3801 [==============================] - 632s 166ms/step - loss: 0.0882 - accuracy: 0.9717 - val_loss: 0.2816 - val_accuracy: 0.9242
Restoring model weights from the end of the best epoch: 5.
```

Figure 4.43 Code Snippet training result of BERT-based uncased model

4.8 Initial Results

In this section, test dataset is implemented to evaluate whether the train Bert model can well be performed for sentiment classification and future sentiment prediction.

4.8.1 Model Evaluation for BERT-based Uncased

The result shows that bert-based uncased model performed with strong and balanced accuracy across all sentiment classes. Achieving a 93% overall accuracy. It excell in classifying positive sentiment classification with high precision and recall of 0.95 and 0.94. this indicating both accurate and consistent in predicting of positive sentiment. The models also well in recognizing neutral and negative sentiment with a F1-score of 0.93 and 0.90 respectively.

```
951/951 [==============================] - 62s 56ms/step

--- Classification Report ---
              precision    recall  f1-score   support

    Negative       0.90      0.90      0.90      1870
     Neutral       0.92      0.94      0.93      2295
    Positive       0.95      0.94      0.95      3437

    accuracy                           0.93      7602
   macro avg       0.93      0.93      0.93      7602
weighted avg       0.93      0.93      0.93      7602
```

Figure 4.44 Classification Report for BERT-based Uncased Model

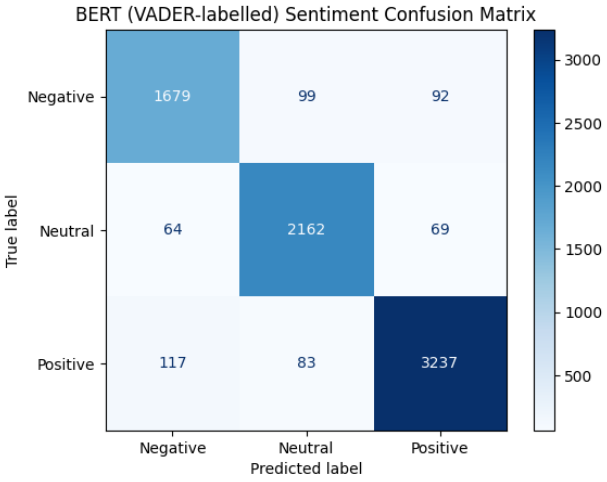


Figure 4.45 Confusion Matric for Bert-based uncased

4.8.2 Model Evaluation for Roberta

The RoBERTa model also shows a strong performance but achieving a slightly lower overall accuracy of 92%. A sentimnt precision of 0.95 indicates fewer false positive when predicting neutral sentiment comment.

```
951/951 [==============================] - 65s 59ms/step

--- Classification Report ---
              precision    recall  f1-score   support

    Negative       0.88      0.91      0.89      1870
     Neutral       0.95      0.92      0.94      2295
    Positive       0.93      0.94      0.93      3437

    accuracy                           0.92      7602
   macro avg       0.92      0.92      0.92      7602
weighted avg       0.92      0.92      0.92      7602
```

Figure 4.44 Classification Report for BERT-based Uncased Model

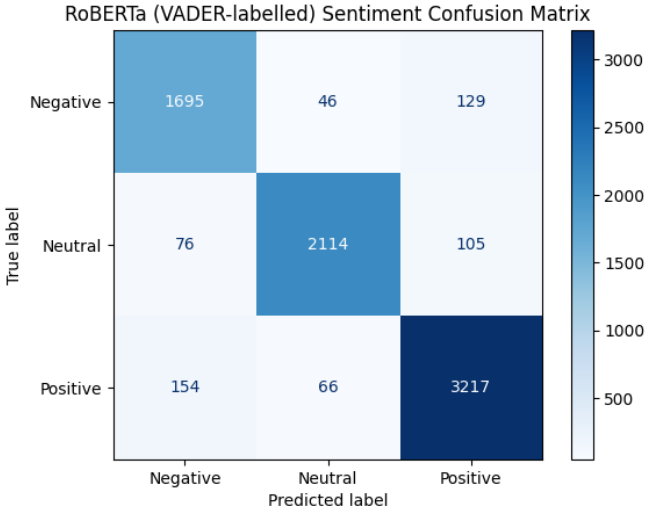


Figure 4.45 Confusion Matric for Bert-based uncased

4.9 Conclusion

The implementation of the project flows is presented in detailed in this chapter. The code flow of each implementation stage is shown starting from data collection to results of the model evaluation.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Introduction

This chapter shows the overall summaries of the finding in for the sentiment analysis of electric vehicle discourse in Malaysia. The result obtain from the exploratory data analysis and sentiment classifier shows an overview result of public reception about electric vehicle related topics. In addition, it also show what possible future improvement can be done on this project for further research improvement on results finding accuracy and more in deep analysis.

