Using Sentiment Analysis to Forecast Stock Short-Term Trend

By

Chong Kai Jian

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

FEBRUARY 2026

ACKNOWLEDGEMENTS

I want to express my sincere thanks and appreciation to my supervisor, Dr Kh'ng Xin Yi, who has given me this bright opportunity to engage in a project using sentiment analysis to forecast the short-term trend of stocks. It is my first step to establish a career in the sentiment analysis field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ABSTRACT

Investor sentiment, alongside historical price movements and macroeconomic signals, plays a significant role in influencing financial markets. However, the effect of news sentiment on stock prices is often delayed rather than instantaneous, and conventional binary sentiment representations are insufficient to capture these dynamic and lagged relationships for short-term forecasting. Furthermore, the effectiveness of sentiment extraction may vary depending on the underlying language model used. This project aims to develop a sentiment-based stock forecasting system that predicts next-day stock prices using financial news data. News articles are collected from multiple sources, including The Star, Free Malaysia Today, i3Investor and Yahoo Finance, and transformed into sentiment scores using different GPT models. These sentiment signals are aligned with stock closing prices to support short-term forecasting for Malaysian banking stocks, specifically Maybank and Public Bank. A comparative analysis of sentiment representations generated by GPT-5, GPT-5-mini, and GPT-5-nano is conducted to evaluate their effectiveness in capturing meaningful market signals. Subsequently, the integrated sentiment and stock data are incorporated into a prediction framework based on the Autoregressive Distributed Lag (ARDL) model, which captures both historical price dependencies and lagged sentiment effects. The model predicts next-day closing prices using current price and sentiment information, with optimal lag structures selected automatically based on model performance. Model evaluation is conducted using RMSE, MAE, and AIC. Last but not least, to enhance usability and interpretability, a web-based user interface is developed, enabling users to configure parameters such as bank selection, GPT model, and news source, perform predictions, visualize forecasting results, compare model performance, and analyse sentiment effects. The results demonstrate that the ARDL model achieves strong short-term forecasting performance, with RMSE values of approximately 0.13 and predicted prices closely tracking actual trends. For Maybank, the selected ARDL(1,3) model incorporates sentiment lags; however, lag removal experiments and Error Correction Model (ECM) analysis indicate that the 3-day sentiment lag is not statistically significant and can be excluded. In contrast, for Public Bank, the selected ARDL(1,0) model excludes sentiment entirely, suggesting limited predictive contribution of

sentiment in this case. Further analysis shows that larger models such as GPT-5 generate more variable and informative sentiment scores, leading to richer lag structures, whereas smaller models such as GPT-5-nano produce more unstable but less informative sentiment, resulting in simpler models and less consistent lag selection. Overall, the project presents a complete end-to-end stock prediction system that integrates data collection, sentiment extraction, modelling, and user interaction, demonstrating the practical applicability of sentiment-aware forecasting in financial analysis.

Area of Study (Minimum 1 and Maximum 2): NLP stock analysis
Keywords (Minimum 5 and Maximum 10): Sentiment Analysis, NLP, ARDL Regression, Short-Term Trend, Stock Price Direction

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

y_t	Dependent variable at time t(e.g., next-day stock price)
y_{t-i}	Lagged value of the dependent variable at lag i
x_t	Independent variable at time t(e.g., sentiment score)
x_{t-l}	Lagged value of the independent variable at lag l
a_i	Coefficient of the i-th lag of the dependent variable
b_l	Coefficient of the l-th lag of the independent variable
j	Number of lags for the dependent variable (autoregressive order)
k	Number of lags for the independent variable (distributed lag order)
e_t	Error term (white noise) at time t
t	Time index
ΔY_t	Change in the dependent variable at time t
ΔY_{t-i}	Lagged change of the dependent variable at lag i
ΔX_{t-j}	Lagged change of the independent variable at lag j
Y_{t-1}	Lagged level of the dependent variable
X_{t-1}	Lagged level of the independent variable
α_0	Intercept term
β_i	Short-run coefficient of the i-th lag of the dependent variable
γ_j	Short-run coefficient of the j-th lag of the independent variable
θ	Long-run coefficient of the independent variable
λ	Error correction coefficient (speed of adjustment towards long-run equilibrium)
$(Y_{t-1} - \theta X_{t-1})$	Error Correction Term (ECT), representing deviation from long-run equilibrium
ε_t	Error term at time t
y	Actual value
$\hat{y}$	Predicted value
n	Total number of observations
k	Number of estimated parameters in the model

$\mathcal{L}$	Maximum value of the likelihood function
$\ln(\mathcal{L})$	Natural logarithm of the likelihood function
p	Number of lags for the dependent variable (autoregressive order)
q	Number of lags for the independent variable (distributed lag order)

LIST OF ABBREVIATIONS

ARDL	Autoregressive Distributed Lag
NLP	Natural Language Processing
ECM	Error Correction Model
ECT	Error Correction Term
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
AIC	Akaike Information Criterion
GUI	Graphical User Interface
CV	Cross-Validation

CHAPTER 1

Introduction

Stock prices are influenced not only by earnings reports, interest rates, or GDP data, but also by investor sentiment, as widely documented in prior studies [1] and [2].In this digital age, a vast number of investor sentiment is shared through financial news, analyst opinions, and discussions on social media platforms such as Twitter, Reddit, and financial blogs. This view is further supported by studies on social media sentiment and financial markets [3].

Although traditional stock forecasting models have performed well using historical and technical indicators, they fall short when it comes to real-time analysis. This is because traditional models are not designed to process unstructured textual data or to understand the nuances of human language. This is the reason that why integration of sentiment analysis into forecasting frameworks is significant. It is enhancing predictive accuracy, particularly for short-term stock trend.

The use of advanced NLP techniques provides an opportunity to overcome the shortcomings of earlier sentiment analysis methods, as discussed in [4]. Tools like ChatGPT, has demonstrate the ability to interpret complex textual data with high accuracy. This project is utilizing ChatGPT to classify and quantify sentiment from financial news and combining these sentiments with historical stock data. This project proposes the application of the ARDL regression model to capture both short-run and long-run relationships between sentiment and stock price trends due to the ability to model delayed sentiment effects, capturing how sentiment from previous days may continue to influence stock behavior.

1.1 Problem Statement and Motivation

Even though the usability of financial news and public comment is increasing, many traditional predictive models still ignoring this useful source. Sentiment analysis in the financial domain remains a challenging task due to several reasons:

i. Ambiguity and Sarcasm: Financial news and commentary often use nuanced language. Sarcastic remarks or ambiguous statements can be misclassified by simple sentiment tools.
ii. Context-Specific Vocabulary: Financial jargon and context-dependent meanings (e.g., "beat estimates" vs. "missed expectations") can lead to inaccurate sentiment labeling.
iii. Simplistic Classification: Some models reduce sentiment into binary (positive/negative) categories, overlooking the spectrum of emotions and market responses.
iv. Lack of Integration with Time-Series Models: Even when sentiment is correctly extracted, few systems adequately integrate this data with models like ARDL that can measure both immediate and delayed effects on stock prices.
v. Variability Across News Sources: Financial news is collected from multiple platforms, each with different writing styles, credibility levels, and reporting focus. These differences may lead to variations in sentiment interpretation.

This project addresses this gap by employing ChatGPT for more nuanced sentiment extraction and integrating this data into an ARDL model to predict short-term movements in stock prices.

1.2 Research Objectives

1.2.1 To perform sentiment analysis on financial news data.

News will be collected from a variety of financial news platforms, such as Yahoo Finance, The Star, Free Malaysia Today, i3Investor, etc., to analyze how sentiments from different sources influence stock market movements. Perform sentiment analysis using the GPT-5 models, specifically the GPT-5 variant developed by OpenAI, to classify news into 1(strongly positive) to -1(strongly negative). This analysis aims to evaluate the emotional tone of financial news from different sources and provide reliable sentiment insights for future applications.

1.2.2 To implement ARDL regression for predicting stock trends.

Apply the Autoregressive Distributed Lag (ARDL) model to quantify the relationship between news sentiment and stock data changes, and forecast short-term stock trends based on the sentiment analysis results. This method allows for analyzing both short-term dynamics and long-term equilibrium between variables. The goal is to enhance the

accuracy of stock trend forecasts by incorporating the temporal effects of news sentiment on stock performance.

1.2.3 To evaluate the performance of different GPT models on stock prediction.

Different GPT models (GPT-5, GPT-5-mini, and GPT-5-nano) will be used to generate sentiment scores, which are then incorporated into the ARDL model. The performance of these models will be evaluated by comparing forecasting accuracy metrics such as RMSE, MAE, and AIC. The objective is to analyze how variations in sentiment generation affect model performance, lag selection, and overall prediction accuracy.

1.2.4 To develop web-based UI for stock prediction based on news.

A web-based user interface will be developed using Streamlit to provide an interactive platform for users to explore forecasting results. The interface allows users to select different banks, GPT models, and news sources, and visualize outputs such as stock price trends, sentiment trends, and ARDL forecasting results.
The objective is to improve the accessibility and interpretability of the system by presenting complex analytical outputs in a user-friendly and interactive format.

1.3 Project Scope and Direction

This project aims to develop a sentiment forecasting system that predicts short-term stock market trends by integrating automated data collection, natural language processing, and time series regression (ARDL regression). The process begins with the development of a custom program to automatically collect real-time and historical stock data along with related financial news from online sources such as financial news platforms. This automated data pipeline ensures that the analysis is based on timely and relevant information.

The next phase is going to apply sentiment analysis to the collected news content using ChatGPT, which is capable of understanding and interpreting financial text. ChatGPT will classify the sentiment of each news into three categories based on its content, whether positive, negative, or neutral. These sentiment scores will be time-stamped and synchronized with the stock data which have the same time-stamp to create a structured dataset for further analysis.

Finally, the project will apply the Autoregressive Distributed Lag (ARDL) regression model to analyze the time relationship between sentiment scores and stock price movements. The ARDL model is selected because it is able to capture both short-term dynamics and long-term trends, making it well-suited for forecasting in financial time series. The system aims to provide a robust and interpretable prediction of stock trends by modeling how past and current sentiment influences market behavior. This project will initially focus on a selected set of stocks or indices, offering a practical demonstration of how advanced NLP and econometric techniques can be combined for financial forecasting.

1.4 Contributions

This project makes several meaningful contributions to the financial forecasting domain and NLP application:

i. **Advanced Sentiment Analysis in Finance:** It reveals the application of ChatGPT for performing sentiment analysis specifically in the financial domain. Different GPT model has different capable of understanding nuanced financial language and delivering sentiment insight.
ii. **Integrated Framework for Forecasting:** The project introduces the integration of sentiment analysis with the ARDL regression model, allowing for a comprehensive examination between sentiment and stock price movements.
iii. **News Source Impact Analysis on Stock Prediction:** This study investigates how sentiment derived from different financial news sources influences stock prediction performance. The analysis highlights that variations in news content and reporting style can affect sentiment signals and model outcomes.
iv. **Web-based user interface development:** A web-based system is developed to visualize stock prices, sentiment trends, forecasting results, and model comparisons. The dashboard enhances interpretability and provides a practical interface for analyzing sentiment-driven stock prediction.

1.5 Report Organization

The details of this research are shown in the following chapters. Chapter 2 presents a literature review, summarizing Previous Works on Stock Prediction and Sentiment Analysis Using Large Language Models, while highlighting their limitations and also some situation that didn't considered by authors. Chapter 3 show the proposed method, basically talk about the sentiment analysis framework and the modeling approach for forecasting short-term stock trends. Chapter 4 discusses the preliminary work, we only focus on data collection, preprocessing, and data integration in FYP1. Finally, Chapter 5 concludes the study and suggesting potential directions for future research.

CHAPTER 2

Literature Reviews

2.1 Previous Works on Stock Prediction and Sentiment Analysis

In recent years, stock forecasting has gained attention due to the growing number of investors. Here is some research that explored various models that integrate machine learning, sentiment analysis, and time series forecasting to predict stock trends. This chapter reviews significant improvements in this domain, focusing on sentiment extraction methods and predictive models used in stock trend forecasting. The strengths and limitations of each approach are discussed to highlight the existing constraints and justify the proposed methodology in this study.