5.2 Project Summary

This project is to develop a sentiment classifier model for electric vehicle sentiment analysis discourse in Malaysia. The scrap data from the Reddit consist of discourse containing Malaysian slang with a sample data of 43657. Then the raw data is undergoing data preprocessing step that require before ongoing auto sentiment labeling and model training. Then the performance of both Bert-uncased model and distilled Bert model is compared. In addition, analysis of the sentiment for the electric vehicle is also conducted.

5.3 Overall Achievement

This project had completed 5 chapter including this chapter. Chapter 1 is introduction of electric vehicle discourse, problem background, problem statement, objectives, project scope and the organization of the project. After that, chapter 2 is the literature review section which consist of recent studies and remaining research gap in electric vehicle reviews related domains and researching what type of approach or classifier that had been implement for the natural language text classification. The

next chapter is methodology, which describe every phase of the methodology part starting from data preparation to model evaluation and performance metric evaluation. Chapter 4 shows overall project flow and initial implementation of project model in detail. The initial result obtains from this project experiments also presented in this chapter and discussed.

5.4 Overall Objective Achievement

Table 5.1 Project Objective Achievement

Objective	Achievement	Percentage
To identify, preprocess and explore electric vehicle related text data from multiple online social media and implementing preprocessing procedure to clean data and discover underlying patterns	Data collection had been done through web scraping the Reddit web and obtained a total sample of 43657. Relevant and suitable data preprocessing method had been implemented for the raw collected data.	100%
To Implement and compare pre-trained Bert-based model in	2 pretrained Bert-based model had been implemented for training and predicting the	100%

Objective	Achievement	Percentage
determining structured sentiment analysis.	sentiment analysis of the electric vehicle discourse.	
To analyze the sentiment insight of Malaysian about electric vehicle	The sentiment insight of the Malaysia discourse whether towards positive, negative or neutral had been analyze.	100%

5.5 Future Works and Recommendations and Limitation

Future work that can be implemented in this project for future improvement and 100% achievement of the project is expanding the data sample by web scraping more online social media not only Reddit. Online social media such as YouTube reviews comments or web review news. This is to explore more variety and diversity of online discourse sentence pattern for the model to learn. In addition, can manually human labelling a small portion of the data sample to have a comparison whether the auto labelling and model training performed well or mostly identical to what human interpretation labelling. As the limitation of raw web scraping data lacks of pre-labelled data. Besides that, include topic modelling for in deep insight extraction. To visualized what is the most common discussion for the subtopic for electric vehicle. This would prevent limitation of just visualizing what are the most common words for each of the positive, negative and neutral sentiment.

REFERENCES

Atlas, L. G., Arockiam, D., Muthusamy, A., Balamurugan Balusamy, Shitharth Selvarajan, Taher Al-Shehari, & Alsadhan, N. A. (2025). A modernized approach to sentiment analysis of product reviews using BiGRU and RNN based LSTM deep learning models. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-01104-0

Azim, K., Tahir, A., Mobeen Shahroz, Hanen Karamti, Vazquez, A. A., Vistorte, A. R., & Ashraf, I. (2025). Ensemble stacked model for enhanced identification of sentiments from IMDB reviews. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-97561-8

Borg, A., & Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, *162*, 113746. https://doi.org/10.1016/j.eswa.2020.113746

Bhola, A., Senthil Athithan, Singh, S., Mittal, S., Sharma, Y. K., & Jagjit Singh Dhatterwal. (2022). Hybrid Framework for Sentiment Analysis Using ConvBiLSTM and BERT. *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, *14.2*, 309–314. https://doi.org/10.1109/ictacs56270.2022.9987774

Debnath, R., Ronita Bardhan, Reiner, D. M., & Miller, J. R. (2021). Political, economic, social, technological, legal and environmental dimensions of electric vehicle adoption in the United States: A social-media interaction analysis. *Renewable and Sustainable Energy Reviews*, *152*, 111707–111707. https://doi.org/10.1016/j.rser.2021.111707

EY. (2024, October 15). Malaysia's EV market slows amid consumer worries over charging docks and high maintenance costs. Retrieved May 8, 2025, from Ey.com website: https://www.ey.com/en_my/newsroom/2024/10/malaysias-ev-market-slows-amid-consumer-worries-over-charging-docks-and-high-maintenance-costs

Fam, A., & Fam, S. (2024). Review of the US 2050 long term strategy to reach net zero carbon emissions. Energy Reports, 12, 845–860. https://doi.org/10.1016/j.egyr.2024.06.031

Ferdous, S. M., Syed, Bin, S., & Uddin, M. (2024). Sentiment Analysis in the Transformative Era of Machine Learning: A Comprehensive Review. *Statistics Optimization & Information Computing*, *13*(1), 331–346. https://doi.org/10.19139/soic-2310-5070-2113

Fernandez, M. I., Go, Y. I., Wong, D., & Wolf-Gerrit Früh. (2024). Malaysia's energy transition and readiness towards attaining net zero: review of the potential, constraints, and enablers. *Renewable Energy Focus*, *51*, 100640–100640. https://doi.org/10.1016/j.ref.2024.100640

Gardazi, N. M., Daud, A., Malik, M. K., Bukhari, A., Tariq Alsahfi, & Bader Alshemaimri. (2025). BERT applications in natural language processing: a review. *Artificial Intelligence Review*, *58*(6). https://doi.org/10.1007/s10462-025-11162-5

Gaurav, A., Gupta, B. B., & Chui, K. T. (2024). BERT Based Model for Robust Mental Health Analysis in Clinical Informatics. *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 153–160. https://doi.org/10.1109/jcsse61278.2024.10613729

Government of Malaysia. (2025). Car Popularity Explorer | data.gov.my. Retrieved May 8, 2025, from Data.gov.my website: https://data.gov.my/dashboard/car-popularity

Guven, Z. A. (2021). Comparison of BERT Models and Machine Learning Methods for Sentiment Analysis on Turkish Tweets. *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 98–101. https://doi.org/10.1109/ubmk52708.2021.9559014

Hafize Nurgul Durmus Senyapar. (2024). Electric Vehicles in the Digital Discourse: A Sentiment Analysis of Social Media Engagement for Turkey. *SAGE Open*, *14*(4). https://doi.org/10.1177/21582440241295945

Higueras-Castillo, E., Molinillo, S., Coca-Stefaniak, J. A., & Liébana-Cabanillas, F. (2019). Perceived Value and Customer Adoption of Electric and Hybrid Vehicles. *Sustainability*, *11*(18), 4956. https://doi.org/10.3390/su11184956

IPCC, 2022: Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental

Panel on Climate Change [P.R. Shukla, J. Skea, R. Slade, A. Al Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, J. Malley, (eds.)]. Cambridge University Press, Cambridge, UK and New York, NY, USA. doi: 10.1017/9781009157926

Liang, D., Li, F., & Chen, X. (2023). Failure mode and effect analysis by exploiting text mining and multi-view group consensus for the defect detection of electric vehicles in social media data. *Annals of Operations Research*, *340*(1), 289–324. https://doi.org/10.1007/s10479-023-05649-z

Littlejohn, C., & Stef Proost. (2022). What role for electric vehicles in the decarbonization of the car transport sector in Europe? *Economics of Transportation*, *32*, 100283–100283. https://doi.org/10.1016/j.ecotra.2022.100283