2.1.1 Using Several Machine Learning Methods

In [5], Q. Xiao and B. Ihnaini proposed a stock prediction model to help stock investors make proper investment decisions by using tweets, news, and historical stock data. They send the news and tweets data to the sentiment analysis test section(Financial PhraseBank and Tweets_labelled_09042020_16072020) to select the most appropriate technology(VADER sentiment analysis, Loughran-McDonald word dictionary, and FinBERT Model) for applying sentiment analysis and generated weighted scores for both of them, both news and tweets scores are summed up as input features after apply holiday effect processing. While stock data, turn into binary changes. If the next day's value is higher than today, change the value to 1; otherwise, change the value to 0 and the outcome considers an expected result in machine learning. By the way, they also contain retweet counts when they are calculating the weighted score of tweet data. Their results show that Naïve Bayes is the best classification algorithm, which indicated an accuracy rate of 62.4% compared to other researchers which only get 63.6% accuracy by using seven features, to show that their system is more effective.

This model has several limitations. First, their sentiment analysis approach relies on predefined models like VADER, Loughran-McDonald, and FinBERT, which may not perfectly handle the financial news and tweets when sarcasm, abbreviations, or emerging financial jargon appears in the news and tweets. Second, their binary

classification of stock movement (1 for increase, 0 for decrease) may oversimplify the market, it might ignore the magnitude of changes that could be crucial for investors. Additionally, the accuracy rate of 62.4% shows that the performance of the model still needs to be improved, particularly in feature engineering and data processing.

To address these limitations, instead of binary classification, we can implement a regression-based approach, which could provide a more accurate prediction of stock price movements. For example, the Autoregressive Distributed Lag (ARDL) regression model is suitable for this task, because it can identify relationships between sentiment scores and stock data. The difference with traditional classification models is that traditional classification models merely predict directional trends (e.g., up or down), but ARDL regression allows us to quantify the extent of stock data changes in response to sentiment shifts while considering time lags. This approach can improve forecasting accuracy by incorporating temporal dependencies and enabling better interpretation of the underlying economic relationships.

2.1.2 Using Dictionary-Based Sentiment Detection and Some Machine Learning

In [6], K. Joshi, B. H. N, and J. Rao proposed a method to automate sentiment detection for stock price prediction to simplify the process. We can identify overall news polarity based on the words in the news. So they used a polarity detection algorithm for initially labeling news and making the training set. This algorithm is a dictionary-based approach, the positive and negative words are created using general and finance-specific sentiment words. They also created their own dictionary for stop word removal, which also includes finance-specific stop words for pre-processing text data.

There are some constraints in [6]. Their dictionary-based polarity detection algorithm is limited in adaptability because financial terminology evolves over time. Sentiment in the financial text cannot be easily classified because it may include sarcasm, abbreviations, etc. A rule-based approach may not always perform well on unseen data. Furthermore, customizing stop-word dictionaries requires huge human resources.

2.1.3 Using Stock Data and Semi-Supervised Learning

In [7], S. Paul and S. Vishnoi proposed two models, one is to extract the sentiments based on the sentiments of news, second is the prediction of movement of stock returns based on adjusted close price. They primarily use the Semi-Supervised Learning

Technique, which includes both Manual Classification and Automatic Classification methods for labeling news sentiment. The labeled data is used to train and evaluate machine learning models, such as SVM and LSTM. The outcome of the Price-Only Method is like the expected outcome for machine learning, to compare its performance with the sentiment-based approach. A key advantage of this approach is that by combining sentiment analysis with stock price movements, the model can leverage both qualitative and quantitative data, which might potentially improve prediction accuracy compared to using the Price-Only Method.

The limitation in [7] that has been mentioned by authors already, is less effective for prediction purpose, insufficient for prediction of large movements in stocks. This is because there have several factors affecting stock price movement, and impact of news only take up 52%. Additionally, their model didn't considered other factors, such as transaction cost, borrowing cost and the tendency of large trading to move the market.

2.1.4 Using News Collected by Sentiment140 and Multilayer Perceptron Models

Chakraborty et al. [8] employed the Sentiment140 dataset in their study on predicting stock movement from Twitter sentiment. This dataset is available on Kaggle, and it contains about 1.6 million labeled tweets obtained via the Sentiment140 API. Each tweet is automatically tagged as either positive or negative, represented by the labels "1" and "0." To prepare the data for model training, the authors randomly split it into two parts: approximately 1.52 million tweets were allocated for training, while around 80,000 tweets were used for testing. The distribution of the dataset between positive and negative categories is shown in Table 2.1.

Table 2.1 Data distribution of research paper [8]

	Training Data (# of Tweets)	Testing Data (# of Tweets)
Positive (# of Tweets with Positive Sentiment)	760,154	39,989
Negative (# of Tweets with Negative Sentiment)	759,846	40,011

As shown in Table 2.1, the data is reasonably balanced with an almost equal amount of Positive and Negative tweets in both the Sentiment 140 testing and training data.

After that, they used Boosted Regression Tree and Multilayer Perceptron models to predict the next day's stock movement by using the present day's tweets containing the

"stock market", "StockTwits", and "AAPL" as input of machine learning. Furthermore, the result of this paper implies that Neural Networks has better performance compare with the Boosted Regression Tree. It is also clear that too high and too low differences in Stock Indexes are challenging to predict with a Boosted Regression Tree. However, except for those days, Their models perform well on the given data set.

The limitation in [8] is collected dataset by using Sentiment 140 only may cause the model not general enough, because Sentiment 140 is a widely used dataset of 1.6 million tweets created by Stanford researchers to help develop and evaluate sentiment analysis models. This dataset may not fully represent the present stock market. Additionally, their Boosted Regression Tree struggles with extreme stock price movements, leading to inaccuracies on volatile trading days.

2.1.5 Using Deep-Learning-Based Financial Text Sentiment Classification Method

In [9], the authors proposed a deep-learning-based financial text sentiment classification method that integrates domain adaptation techniques to address the issue of insufficient labeled samples in financial text classification. Their approach consists of three main components: an abstract extraction module using a pre-trained model(seq2seq), a feature extraction subnetwork based on a bidirectional LSTM, and the last, a Sentiment Classification Subnetwork and Domain Classification Subnetwork. The model effectively leverages both labeled and unlabeled data by incorporating domain adaptation with sentiment classification and domain classification losses. The results of this paper show that this method improves the accuracy of sentiment classification in financial texts compared to traditional machine learning. A key advantage of this approach is it reduces the impact of invalid information on classification, which might enhance model performance in real-world financial applications.

In [9], the pre-trained seq2seq abstract extraction model is directly adopted, but the financial terminology evolves over time, it may cause outdated after few years ago. Moreover, domain adaptation techniques may introduce transfer learning limitations if the target domain is significantly different from the source domain.

2.2 Sentiment Analysis Using Large Language Models

With the rise of Large Language Models (LLMs), sentiment analysis in financial domains has entered a new phase of development. LLMs can understand context, semantics, and domain-specific expressions with greater depth and nuance compared with traditional sentiment analysis tools. Their capacity to perform zero-shot and few-shot learning enables them to adapt to diverse financial texts without vast labeled datasets. This section explores how LLMs are applied in financial sentiment analysis and prediction and the limitations of LLM-based models, such as computational complexity and reduced performance on fine-grained sentiment tasks.

2.2.1 Using ChatGPT for Sentiment Analysis and Price Prediction in the Cryptocurrency Market

In [10], Large Language Models (LLMs) have emerged as powerful tools for financial sentiment analysis due to their ability to efficiently extract, quantify, and interpret sentiment from textual data. One of the most advanced LLM-based systems that leverages natural language processing and deep learning techniques to understand and generate human-like language is ChatGPT. Researchers have started exploring AI applications in analyzing financial sentiment, which means AI techniques are integrated into economic and financial domains. Researchers use ChatGPT to capture, clean, filter, and perform sentiment analysis on online data so that they are able to analyze the sentiment of textual data to ensure high reliability of the result. The regression models showed that adding sentiment indicators greatly improved prediction accuracy, underscoring the critical role that sentiment acts in financial risk forecasting and decision-making.

However, the study acknowledges certain limitations. One of the significant constraints is the inherent volatility and unpredictability of the cryptocurrency market can challenge the model's forecasting performances. Additionally, the reliance on textual data from social media and news sources may introduce biases and noise, potentially affecting the sentiment analysis accuracy.

2.2.2 Using Large Language Models to Enhancing Financial Sentiment Analysis

This study targets the application of LLMs in the financial sentiment analysis domain, where understanding sentiment often requires domain-specific knowledge. Zhang et al.

propose a retrieval-augmented language modeling framework (RA-LLM) that enhances LLMs with extra context. The model will retrieve relevant documents or prior reports and incorporate them into the cue when analyzing a financial news item or tweet. These contextual cues help the LLM interpret implicit meaning and jargon more accurately [11].

A significant limitation mentioned by the authors is that financial texts are typically short and lacking in context, it causes making them difficult to interpret even for large models [11]. Furthermore, the quality of the retrieval mechanism will directly affect performance. If it retrieved irrelevant documents, it may introduce bias or confusion, and reduce the overall effectiveness of the system.

The authors recommend improving the retrieval mechanism with better ranking and filtering mechanisms to address these issues [11]. They also suggest integrating a fine-tuning phase by using domain-specific corpora, which could help the LLM understand the implicit cues of financial texts.

2.2.3 A Reality Check of Sentiment Analysis in the Era of Large Language Models

Zhang et al. conduct a thorough evaluation of how large language models (LLMs) perform on a wide range of sentiment analysis tasks, such as GPT and BERT-based variants [12]. These tasks consist of simple polarity classification and complex forms like aspect-based sentiment analysis and emotion detection. The authors test 13 tasks across 26 datasets, and finally provide a realistic benchmark named "SentiEval". They compare performance in both zero-shot and few-shot settings to understand the actual utility of LLMs when labeled data is not enough.

Authors have identified one key limitation, which is LLMs can perform basic sentiment classification tasks normally, but they often fall short on more complex sentiment understanding, such as identifying sentiment tied to specific aspects or managing nuanced and implicit expressions of emotion. The models also don't perform well in tasks that require deeper contextual reasoning or multi-hop inference, these scenarios indicate a gap between general language understanding and fine-grained sentiment interpretation.

2.2.4 Integration of Explainable AI Techniques with Large Language Models for Enhanced Interpretability in Sentiment Analysis

Thogesan and his colleagues explore the integration of explainable AI (XAI) techniques with LLMs to improve transparency in sentiment analysis. They propose a framework where the outputs of LLMs are fulfilled by interpretability methods such as SHAP (Shapley Additive explanations) and attention-based visualizations [13]. The goal is to make sentiment predictions more transparent, especially in sensitive areas like mental health and finance, it means the experts that have certain knowledge understand the LLMs prediction, because understanding why a model predicted a certain sentiment is just as important as the prediction itself.

The limitation of the paper [13] is adding explainability layers to LLM-based models will introduced computational overhead. Furthermore, interpretability tools like SHAP are traditionally designed for smaller models and may not scale effectively with the complexity of LLMs. There's also a risk that explanations can be misleading if the LLM prediction is not readable to experts.

2.3 Prediction Method Implemented in This Project

While various machine learning models and sentiment analysis techniques have been applied in stock prediction, many studies still cannot capture the time-dependent relationship between investor sentiment and stock price movements effectively. Most approaches are limited to predicting the price changes without considering the temporal lag of sentiment effects. This project adopts a regression-based method to address this issue. The ARDL regression is particularly suitable for examining both short-term and long-term relationships between sentiment and stock data, even when the variables are integrated at different orders.

2.3.1 ARDL Regression Model

The equation of ARDL (Autoregressive Distributed Lag) regression model is

$$y_t = a_1 y_{t-1} + \ldots a_j y_{t-j} + b_0 x_t + \ldots b_k x_{t-k} + e_t$$

The same model can be written as

$$y_t = \sum_{i=1}^{j} a_i y_{t-i} + \sum_{l=0}^{k} b_l x_{t-l} + e_t$$

An ARDL (Autoregressive Distributed Lag) regression model is a dynamic time series model used to analyze the relationship between a dependent variable and one or more independent variables, where the independent variables can have both current and lagged values. ARDL models are technically AR-X models that have been mentioned in the paper [14], it is focused on the exogenous variables and select the correct lag structure from both the endogenous variable and the exogenous variables. ARDL models are also closely related to Vector Autoregressions, and a single ARDL is effectively one row of a VAR. The key distinction is that an ARDL assumes that the exogenous variables are exogenous in the sense that it is not necessary to include the endogenous variable as a predictor of the exogenous variables.

In this project, we extracted sentiment scores from online financial news using natural language processing (NLP) techniques. ARDL regression enables the modeling of delayed sentiment effects, capturing how sentiment from previous days may continue to influence stock behavior. This approach provides more insight than binary classification models, as it not only forecasts direction but also quantifies the magnitude of price movements over time.

I have found support for this methodology in a recent study, which is Alawajee et al., who applied a Nonlinear ARDL (NARDL) model to investigate the asymmetric impact of investor sentiment on housing and stock market returns in Saudi Arabia [15]. Although their sentiment data was constructed using principal component analysis from traditional economic indicators, their application of NARDL illustrates the model's strength in capturing nuanced sentiment effects on market performance.

2.4 Summary

In summary, the literature that has been reviewed emphasizes the significance of sentiment analysis in stock trend prediction. It demonstrates that predictive accuracy may be improved because of integrating different artificial intelligence (AI) techniques with various data sources, such as news, tweets, and stock data. While traditional models like dictionary-based sentiment analysis and basic machine learning techniques offer foundational insights, they have poor performance when capturing contextual nuances in financial language. Large Language Models (LLMs) present an opportunity

to enhance sentiment extraction through deeper linguistic understanding, such as ChatGPT. However, predictive accuracy is insufficient without a robust framework to model the time effects of sentiment on stock data. The study suggested the use of sentiment analysis and ARDL regression to more precisely predict short-term stock movements.

This is the reason we integrate sentiment analysis with the Autoregressive Distributed Lag (ARDL) regression model. The ARDL model enables the modeling of delayed sentiment effects, capturing how sentiment from previous days may continue to influence stock behavior. This directly aligns with the project's objective, leveraging ChatGPT for sentiment classification and applying ARDL to predict short-term stock trends. The proposed solution aims to produce a more comprehensive and interpretable model that addresses both the limitations and opportunities highlighted throughout the literature review by combining modern NLP with established econometric techniques.

CHAPTER 3

SYSTEM METHODOLOGY/APPROACH

The methodology of this project is designed to develop a web-based forecasting system that predicts short-term stock trends using sentiment analysis. All of the techniques that have been implemented in this project will be clarify in this chapter.

3.1 System Overview

This research adopts a mixed-method research design that integrates quantitative time series analysis of stock data with sentiment analysis of news. The objective is to examine how news sentiment influences short-term stock trends and understand the interactions between financial news and investors' behavior. The following research processes are designed to achieve this aim.

Figure 3.2 presents an overview of the proposed methodology. The study begins with data collection, where news and stock data are automatically retrieved using web scraping techniques in Python. After that, the raw data retrieved in the previous stage will go through the pre-processing stage to convert it into a format suitable for analysis. Afterward, sentiment analysis will be conducted on the news data after conversion, where the news are evaluate sentiments using ChatGPT (-1 to 1). Subsequently, appropriate statistical methods are used to analyze the relationship between news sentiment and stock price movements. Finally, the ARDL regression model will be implemented to forecast the short-term trends of stock prices based on the relationships.

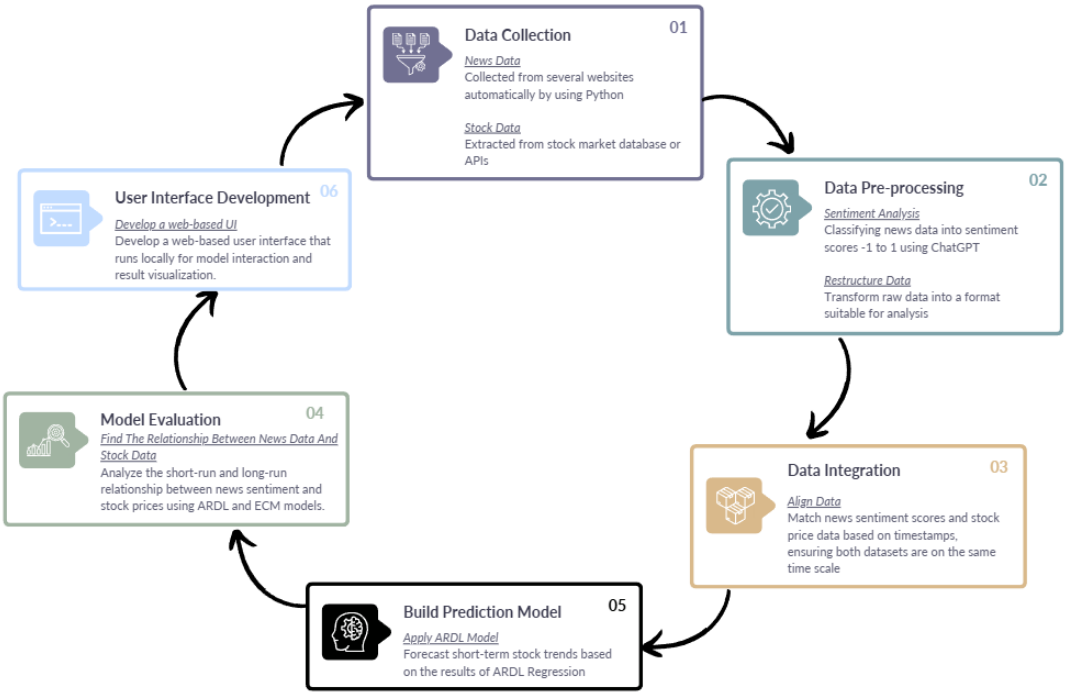


Figure 3.1 The process of developing a stock prediction model

3.2 Data Collection

The data collection is done using Python programming to automate the data collection process of news and stock market data. We gather news from a number of trusted financial news sources for 2 years (The Star, i3Investor, Yahoo Finance, Free Malaysia Today, etc.) and stock market data from stock market databases or APIs, which are open source. Data extraction rules and API calls allow the data collection process to be efficient, consistent and scalable. The data is extracted within a time period to make it relevant and to be able to compare news sentiment and short-term stock performance.

The news data is financial news stories about the chosen Malaysian banks. We have used The Star, Free Malaysia Today, i3Investor and Yahoo Finance. This was because they offer publicly available financial/market-related information and they cover a variety of news. The Star and Free Malaysia Today offer general financial and business news, i3Investor offers market discussions and company news from an investor's perspective, and Yahoo Finance offers financial market and stock information.

The stock data set includes the closing prices of the banks, primarily Maybank and Public Bank. Closing price is our main financial variable because it is the final price at which a stock is traded for the day. The aim is to forecast the closing price of the next day based on the current and past days' information. Other price variables (such as open, high, low, volume) might be helpful in future studies but have been excluded for this project in order to preserve a link between sentiment and the target variable.

3.3 Data Pre-processing

Pre-processing of data is needed because the scraped news and financial time-series data are generally noisy, have missing values, different formats, duplicate records and gaps during non-trading days. The pre-processing process guarantees the final modelling data is consistent, meaningful and ready for ARDL modelling.

For the scraped news data, pre-processing consists of filtering out duplicate news, standardising the date format, validating that the article content is not missing, and that the news article is linked to the right source. The date is converted to a daily basis that would allow the sentiment score to be consolidated and matched with daily stock prices. The text content is cleaned before sentiment scoring. This involves removing extra spaces, formatting characters and others that do not have financial information. The cleaning of the text is not about simplifying the text, as the GPT models need this information for sentiment analysis. Rather, the purpose of text cleaning is to remove noise that may mislead the sentiment analysis but retain the financial information of the article.

The news are analysed for sentiment by different GPT-5 models (GPT-5, GPT-5-mini, GPT-5-nano). GPT-5 models are used to assess the sentiment of each news data and classify them into three classes (positive, negative and neutral). These GPT-5 models gives a sentiment score of each news from -1.00 (negative) to 1.00 (positive) and a sentiment score of 0.00 would indicate a neutral sentiment for the whole data. Pang and Lee highlight that sentiment can be represented using different types of scales, including categorical labels and numerical ratings, depending on the task requirements [16]. While Pang and Lee (2008) discuss numerical sentiment scales such as 1–10, they do not enforce a specific range. The key requirement is that sentiment should be represented in a way that preserves relative polarity and intensity. In this project, a -1 to 1 scale is used instead because it provides a symmetric and centered representation,

where 0 clearly represents neutral sentiment. This is particularly suitable for regression models like ARDL, where having a zero-centered variable improves interpretability and stability. This is superior to only applying machine learning because it is able to integrate the context of the articles and subtle hints. It can help to gain a better understanding of the news' impact on investors.

For stock price data the date column is re-formate and sorted. The data are checked for missing trading-day values for stock markets, which are not operating on weekends and public holidays. These non-trading days are not the same as missing observations due to data problems. The stock price value is treated with care by using other stock data, if available. The resulting stock data set has a uniform date field and closing price.

3.4 Data Integration

After the data has been pre-processed, the sentiment data and the stock data need to be synchronised for sentiment data and stock data to be assessed jointly. We need to make sure that both data sets are aligned in time by joining them on dates. We may need to gather the data again or merge at this point. By this, we ensure that each sentiment score can be paired up with the stock data and combine the two datasets into one dataset for the next stage analysis.

Next, the data is cleaned and daily sentiment values are created. When more than one news item is published for a bank on a day, the sentiment scores of these news items are combined into a daily sentiment value. This makes the sentiment variable to be at the same daily level as the stock closing prices. The result of the pre-processing stage is the creation of a dataset with date, stock closing price and daily sentiment score for various sources.

3.5 Stock Prediction

In this study, the stock prediction model is derived from Autoregressive Distributed Lag (ARDL) model, which takes into account the autoregressive nature of stock prices and the lagged impact of sentiment-related factors.

The equation of ARDL (Autoregressive Distributed Lag) regression model is

$$y_t = a_1 y_{t-1} + \dots a_j y_{t-j} + b_0 x_t + \dots b_k x_{t-k} + e_t$$

The same model can be written as

$$y_t = \sum_{i=1}^{j} a_i y_{t-i} + \sum_{l=0}^{k} b_l x_{t-l} + e_t$$

In the ARDL model, $y_t$represents the dependent variable at time t, which in this study refers to the stock closing price. The term y_{t-i}denotes the lagged values of the dependent variable, capturing the autoregressive behavior of stock prices. The variable x_t represents the independent variable, which corresponds to the sentiment score derived from news data, while x_{t-l} denotes its lagged values, capturing delayed sentiment effects.

The coefficients $a_i$measure the influence of past stock prices on the current value, whereas $b_l$represent the impact of current and past sentiment values on stock prices. The parameters $j$and $k$denote the number of lags included for the dependent and independent variables, respectively. Finally, e_t is the error term, which captures unexplained variations and is assumed to be white noise.

In this paper, we cast the forecasting problem as a one-step-ahead prediction problem. The response variable is defined as a one-day ahead shift of the closing price, so that information about the current and past days is used to forecast the closing price of the following day. The optimal orders (p, q) are selected via expanding-window cross-validation. The training set is expanded in multiple folds, and the models are validated on the next set of data. The average performance across folds is calculated for each lag combination and the best model is determined according to its predictive accuracy and quality. Once the optimal lag structure is identified, the final ARDL model is estimated using the training data set and validated using a walk-forward strategy. This involves re-estimating the model as the data set is expanded.