Majekodunmi, T. B., Mohd Shahidan Shaari, Nor Fadzilah Zainal, Harun, N. H., Ridzuan, A. R., Noorazeela Zainol Abidin, & Nur. (2023). Gas Consumption as a Key for Low Carbon State and its Impact on Economic Growth in Malaysia: ARDL Approach. *International Journal of Energy Economics and Policy*, *13*(3), 469–477. https://doi.org/10.32479/ijeep.14134

Malaysia - Countries & Regions - IEA. (2025). Malaysia - Countries & Regions - IEA. Retrieved May 8, 2025, from IEA website: https://www.iea.org/countries/malaysia/emissions

Portal Rasmi Kementerian Sumber Asli dan Kelestarian Alam. (2025). Retrieved May 8, 2025, from Nres.gov.my website: https://www.nres.gov.my/ms-my/_layouts/15/osssearchresults.aspx#k=50%25electric%20vehicle#l=1086

Ruan, T., & Qin Lv. (2023). Public perception of electric vehicles on Reddit and Twitter: A cross-platform analysis. *Transportation Research Interdisciplinary Perspectives*, *21*, 100872–100872. https://doi.org/10.1016/j.trip.2023.100872

Rahman, B., & Maryani. (2024). Optimizing Customer Satisfaction Through Sentiment Analysis: A BERT-Based Machine Learning Approach to Extract Insights. *IEEE Access*, *12*, 151476–151489. https://doi.org/10.1109/access.2024.3478835

Wibowo, A. S., & Dovi Septiari. (2023). How Does the Public React to the Electric Vehicle Tax Incentive Policy? A Sentiment Analysis. *Journal of Tax Reform*, *9*(3), 413–429. https://doi.org/10.15826/jtr.2023.9.3.150

Wu, Z., He, Q., Li, J., Bi, G., & Antwi-Afari, M. F. (2023). Public attitudes and sentiments towards new energy vehicles in China: A text mining approach. *Renewable and Sustainable Energy Reviews*, *178*, 113242–113242. https://doi.org/10.1016/j.rser.2023.113242

Saleh Mohammed Kutabish, Ana Maria Soares, & Casais, B. (2023). The Influence of Online Ratings and Reviews in Consumer Buying Behavior: a Systematic Literature Review. *Lecture Notes in Business Information Processing*, *485*, 113–136. https://doi.org/10.1007/978-3-031-42788-6_8

Siew, T., Hui, N., & Rebecca, Y. (2024). *A Study of the Emerging Electric Vehicle (EV) Supply Chain in Malaysia*. Retrieved from https://www.iseas.edu.sg/wp-content/uploads/2024/04/ISEAS_Perspective_2024_33.pdf

Sornalakshmi, R. R., Ramu, M., Raghuveer, K., Vuyyuru, V. A., Venkateswarlu, D., & Balakumar, A. (2024). Harmful News Detection using BERT model through sentimental analysis. *2024 International Conference on Intelligent Systems and Advanced Applications (ICISAA)*, 1–5. https://doi.org/10.1109/icisaa62385.2024.10828690

Sharma, H., Din, F. U., & Ogunleye, B. (2024). Electric Vehicle Sentiment Analysis Using Large Language Models. *Analytics*, *3*(4), 425–438. https://doi.org/10.3390/analytics3040023

Tax Relief | Lembaga Hasil Dalam Negeri Malaysia. (2025). Retrieved May 8, 2025, from Lembaga Hasil Dalam Negeri Malaysia website: https://www.hasil.gov.my/individu/kitaran-cukai-individu/lapor-pendapatan/pelepasan-cukai/

Cao, J. (2024). *An improved RCBT model and its application to sentiment analysis of movie comments*. 1016–1023. https://doi.org/10.1109/ispcem64498.2024.00180

Cui, Q., Zhang, Y., Ma, H., Zhang, K., Peng, J., Chen, Z., … Lin, Z. (2025). How about electric vehicle? Sensing owners' experiences and attitudes through online short video. *Transport Policy*, *167*, 1–15. https://doi.org/10.1016/j.tranpol.2025.03.012

Du, J., Mamo, Y. Z., Floyd, C., Karthikeyan, N., & James, J. D. (2023). Machine Learning in Sport Social Media Research: Practical Uses and Opportunities. *International Journal of Sport Communication*, *17*(1), 97–106. https://doi.org/10.1123/ijsc.2023-0151

Gadi, M. F. A., & Sicilia, M. Á. (2024). A sentiment corpus for the cryptocurrency financial domain: the CryptoLin corpus. Language Resources and Evaluation, 59(2), 871–889. https://doi.org/10.1007/s10579-024-09743-x

Hussein, H. H., & Lakizadeh, A. (2025). A systematic assessment of sentiment analysis models on iraqi dialect-based texts. Systems and Soft Computing, 7, 200203. https://doi.org/10.1016/j.sasc.2025.200203