3.6 Model Evaluation

To further analyze the dynamic relationship between sentiment and stock price movements, the Error Correction Model (ECM) representation of the ARDL model is employed. The ECM is formulated as:

$$\Delta Y_t = \alpha_0 + \sum_{i=1}^{p-1} \beta_i \Delta Y_{t-i} + \sum_{j=1}^{q-1} \gamma_j \Delta X_{t-j} + \lambda(Y_{t-1} - \theta X_{t-1}) + \varepsilon_t$$

In the ECM formulation, ΔY_trepresents the first difference of the dependent variable, defined as $\Delta Y_t = Y_t - Y_{t-1}$, capturing short-run changes in stock prices.

Similarly, $\Delta Y_{t-i}$and ΔX_{t-j}denote lagged differences of the dependent and independent variables, respectively.

The coefficients $\beta_i$and $\gamma_j$measure the short-run dynamic effects of past changes in stock prices and sentiment on current price changes. The term$(Y_{t-1} - \theta X_{t-1})$ is an error correction term (ECT), it is derived from the long-run relation between stock prices and sentiment. It is a measure of the previous period's out of equilibrium. The adjustment coefficient, λ, measures the speed at which the equilibrium is restored. If the coefficient λ is negative and statistically significant, this implies the presence of a long-run relationship between sentiment and stock prices, and that the adjustment process takes place. In this analysis, the ECM is built up in several steps. First, we create lagged variables for the dependent variable and sentiment variables. Next, first differences are taken to represent short-run dynamics. Third, a long-run regression is run to obtain the long-run relationship, which is used to generate the error correction term. Finally, the ECM regression is estimated with the differenced variables and the error correction term.

Besides the ECM analysis, a lag significance analysis is also performed to assess the impact of sentiment with various lags. We examine the coefficients and significance of lagged sentiment variables, to test whether sentiment has a lagged or instantaneous effect on stock prices. This helps to understand the timing of sentiment effects, such as whether sentiment has an immediate or delayed impact on stock prices, and whether sentiment is a part of the long-run equilibrium.

Model performance is evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Akaike Information Criterion (AIC), defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$$

$$AIC = 2k - 2\ln(\mathcal{L})$$

In the evaluation metrics, $y$represents the actual observed value, while $\hat{y}$denotes the predicted value generated by the model. The variable $n$is the total number of observations in the evaluation set.

For AIC, k represents the number of estimated parameters in the model, and $\mathcal{L}$ denotes the maximized value of the likelihood function. The AIC metric balances model goodness-of-fit and complexity, penalizing models with a larger number of parameters to avoid overfitting.

RMSE and MAE evaluate the accuracy of the prediction, in which lower values are better. AIC measures the goodness-of-fit and complexity of a model, with lower values signifying a better model. Lag removal experiments are performed to test the effect of removing lag terms from the lag structure. A comparison of lag structures can be made using RMSE and MAE, and the optimal lag structure can be determined.

3.7 Web-Based User Interface Development

In the last stage, a user interface is created to tie all system components together. A local web-based framework is used to develop the interface, enabling interactive use of the model. The system offers multiple interactive capabilities, such as selecting banks, GPT models and news providers. The user interface allows users to display stock price, sentiment, and comparison charts like Actual vs Predicted.

Furthermore, the user interface provides the user with the ability to assess model performance (RMSE, MAE and AIC), next-day prediction, and dynamic experiments with lag removal. The ECM module is also provided, allowing users to examine the error correction term and its significance. This interactive interface turns the prediction model into an interactive system with both prediction and analysis features.

CHAPTER 4

SYSTEM DESIGN

4.1 Integrated Data Processing and Prediction Pipeline

The integrated data processing and prediction pipeline of the proposed stock trend forecasting system is illustrated in Figure 4.1. The diagram presents a high-level overview of how data flows through the system, from data acquisition to final prediction and visualization, rather than focusing solely on user interactions.

The process begins with the collection of stock price data and financial news from multiple sources. The collected data is then passed through a data preprocessing stage, where cleaning, alignment, and transformation are performed to ensure consistency across time-series and sentiment data.

Following preprocessing, sentiment analysis is conducted using GPT-based models, which assign a sentiment score to each news item. The processed sentiment data is then integrated with stock price data and used as input for the ARDL modeling component.

The ARDL model captures both the autoregressive nature of stock prices and the lagged effects of sentiment variables. The model is trained using an expanding-window cross-validation approach to determine the optimal lag structure. Model performance is evaluated using metrics such as RMSE, MAE, and AIC.

To further analyze the relationship between sentiment and stock prices, the Error Correction Model (ECM) is applied to examine both short-run dynamics and long-run equilibrium behavior. Additional analyses, including lag significance and lag removal experiments, are conducted to better understand the temporal impact of sentiment.

Finally, the results of the forecasting and analysis are presented through a graphical user interface (GUI), which allows users to visualize stock trends, compare model performance, and interpret sentiment effects.

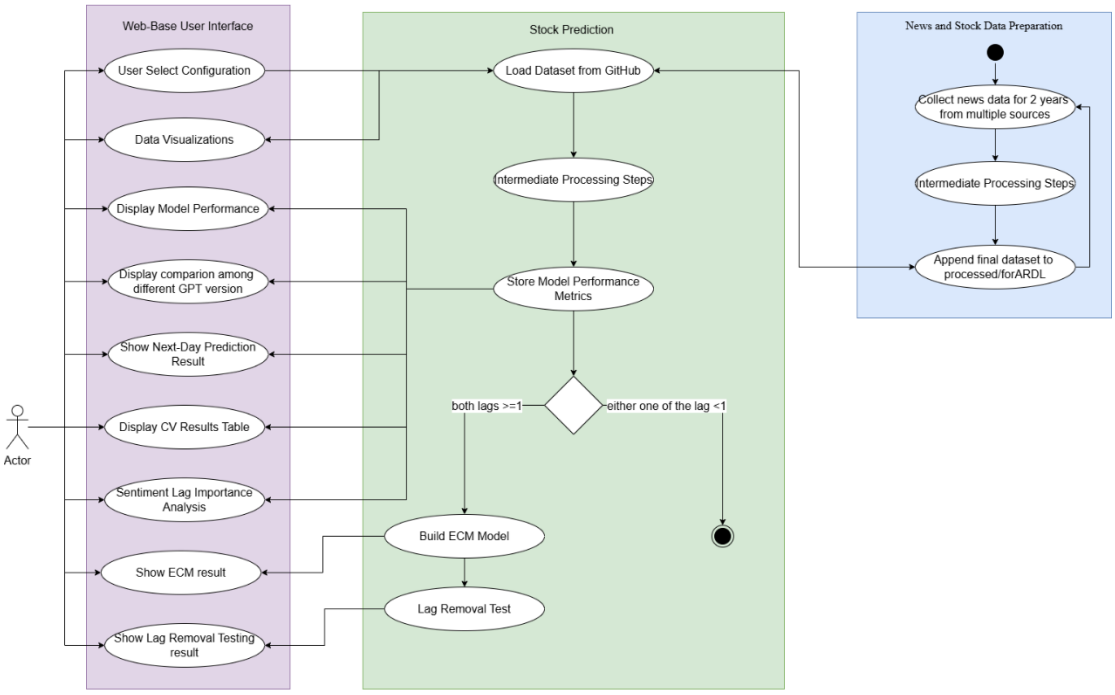


Figure 4.1 Integrated Data Processing and Prediction Pipeline for Sentiment-Based Stock Forecasting

4.2 System Components Specifications

The proposed system is organized into three main components, which is News and Stock Data Preparation, Stock Prediction and Web-Based User Interface. Each component corresponds to a specific part of the system workflow, as illustrated in the system block diagram, and performs a well-defined function within the overall forecasting pipeline.

4.2.1 News and Stock Data Preparation

The News and Stock Data Preparation is presented in Figure 4.2.1, the News and Stock Data Preparation is responsible for generating a ready-to-model data set for forecasting purpose. The module takes financial news and stock price data from various sources.

The system gathers around two years of financial news data from various sources, such as The Star, Free Malaysia Today, i3Investor and Yahoo Finance. These include varied financial news, investor-oriented comments and market-specific news. The news data is saved and appended to CSV files.

The news data is cleaned and merged to eliminate inconsistencies and standardise its format. The sentiment values are calculated using various GPT models and are then appended and averaged daily. This ensures that sentiment scores correspond to the stock price data's time scale.

Meanwhile, daily stock prices are also obtained, mainly from Yahoo Finance. Data is supplemented through additional data collection to address missing values. The sentiment and stock closing price datasets are then merged by date to create a single dataset. This process guarantees that stock data is matched with the corresponding sentiment data.

The last step is converting the dataset into a format suitable for further modeling by creating the target variable for stock price prediction. This results in a compact data set with stock price, sentiment and prediction target, which is saved for the Stock Prediction component.

Figure 4.2.1 Flowchart of News and Stock Data Preparation

4.2.2 Stock Prediction

Figure 4.2.2.2 shows the Stock Prediction process in details, the Stock Prediction component is the main module of the system which is used for stock price prediction and model analysis. It uses the Autoregressive Distributed Lag (ARDL) model to incorporate the temporal aspect of the stock prices and the lagged effects of sentiment variables.

The Stock Prediction process starts by loading the dataset from the News and Stock Data Preparation part. The dataset consists of aligned stock closing prices and aggregated sentiment scores, where sentiment values are derived from news data using different GPT models. Although this dataset is already prepared for modeling, but an additional data preparation step is conducted to clean the data. Figure 4.2.2.1 shows the missing stock closing price values in the final dataset, this may occur due to external data source limitations such as network fluctuation. Therefore, additional data validation and cleaning procedures are performed, including handling missing stock closing price values using external data sources (e.g., Yahoo Finance API), to ensure data completeness for time-series modeling.

	A	B	C	D	E	F
1	Date	Close	Sentiment_Free_Malaysia_Today	Sentiment_The_Star	Sentiment_Yahoo_Finance	Sentiment_i3Investor
2	4/24/2026	11.16	0	0	0	0
3	4/23/2026	11.22	0	0	0	0
4	4/22/2026	11.22	0	0	0	0.2
5	4/21/2026		0	0	0	0.125
6	4/20/2026	11.34	0	0	0	0.2
7	4/17/2026	11.1	0	0	0	0
8	4/16/2026	11.12	0	0	0	0
9	4/15/2026	11.06	0	0	0	0.46
10	4/14/2026	11.04	0	0	0	-0.2
11	4/13/2026	10.98	0	0	0	0.2
12	4/10/2026	11.16	0	0.74	0	-0.2
13	4/9/2026	11.16	0	0.2	0	0
14	4/8/2026	11.32	0	0	0	0.0667
15	4/7/2026	11.16	0	0.3	0	-0.35
16	4/6/2026	11.22	0	0.1	0	-0.2
17	4/3/2026	11.26	0	0	0	-0.35
18	4/2/2026	11.46	0	0.1	0	0.3667
19	4/1/2026	11.66	0	0	0	0.27
20	3/31/2026	11.36	0	0	0	0
21	3/30/2026	11.2	0	0.775	0	0.6
22	3/27/2026	11.46	0	0.7	0	0
23	3/26/2026	11.44	0	0	0	0.4
24	3/25/2026	11.44	0	0	0	0.325
25	3/24/2026	11.34	0	0	0	0.08
26	3/20/2026	11.6	0	0.66	0	-0.1
27	3/19/2026	11.6	0	-0.025	0	0.9
28	3/18/2026	11.74	0	0	0	0.2

gpt-5_And_Closing_Price

Figure 4.2.2.1 Missing Stock Closing Price Values Occur in final dataset

The dataset is split into training and testing sets using a time-series split, where the first 80% of observations are used for training and the remaining 20% for testing, ensuring that future information is not used in model training. The optimal lag orders (p, q) are selected using an expanding-window cross-validation method. This allows the model to be evaluated with increasing training data sizes, which enhances the model's generalisation performance.

After determining the best lag order, the ARDL model is refitted on the training data set. The model is then used to conduct walk-forward forecasting, in which forecasts are made in a rolling manner. This mimics real-life forecasting, where models are learned incremental to the incoming data.

The performance of forecasting is assessed using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Akaike Information Criterion (AIC). These measures offer a combination of an indicator of forecasting accuracy and quality, combining both model error and complexity.

Besides forecasting, this component also provides a number of analytical functions to improve interpretability. It permits to display performance of different GPT models to evaluate the effect of sentiment generation techniques. To support model comparison across different sentiment models, an intermediate result storage mechanism is implemented. The ARDL model results, including forecasting outputs and performance metrics, are stored after each model is trained. This allows the system to accumulate results from multiple sentiment models before conducting comparative analysis. It also supports sentiment lag importance analysis to assess the impact of sentiment at various lags, which helps to determine whether stock price response is instantaneous or lagged.

Also, if the selected lag structure meets the necessary criteria (p and q >=1), the Error Correction Model (ECM) is built to examine short-run and long-run relationships between sentiment and stock returns. The system also allows lag removal testing by removing some of the lag variables to check whether reduced model complexity can lead to better prediction accuracy. These analytical functions offer in-depth insights into the interaction between sentiment and stock prices, contributing to the robustness and interpretability of the model.

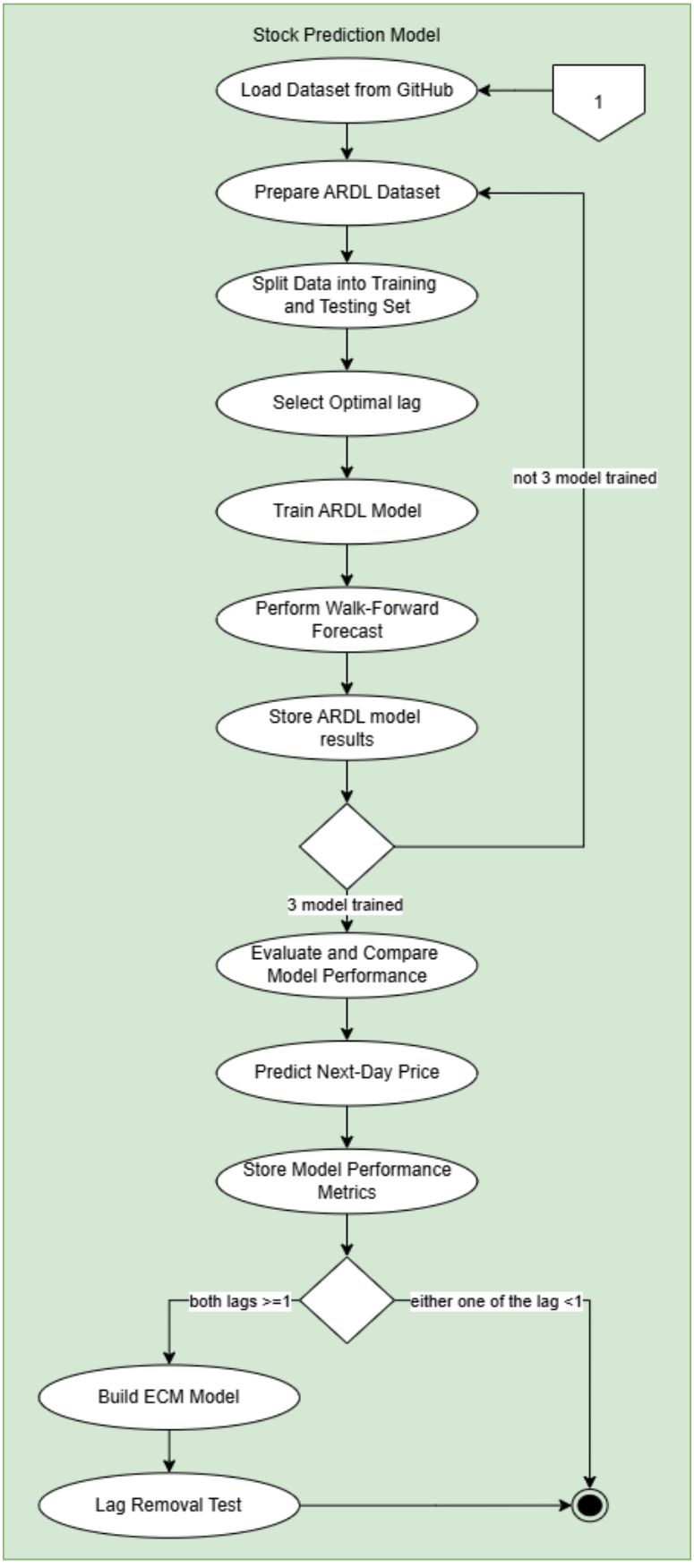


Figure 4.2.2.2 Flowchart of Stock Prediction

4.2.3 Web-Based User Interface

The functions of Web-Based User Interface are as shown in Figure 4.2.3, the user interface offers an interactive platform, allowing users to set up the stock price forecasting model and display the analysis results. Users can choose the bank they are interested in, the GPT model for generating sentiment, and the news source. They define the data set and the modeling process undertaken by the system. This customization approach enables the system to perform experiments under varying conditions.

The interface also supports a comprehensive set of visualization and analytical functions. It shows the trends of stock price and sentiment over time, and integrated views that demonstrate the interaction between sentiment and stock price. Furthermore, it shows results of stock price forecasting by plotting the actual and forecasted stock prices.

The interface also presents the results of quantitative assessments, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Akaike Information Criterion (AIC). Additionally, it allows the analysis of multiple GPT models, to illustrate the impact of different sentiments on forecasting outcomes.

In addition, the interface displays the results of lag selection, which are obtained using cross-validation (CV), to indicate how the best ARDL models are chosen. It also presents sentiment lag importance analysis, which assesses the importance and statistical significance of sentiment lags.

For more advanced analysis, the system also presents the results of the Error Correction Model (ECM), including the coefficient and significance of the error correction term, to reveal the existence of long-run equilibrium relationships. It also displays the results of lag removal tests, which assess the impact of changing the lag structure on the model's performance.

In summary, the Web-Based User Interface brings together all the components of the system and provides an interactive display of the results. This allows users to better understand the forecasting results and better understand how sentiment influences stock returns.

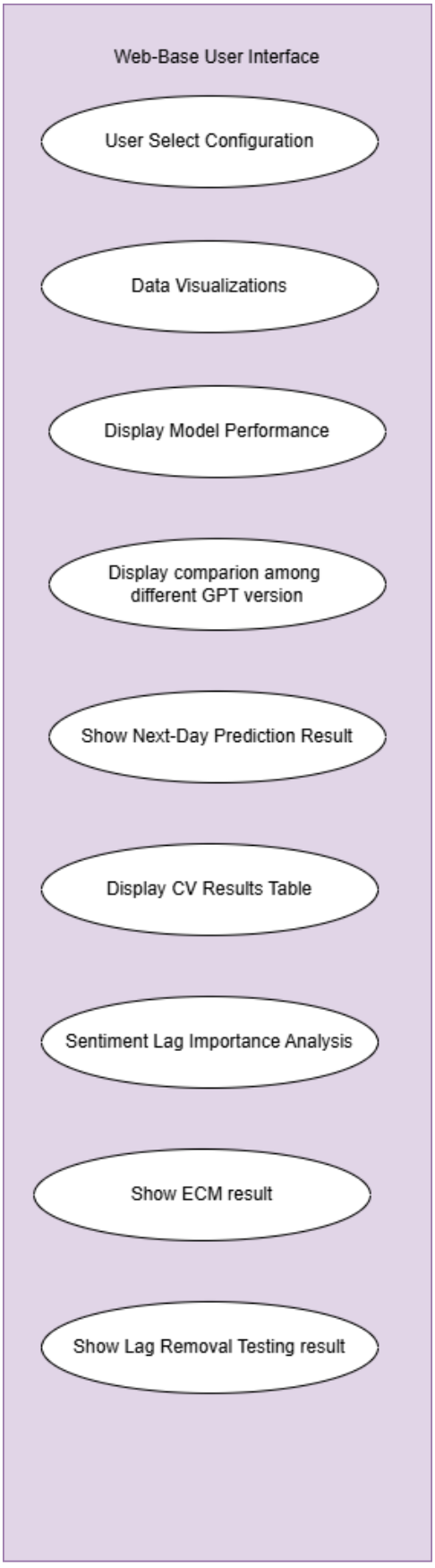


Figure 4.2.3 Web-Based User Interface Functional Components Diagram

4.3 System Components Interaction Operations

The system components interact in a specified sequence of steps, as shown in the use case-driven diagram.

It all starts with the user interface, where the user chooses the parameters. The parameters include which bank to focus on, which GPT model to use for sentiment analysis and which news source to use. Depending on the user's choice, the system retrieves the relevant dataset from the data source component. This data is then fed into the data processing component, which cleans, aligns, and pre-processes the data in a way that is ready for modeling. The cleaned data is then sent to the ARDL modeling component.

In the modeling component, the system begins by preparing the data by creating lag variables and the dependent variable for prediction of the next day. Next, the data is divided into training and testing sets of data. Cross-validation is used to determine the lag order of the ARDL(p, q) model. Once the lag order is selected, the ARDL model is fitted. Walk-forward forecasting is then applied to provide forecasts for the test data. Model performance is assessed by comparing the predictions with the real values.

The evaluation and analysis module processes the forecasting output to calculate the root mean square error (RMSE), mean absolute error (MAE) and AIC. It also conducts further analyses, such as comparison of different models, importance of sentiment lags, and ECM analysis. If the user chooses to remove lags, the system automatically adjusts the lag structure and re-computes the model performance, enabling comparison of the two models.

Lastly, the results are displayed on the user interface using different plots, tables, and other performance metrics. This allows users to understand the forecasting results and how sentiments and stock prices are related.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Hardware Setup

The hardware used in this project is a personal laptop, as shown in Table 5.1. The system is developed and executed in a specific environment to support data collection, preprocessing, and model implementation.

The laptop is equipped with an Intel Core i5-12500H processor, 8GB of RAM, and sufficient storage capacity. This configuration is adequate for handling data processing tasks and time series modeling, including ARDL regression and forecasting operations. The sentiment analysis is performed using external GPT APIs rather than locally trained models. Therefore, the system can be effectively implemented using commonly available hardware without the need for high-performance computing resources.

Table 5.1 Specifications of laptop

Description	Specifications
Model	Acer Aspier N515-58
Processor	Intel Core i5-12500H
Operating System	Windows 11
Graphic	Intel UHD Graphics, NVIDIA GeForce RTX 3060 Laptop GPU
Memory	8GB RAM
Storage	475GB

5.2 System Implementation

This section outlines the overall implementation process of the proposed system, its cover the process from raw data collection to structured dataset preparation. The implementation is divided into three main parts: data collection, data pre-processing, and data integration.

5.2.1 News and Stock Data Preparation

5.2.1.1 Data Collection

This stage involves gathering raw datasets required for the project, including financial news articles and historical stock price data. Web scraping techniques and automated extraction scripts were employed to retrieve data from multiple online sources, ensuring both breadth and relevance. In addition, other scripts might be developed to scrape news from other sources in FYP2, such as Maybank's official website.

i. News Data Extraction

The code below calls several functions: theStar(), i3Investor(), yahooFinance(), and freeMalaysiaToday(). Since each website has a different HTML structure, a separate web scraping script is required for each one. The extracted data is cleaned and standardized (for example, dates are converted into a consistent format like YYYY-MM-DD). Finally, each function saves the results into a separate CSV file (e.g., theStar() generates TheStarNews.csv, i3Investor() generates I3InvestorNews.csv, etc.), so that the scraped news data can be reused later for sentiment analysis or further processing.