Khalid, M., Raza, A., Younas, F., Furqan Rustam, Villar, M. G., Ashraf, I., & Akhtar, A. (2024). Novel Sentiment Majority Voting Classifier and Transfer Learning-Based Feature Engineering for Sentiment Analysis of Deepfake Tweets. *IEEE Access*, *12*, 67117–67129. https://doi.org/10.1109/access.2024.3398582

Liu, J., Pan, H., Luo, R., Chen, H., Tao, Z., & Wu, Z. (2025). An electric vehicle sales hybrid forecasting method based on improved sentiment analysis model and secondary decomposition. *Engineering Applications of Artificial Intelligence*, *150*, 110561–110561. https://doi.org/10.1016/j.engappai.2025.110561

Mao, J., Qian, Z., & Lucas, T. (2023). Sentiment Analysis of Animated Online Education Texts Using Long Short-Term Memory Networks in the Context of the Internet of Things. *IEEE Access*, *11*, 109121–109130. https://doi.org/10.1109/access.2023.3321303

Mahadevaswamy, U. B., & Swathi, P. (2023). Sentiment Analysis using Bidirectional LSTM Network. *Procedia Computer Science*, *218*, 45–56. https://doi.org/10.1016/j.procs.2022.12.400

M Oriza Syahputra, Bustami Bustami, & Lidya Rosnita. (2024). Analysis of Public Sentiment Towards Celebrity Endorsment On Social Media Using Support Vector Machine. International Journal of Engineering Science and Information Technology, 4(3), 118–127. https://doi.org/10.52088/ijesty.v4i3.543

Özkara, Y., Bilişli, Y., Yildirim, F. S., Kayan, F., Başdeğirmen, A., Kayakuş, M., & Yiğit Açıkgöz, F. (2025). Analysing Social Media Discourse on Electric Vehicles with Machine Learning. *Applied Sciences*, *15*(8), 4395. https://doi.org/10.3390/app15084395

Penava, P., & Buettner, R. (2025). A novel subject-independent deep learning approach for user behavior prediction in electronic markets based on electroencephalographic data. *Electronic Markets*, *35*(1). https://doi.org/10.1007/s12525-025-00778-8

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). MPNet: Masked and Permuted Pre-training for Language Understanding. Retrieved May 22, 2025, from arXiv.org website: https://arxiv.org/abs/2004.09297

Smith-Mutegi, D., Mamo, Y., Kim, J., Crompton, H., & McConnell, M. (2025). Perceptions of STEM education and artificial intelligence: a Twitter (X) sentiment analysis. International Journal of STEM Education, 12(1). https://doi.org/10.1186/s40594-025-00527-5

Sayyida Tabinda Kokab, Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. Array, 14, 100157–100157. https://doi.org/10.1016/j.array.2022.100157

Tabinda Kokab, S., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. Array, 14, 100157. https://doi.org/10.1016/j.array.2022.100157

Tiwari, P., Mishra, B. K., Kumar, S., & Kumar, V. (2020). Implementation of n-gram Methodology for Rotten Tomatoes Review Dataset Sentiment Analysis. *IGI Global EBooks*, 689–701. https://doi.org/10.4018/978-1-7998-2460-2.ch036

Ullah, A., Khan, K., Khan, A., & Ullah, S. (2023). Understanding Quality of Products from Customers' Attitude Using Advanced Machine Learning Methods. Computers, 12(3), 49. https://doi.org/10.3390/computers12030049

Wang, M., You, H., Ma, H., Sun, X., & Wang, Z. (2023). Sentiment Analysis of Online New Energy Vehicle Reviews. *Applied Sciences*, *13*(14), 8176–8176. https://doi.org/10.3390/app13148176

Wei, J., Liao, J., Yang, Z., Wang, S., & Zhao, Q. (2020). BiLSTM with Multi-Polarity Orthogonal Attention for Implicit Sentiment Analysis. *Neurocomputing*, *383*, 165–173. https://doi.org/10.1016/j.neucom.2019.11.054

Wibowo, A. S., & Septiari, D. (2023). How Does the Public React to the Electric Vehicle Tax Incentive Policy? A Sentiment Analysis. *Journal of Tax Reform*, *9*(3), 413–429. https://doi.org/10.15826/jtr.2023.9.3.150

Wu, Z., He, Q., Li, J., Bi, G., & Antwi-Afari, M. F. (2023). Public attitudes and sentiments towards new energy vehicles in China: A text mining approach. *Renewable and Sustainable Energy Reviews*, *178*, 113242–113242. https://doi.org/10.1016/j.rser.2023.113242

Xu, Z., Wen, X., Zhong, G., & Fang, Q. (2025). Public perception towards deepfake through topic modelling and sentiment analysis of social media data. Social Network Analysis and Mining, 15(1). https://doi.org/10.1007/s13278-025-01445-8