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
from datetime import datetime, timedelta
import pandas as pd
import time
import requests
import re

# Set up Chrome options
options = Options()
options.add_argument("--headless")  # Run in background
options.add_argument("--disable-gpu")
options.add_argument("--window-size=1920,1080")
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64)")

theStar()
i3Investor()
yahooFinance()
freeMalaysiaToday()
```

Figure 5.2.1 Python script of news data retrieval

The next section of code is responsible for merging multiple news CSV files into a single dataset. Firstly, it searches the current directory for all CSV files that end with “News.csv” and prints all of them. Each file corresponds to news articles scraped from a different website. The function infer_source_from_filename() automatically assigns a news source label (e.g., “The Star”, “i3Investor”, “Free Malaysia Today”, or “Yahoo Finance”) based on the filename. This ensures that after merging, you can still identify where each article came from.

After that, all of the CSV files are read into Pandas DataFrames and stored in a list. These DataFrames are concatenated into a single merged DataFrame. Next, the script performs data cleaning. It drops any records where either the date or the news content is missing (NaN), removes duplicate articles, and sorts the articles by date in descending order (latest → oldest) and resets the index. Finally, the cleaned and merged dataset is saved into a new file named NewsData.csv.

```python
import os
import pandas as pd
import glob
from dateutil import parser
import csv

# Find all CSV files ending with "News"
csv_files = glob.glob("*News.csv")

print("News csv: ", csv_files)

def infer_source_from_filename(filename):
    fname = os.path.basename(filename).lower()
    if "freemalaysiatoday" in fname:
        return "Free Malaysia Today"
    if "i3investor" in fname:
        return "i3Investor"
    if "thestar" in fname:
        return "The Star"
    if "yahoofinance" in fname:
        return "Yahoo Finance"
    return "Unknown"

# read all records in csv files
df_list = []

for f in csv_files:
    df = pd.read_csv(f, encoding="utf-8")
    source = infer_source_from_filename(f)
    df["Source"] = source
    df_list.append(df)

# merge all records in csv files
merged_df = pd.concat(df_list, ignore_index=True)

# Drop the records which either date or news is NaN value
merged_df = merged_df.dropna(subset=["Date", "News"])

# Drop the duplicate records and sort by date descending
merged_df = merged_df.drop_duplicates()
merged_df = merged_df.sort_values(by="Date", ascending=False).reset_index(drop=True)

# save as new csv file
merged_df.to_csv("NewsData.csv", index=False, encoding="utf-8")
# observe NewsData.csv, some of the records will use many columns, because the news contain too many comma, csv file is comma-separated values

print("Merge Complete, Total News:", len(merged_df))
print(merged_df)
```

Figure 5.2.2 Python script of merge news data

ii. Stocks Data Extraction

This section of code is designed to scrape and store historical stock data for Maybank (1155.KL) from Yahoo Finance. The script navigates to the Yahoo Finance page containing Maybank's historical stock data and pauses for a while to ensure that all JavaScript elements are fully rendered before proceeding. Once the page is ready, BeautifulSoup is used to parse the HTML and identify the table that contains the stock price information. To keep the dataset relevant, the script applies a filter to include only records from the past 365 days, discarding any older entries.

The extracted values are then cleaned and standardized. The commas within numeric values are removed, missing values represented by a dash are replaced with NaN, and all numerical fields are converted into floating-point format to enable proper calculations. The Date column is also converted into a standardized datetime format, preparing for future work. Finally, the cleaned and structured stock data is stored in a CSV file named *Maybank_Stocks_Data.csv*.

```python
# Parse with BeautifulSoup
soup = BeautifulSoup(html, "html.parser")
table = soup.find("table", class_="table yf-1jecxey noDl hideOnPrint")

# Prepare time range (last 365 days)
today = datetime.today()
threshold = today - timedelta(days=365)

# Check if table exists
if table:
    rows = table.find_all("tr")
    data = []

    for row in rows[1:]:  # Skip header
        cols = row.find_all("td")
        if len(cols) != 7:
            continue  # skip if not a complete row (maybe shared dividend)

        date_text = cols[0].text.strip()

        try:
            date_obj = datetime.strptime(date_text, "%b %d, %Y")
        except ValueError:
            continue  # if not a valid date (e.g., dividends)

        if date_obj < threshold:
            break  # Stop loop if data is older than 30 days

        row_data = [col.text.strip() for col in cols]
        data.append(row_data)

    # Define headers
    columns = ["Date", "Open", "High", "Low", "Close", "Adj Close", "Volume"]
    df = pd.DataFrame(data, columns=columns)

    # Clean numeric columns (column volume)
    for col in columns[1:]:
        df[col] = (
            df[col]
            .str.replace(",", "", regex=False)  # remove comma
            .replace("-", np.nan)               # replace "-" to NaN
            .astype(float)                      # convert into float type
        )

    # Convert Date to datetime
    df["Date"] = pd.to_datetime(df["Date"])

    # Show top 5 results
    print(df.head())

else:
    print("Table not found.")

df.to_csv("Maybank_Stocks_Data.csv", index=False)
```

Figure 5.2.3 Python script of stocks data retrieval

The code below is used to visualize Maybank's stock closing price over the past year. The script reads the stock data from the CSV file *Maybank_Stocks_Data.csv* and ensures that the "Date" column is correctly converted into a datetime format so that it can be properly plotted on the x-axis. The script creates a line graph that plots the "Date" against the "Close" values, representing the stock's daily closing price using matplotlib. A line is drawn to connect all these points, to show how the stock price fluctuates over time. The graph is labeled with a title, axis labels, and a grid to improve readability. Finally, the chart is displayed, allowing for a straightforward visualization of Maybank's stock performance across the last 12 months.

```
import pandas as pd
import matplotlib.pyplot as plt
import mplfinance as mpf

# Load Maybank_Stocks_Data.csv
df = pd.read_csv("Maybank_Stocks_Data.csv")

# Convert Date to datetime
df["Date"] = pd.to_datetime(df["Date"])

# Plot the Maybank Stock Closing Price Graph for the Last 1 Year
plt.figure(figsize=(10, 5))
plt.plot(df["Date"], df["Close"], linestyle="-")
plt.title("Maybank Stock Closing Price (Last 1 Year)")
plt.xlabel("Date")
plt.ylabel("Closing Price (MYR)")
plt.grid(True)
plt.show()
```

Figure 5.2.4 Python script of stocks data visualization

5.2.1.2 Data Pre-Processing

The raw data are go through pre-processing to improve its quality and usability after collection. This includes cleaning, formatting, and transforming data into a consistent structure. For news articles, sentiment analysis was performed using different GPT models, while for stock data, missing values were handled and date formats standardized.

i. Sentiment Analysis

The code below is designed to perform sentiment analysis on financial news articles about Maybank using OpenAI's GPT models. It starts by importing the required libraries such as pandas, OpenAI, time and etc. The script loads the dataset of news articles from newsData.csv and initializes the OpenAI client with an API key.

After that, a function named gpt_5_sentiment is defined. This function constructs a prompt that instructs GPT to act as a financial analyst and assign a sentiment score ranging from -1.00 (highly negative) to 1.00 (highly positive), with 0.00 representing neutral. The function sends this prompt to GPT-5 and retrieves the sentiment score, while handling errors by returning None if the request fails. The script iterates through every article in the dataset, applies the sentiment analysis function, appends the resulting scores to a list, and prints the progress of processing each record. The delay of 3 seconds between API calls is to avoid exceeding rate limits. Finally, the sentiment scores are stored in a new column of the DataFrame, and saved as gpt-5_sentiment.csv. This section of code applies to other models also, such as gpt-5-mini and gpt-5-nano.

```python
import pandas as pd
from openai import OpenAI
import time
import re

# load the news data
df = pd.read_csv("newsData.csv")

# initialize OpenAI client
API=
client = OpenAI(api_key = API)

# define a function to get sentiment score by using gpt-5
def gpt_5_sentiment(news_text):
    prompt = f"""You are a financial analyst analyzing the news sentiment of Maybank.
{news_text}
Review this news article and respond to its sentiment score (-1 to 1, 2 decimal places). For
example, respond `-1.00` when the article has a highly negative impact on the company,
respond `0.00` when the article is neutral or irrelevant, respond `1.00` when the article has
a highly positive impact on the company. You can have any number in between depending
on its sentiment. Respond with only 1 number and do not include any other words."""

    try:
        response = client.responses.create(
            model="gpt-5",
            input=[{"role": "user", "content": prompt}],
            reasoning={
                "effort": "minimal"
            }
        )
        return response.output[1].content[0].text
    except Exception as e:
        print("Error:", e)
        return None

# iterate over every record, and add the sentiment score
sentiments = []
for idx, row in df.iterrows():
    score = gpt_5_sentiment(row["News"])
    sentiments.append(score)
    print(f"Processed {idx+1}/{len(df)} -> {score}")
    time.sleep(3)  # delay the access speed to gpt to avoid rate limit

df["Sentiment"] = sentiments

# save into gpt-5_sentiment.csv
df.to_csv("gpt-5_sentiment.csv", index=False)
print("Sentiment analysis done, saved to gpt-5_sentiment.csv")
```

Figure 5.2.5 Python script of calling GPT API to perform sentiment analysis

The code below processes the sentiment analysis results stored in gpt-5_sentiment.csv to compute the daily average sentiment score of news data. The results are rounded to four decimal places for precision and readability. After that, the DataFrame is sorted in descending order by date so that the most recent average sentiment appears first. Finally, the processed dataset is saved as gpt-5_avg_sentiment.csv, and both a confirmation message and the resulting DataFrame are printed to the console. This step applies to other models also, such as gpt-5-mini and gpt-5-nano.

```python
import pandas as pd

# load gpt-5_sentiment.csv
df_sentiment = pd.read_csv("gpt-5_sentiment.csv")

# calculate the average sentiment score by date
df_avg = df_sentiment.groupby("Date", as_index=False)["Sentiment"].mean()

# Rounded to 4 decimal places
df_avg["Sentiment"] = df_avg["Sentiment"].round(4)

# sort by date descending
df_avg = df_avg.sort_values(by="Date", ascending=False)

# save to gpt-5_avg_sentiment.csv
df_avg.to_csv("gpt-5_avg_sentiment.csv", index=False)

print("Average sentiment was computed, and stored in gpt-5_avg_sentiment.csv")
print(df_avg)
```

Figure 5.2.6 Python script for calculating average sentiment scores

5.2.1.3 Data Integration

The final step integrates the sentiment scores with stock market data into a single dataset. By aligning both datasets based on date to enable further analysis and the development of predictive models.

i. Data Alignment

This code integrates Maybank's stock data with the average sentiment scores generated from news articles. It loads the stock data (Maybank_Stocks_Data.csv) and the sentiment data (gpt-5_avg_sentiment.csv) into two DataFrames. The Date columns in both DataFrames are converted into proper datetime objects to ensure accurate merging. From the stock dataset, only the Date and Close (closing price) columns are retained, as these are the relevant features for analysis. A left join is then performed to merge the sentiment scores with the stock closing prices based on matching dates, ensuring that all sentiment records are retained even if stock data for a given date is unavailable. After merging, the data is sorted in descending order by date, and the row index is reset. To handle missing values in the stock prices, forward fill (ffill) is applied so that the previous available closing price is used whenever data is missing. Finally, the cleaned and merged dataset is saved as gpt-5_Sentiment_And_Closing_Price.csv. The same process is used for gpt-5-mini and gpt-5-nano models also.

```python
import pandas as pd

# Load Maybank_Stocks_Data.csv and gpt-5_avg_sentiment.csv
stocks = pd.read_csv("Maybank_Stocks_Data.csv")
sentiment = pd.read_csv("gpt-5_avg_sentiment.csv")

# Convert Date format
stocks["Date"] = pd.to_datetime(stocks["Date"])
sentiment["Date"] = pd.to_datetime(sentiment["Date"])

# Keep only the date and closing price
stocks_close = stocks[["Date", "Close"]]

# Merge sentiment score and closing price
merged_df = pd.merge(sentiment, stocks_close, on="Date", how="left") # how="left" means left join

# Sort the date by descending order and resets the row index back to 0, 1, 2, … after sorting, and drops the old index
merged_df = merged_df.sort_values("Date", ascending=False).reset_index(drop=True)

# Fills any missing values (NaN) by copying the value from the previous row
merged_df["Close"] = merged_df["Close"].ffill()

# save to gpt-5_Sentiment_And_Closing_Price.csv
merged_df.to_csv("gpt-5_Sentiment_And_Closing_Price.csv", index=False)

print("Merge Complete! Result save as gpt-5_Sentiment_And_Closing_Price.csv")
```

Figure 5.2.7 Python script of aligning sentiment scores and closing price

5.2.1.4 Github

Figure 5.2.8 shows the Github repository of this project, Github is used as a data storage repository to store data, code, and workflow configurations. The repository is used to execute the data preparation code periodically, and store all data generated, including initial news data, sentiment analyses, stock price data, and the final ARDL datasets. This method of data storage guarantees that the entire system, including the Stock Prediction and Web-Based User Interface, has access to the same data.

The process of updating the repository is integrated into the automated workflow. Once the data collection, preprocessing and align steps are run, the new datasets are added and pushed to the GitHub repository. This means that the repository is always up-to-date with the latest data pipeline status without needing separate uploads.

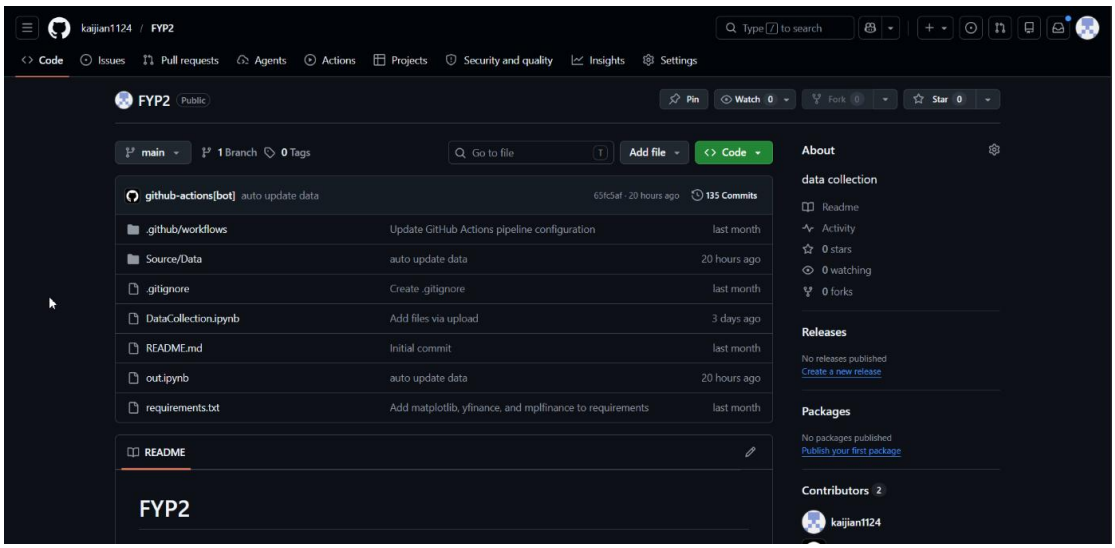


Figure 5.2.8 Github Repository

Figure 5.2.9 shows the OpenAI API key that used in this project, API key is required to securely access the GPT API for sentiment analysis. Instead of hard-coding the API key within the source code, the system uses environment variables to manage sensitive credentials. The API key is stored as a secret within the GitHub repository using the GitHub Actions secrets configuration.

During workflow execution, the API key is retrieved from the repository's secret storage and injected into the execution environment. This ensures that sensitive details are not exposed in the source code and access to external services is secure during automated workflows.

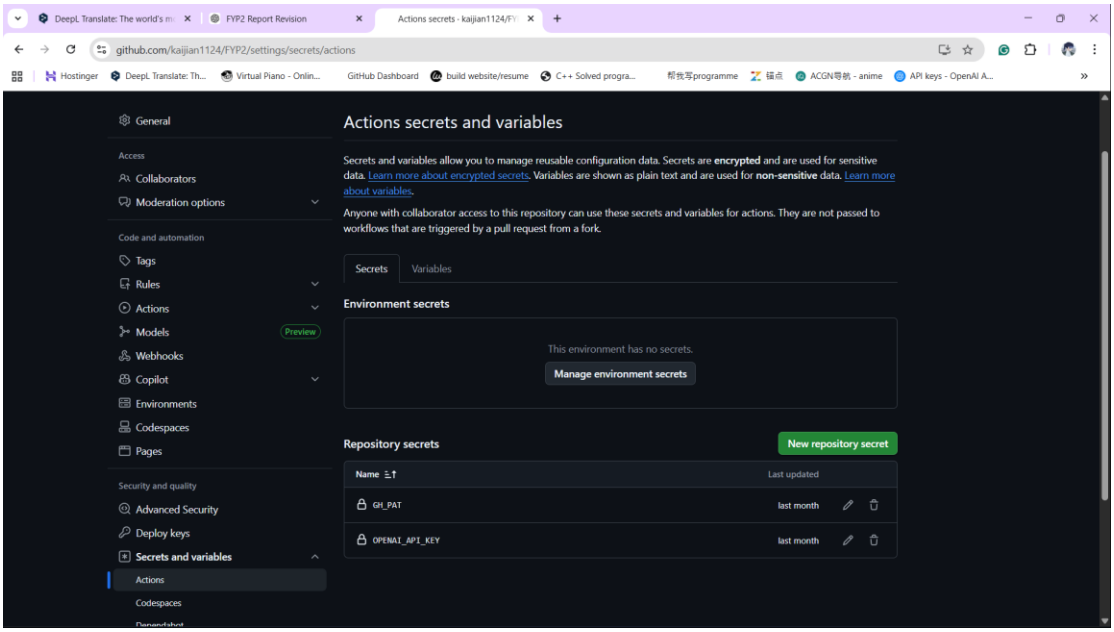


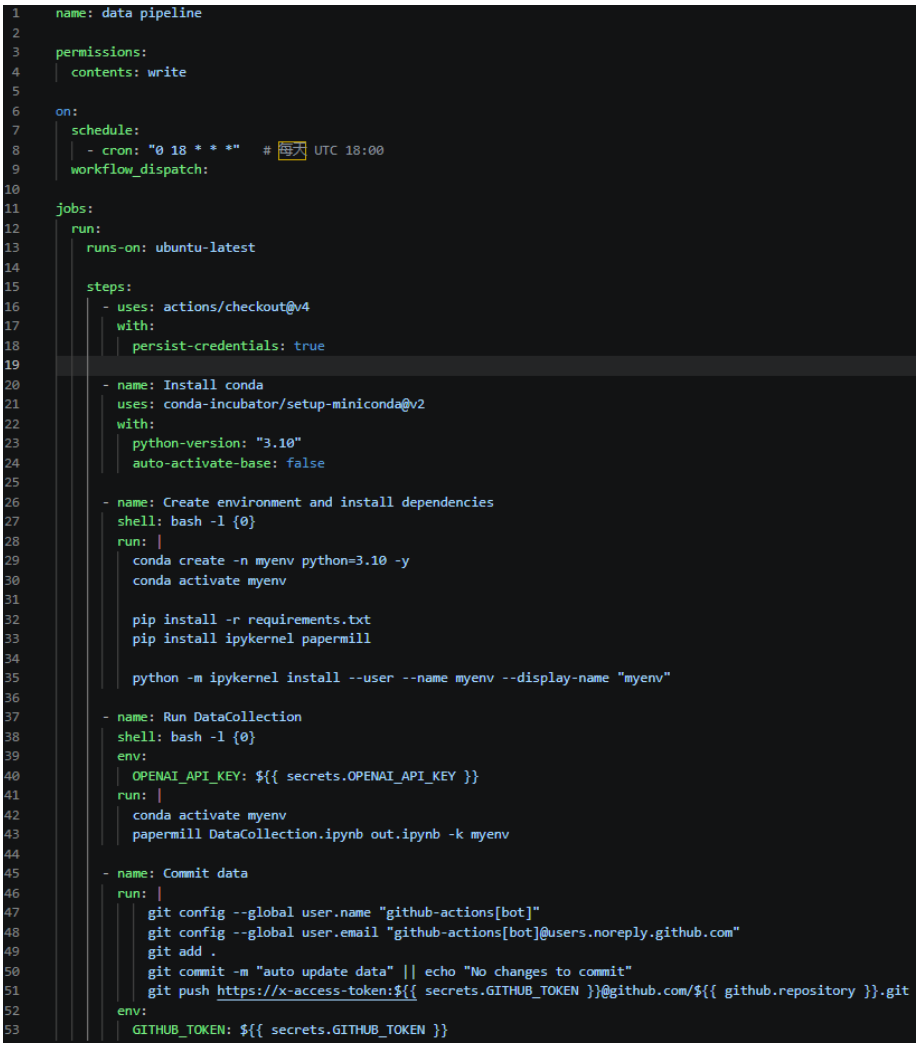
Figure 5.2.9 Secrets key configure in Github

The automation of the data preparation process is implemented using GitHub Actions in combination with Papermill. GitHub Actions offers a cloud platform for running jobs, which can be scheduled or triggered manually. The DataCollection.ipynb notebook is run in a reproducible and parameterized way using Papermill. The details of the workflow are specified in the pipeline.yml file within the .github/workflows directory. The workflow runs various tasks in sequence, such as cloning the repository, setting up the Python environment, installing dependencies and running the notebook with Papermill. The execution of the notebook produces new datasets, which are committed to the repository. This combination allows for a completely automated data workflow, with data collection, sentiment generation, preprocessing and dataset updating done automatically. This allows for an ever-evolving dataset to be used as a source for forecasting and analysis.

5.2.1.5 Papermill

In addition to the notebook implementation, GitHub is used as the central storage platform for the project files and generated datasets. The repository stores the raw data, processed sentiment files, stock data, final ARDL datasets, notebook files, Python scripts, and workflow configuration files. This allows the Web-Based User Interface and Stock Prediction to retrieve updated datasets directly from the repository.

The automated execution of the data preparation notebook is handled using GitHub Actions and Papermill. GitHub Actions provides the scheduled automation environment, while Papermill executes the Jupyter Notebook in a reproducible manner. The workflow file is stored under .github/workflows/pipeline.yml. During each scheduled execution, the workflow installs the required environment, runs the notebook using Papermill, and updates the repository with newly generated data files.



```yaml
name: data pipeline

permissions:
  contents: write

on:
  schedule:
    - cron: "0 18 * * *"   # 每天 UTC 18:00
  workflow_dispatch:

jobs:
  run:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v4
        with:
          persist-credentials: true

      - name: Install conda
        uses: conda-incubator/setup-miniconda@v2
        with:
          python-version: "3.10"
          auto-activate-base: false

      - name: Create environment and install dependencies
        shell: bash -l {0}
        run: |
          conda create -n myenv python=3.10 -y
          conda activate myenv

          pip install -r requirements.txt
          pip install ipykernel papermill

          python -m ipykernel install --user --name myenv --display-name "myenv"

      - name: Run DataCollection
        shell: bash -l {0}
        env:
          OPENAI_API_KEY: ${{ secrets.OPENAI_API_KEY }}
        run: |
          conda activate myenv
          papermill DataCollection.ipynb out.ipynb -k myenv

      - name: Commit data
        run: |
            git config --global user.name "github-actions[bot]"
            git config --global user.email "github-actions[bot]@users.noreply.github.com"
            git add .
            git commit -m "auto update data" || echo "No changes to commit"
            git push https://x-access-token:${{ secrets.GITHUB_TOKEN }}@github.com/${{ github.repository }}.git
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

Figure 5.2.10 The Content of Pipeline.yml

5.2.2 Stock Prediction

The Stock Prediction is implemented in Python using the ardl.py script. This module is responsible for preparing the ARDL modeling dataset, selecting the optimal lag order, training the ARDL model, generating forecasts, evaluating prediction performance, and conducting additional model analysis such as ECM and lag removal testing.

The ARDL modeling, evaluation, and analysis component is developed in a local Python environment using Visual Studio Code (VS Code). Python version 3.11.x is used because it is compatible with the required data science, statistical modeling, and visualization libraries. The development environment is supported by the Python, Pylance, and Jupyter extensions in VS Code. These extensions assist with code execution, debugging, syntax checking, and notebook-based experimentation during model development.

5.2.2.1 VSCode

The ARDL modeling, evaluation, and analysis component is implemented using Python in a local development environment. The system uses Python version 3.11.x, which provides compatibility with required data science and statistical libraries.

The development environment is set up using Visual Studio Code (VS Code) with the following extensions:

- Python
- Pylance
- Jupyter

These extensions support code execution, debugging, and interactive notebook operations.

5.2.2.2 Virtual Environment Setup

A virtual environment is created to manage project dependencies and ensure isolation from other Python projects. The following commands are executed in the VS Code terminal:

```
python -m venv .venv
.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
pip install streamlit pandas matplotlib numpy scikit-learn statsmodels
```

The installed libraries support different functions in the Stock Prediction. Pandas and NumPy are used for data manipulation and numerical operations. Statsmodels is used to implement the ARDL model and OLS-based ECM analysis. Scikit-learn is used to calculate evaluation metrics such as MAE and RMSE. Yfinance is used as a supplementary source to retrieve missing stock closing prices when required.

5.2.2.3 Main Workflow of Stock Prediction

The main workflow of the Stock Prediction is controlled by the run_full_ardl_pipeline() function which is shown as figure below. This function connects all major modelling steps into a single pipeline. It receives the dataset as input, validates the date and closing price columns, handles missing values when necessary, prepares the ARDL dataset, splits the data into training and testing sets, selects the best lag order, trains the ARDL model, performs walk-forward forecasting, calculates evaluation metrics, and produces a next-day prediction.

Before the ARDL model is trained, the dataset must be converted into a modeling-ready format. This step is implemented using the prepare_ardl_data() function. The function identifies sentiment columns, selects the required variables, sorts records by date, removes incomplete records, and constructs the next-day closing price as the prediction target. The target variable is created by shifting the closing price one step forward, which allows the model to predict the next trading day's closing price using current and historical information.

After the ARDL dataset is prepared, the data is divided into training and testing sets using a time-series split. Unlike random splitting, this approach preserves the chronological order of the data, which is necessary for stock forecasting because future observations must not be used to train models for past predictions.

The optimal ARDL lag order is selected using the get_best_order() function. This function evaluates different combinations of autoregressive lag p and sentiment lag q. In the current implementation, the search range is from 0 to 5 for both p and q. For each lag combination, an expanding-window cross-validation process is applied. The training window increases progressively, while the next portion of the data is used as the validation set. This approach is suitable for time-series data because it maintains the temporal sequence of observations.

For every valid ARDL(p, q) combination, the model is fitted using the training fold and evaluated on the validation fold. The system calculates MAE, RMSE, and AIC for each fold. The average values across all folds are then stored in the cross-validation result table. The best lag order is selected by sorting the results based on RMSE, followed by MAE and AIC. This selection strategy prioritizes forecasting accuracy while still considering model quality.

After the optimal lag order is identified, the ARDL model is estimated using the run_ardl_model() function. This function fits the ARDL model using the selected p and q values, the next-day closing price as the dependent variable, and the sentiment variables as explanatory variables.

```python
def run_full_ardl_pipeline(
    df: pd.DataFrame,
    split_ratio: float = 0.8,
    max_p: int = 5,
    max_q: int = 5,
    n_splits: int = 5,
    next_day_sentiment: float | dict | None = None,
    yahoo_ticker: str | None = None,
) -> dict:
    df = df.copy()
    df["Date"] = pd.to_datetime(df["Date"])
    df = df.sort_values("Date").reset_index(drop=True)

    if yahoo_ticker is not None and "Close" in df.columns and df["Close"].isna().any():
        df = fill_missing_close_from_yahoo(df, yahoo_ticker)

    df_model = prepare_ardl_data(df)
    x_cols = [c for c in df_model.columns if c not in ["Date", "Close", "Next_Day_Close_Price"]]

    train_df, test_df = train_test_split_time_series(df_model, split_ratio=split_ratio)

    best_p, best_q, cv_results = get_best_order(
        train_df=train_df,
        y_col="Next_Day_Close_Price",
        x_cols=x_cols,
        max_p=max_p,
        max_q=max_q,
        n_splits=n_splits
    )

    final_model = run_ardl_model(train_df, best_p, best_q, y_col="Next_Day_Close_Price", x_cols=x_cols)

    forecast_df = walk_forward_forecast(
        train_df=train_df,
        test_df=test_df,
        p=best_p,
        q=best_q,
        y_col="Next_Day_Close_Price",
        x_cols=x_cols
    )
    metrics = calculate_metrics(
        y_true=forecast_df["Actual"],
        y_pred=forecast_df["Predicted"],
        aic=final_model.aic
    )
    next_day_prediction = predict_next_day(
        full_df=df_model,
        p=best_p,
        q=best_q,
        next_day_sentiment=next_day_sentiment,
        y_col="Next_Day_Close_Price",
        x_cols=x_cols
    )

    return {
        "raw_data": df,
        "model_data": df_model,
        "train_data": train_df,
        "test_data": test_df,
        "cv_results": cv_results,
        "best_order": (best_p, best_q),
        "x_cols": x_cols,
        "model": final_model,
        "forecast_df": forecast_df,
        "metrics": metrics,
        "next_day_prediction": next_day_prediction,
    }
```

Figure 5.2.11 The run_full_ardl_pipeline() function

5.2.2.4 Walk-Forward Forecasting

The forecasting process is implemented using walk-forward forecasting. This method simulates real-world prediction conditions by training the model on available historical data and predicting the next observation. After each prediction, the actual observation is added to the historical dataset before the next prediction is made. This prevents the model from using future data and provides a more realistic evaluation of forecasting performance. The prediction results are stored together with their corresponding actual values. This output is later used to generate the actual-versus-predicted forecast chart and to compute performance metrics.

Model performance is evaluated using the calculate_metrics() function. The function computes Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and AIC. MAE measures the average absolute forecasting error, while RMSE penalizes larger prediction errors more heavily. AIC is obtained from the fitted ARDL model and is used to evaluate model quality while considering model complexity.

In addition to testing-set forecasting, the system also generates a next-day prediction using the predict_next_day() function. This function fits the ARDL model using the full available dataset and predicts the next closing price. If no future sentiment value is available, the system uses a neutral sentiment value as the default input. This allows the dashboard to provide an estimated next-day closing price after the latest available observation.

```python
def predict_next_day(
    full_df: pd.DataFrame,
    p: int,
    q: int,
    next_day_sentiment=None,
    y_col: str = "Next_Day_Close_Price",
    x_cols: list[str] = ["Sentiment"],
) -> float:
    full_model = run_ardl_model(
        train_df=full_df,
        p=p,
        q=q,
        y_col=y_col,
        x_cols=x_cols
    )

    if next_day_sentiment is None:
        x_next = pd.DataFrame([{col: 0.0 for col in x_cols}])
    elif isinstance(next_day_sentiment, dict):
        x_next = pd.DataFrame([{col: float(next_day_sentiment.get(col, 0.0)) for col in x_cols}])
    else:
        # 兼容旧版只有一个 Sentiment
        if len(x_cols) != 1:
            raise ValueError("Scalar next_day_sentiment only works when there is exactly one sentiment column.")
        x_next = pd.DataFrame({x_cols[0]: [float(next_day_sentiment)]})

    next_pred = full_model.predict(
        start=len(full_df),
        end=len(full_df),
        exog_oos=x_next
    )

    return float(np.asarray(next_pred)[0])
```

Figure 5.2.12 The predict_next_day() function

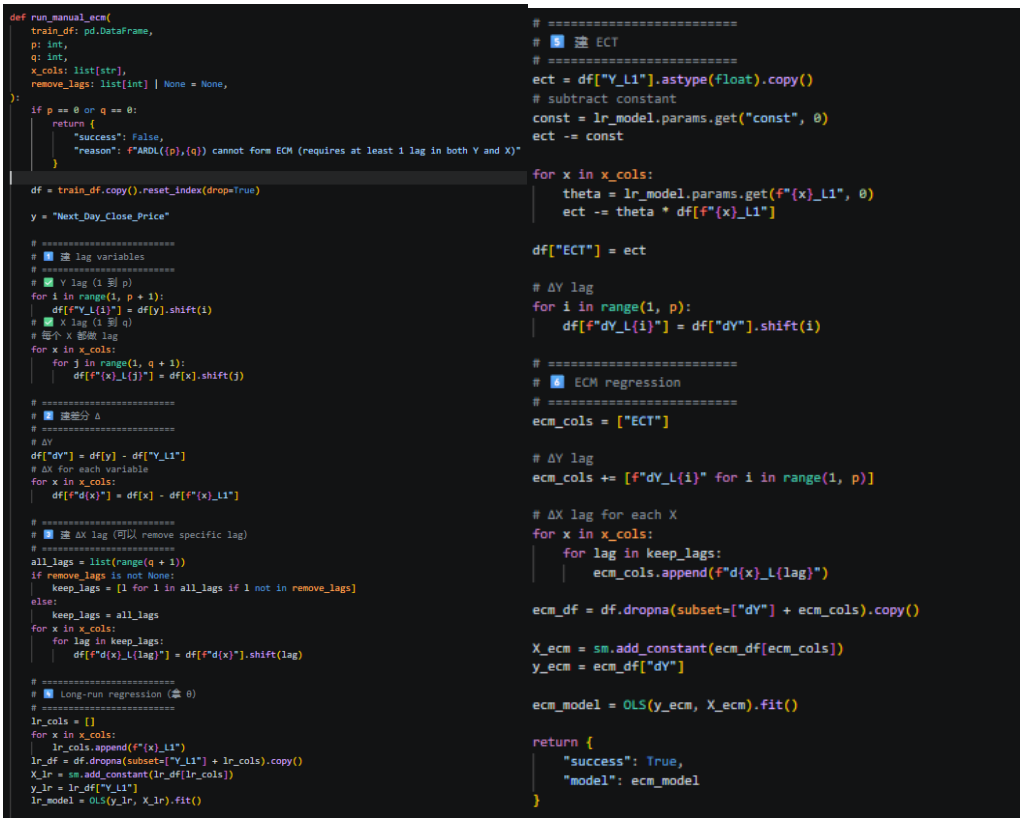
5.2.2.5 Error Correction Model

The Stock Prediction also includes an Error Correction Model implementation through the run_manual_ecm() function. ECM is only constructed when both the autoregressive lag p and sentiment lag q are at least one. This condition is required because the ECM formulation depends on lagged values of both the dependent variable and the explanatory variable.

The ECM implementation first creates lagged variables for the dependent variable and sentiment variables. It then computes first differences to represent short-run changes. A long-run regression is estimated using lagged variables, and the residual relationship is used to construct the Error Correction Term (ECT). Finally, an OLS regression is fitted using the ECT, differenced variables, and selected lag terms. The ECT coefficient and p-value are later extracted to evaluate whether the adjustment toward long-run equilibrium is negative and statistically significant.

The Error Correction Term is constructed from the long-run relationship between the lagged dependent variable and lagged sentiment variables. This allows the system to evaluate whether deviations from long-run equilibrium are corrected over time.

The model also supports lag removal testing. This feature allows selected sentiment lags to be excluded from the model so that the forecasting performance of the original lag structure can be compared with a simplified structure. The comparison is based on RMSE and MAE, which helps determine whether removing less useful lags improves prediction accuracy.



```python
def run_manual_ecm(
    train_df: pd.DataFrame,
    p: int,
    q: int,
    x_cols: list[str],
    remove_lags: list[int] | None = None,
):
    if p == 0 or q == 0:
        return {
            "success": False,
            "reason": f"ARDL({p},{q}) cannot form ECM (requires at least 1 lag in both Y and X)"
        }

    df = train_df.copy().reset_index(drop=True)

    y = "Next_Day_Close_Price"

    # ========================
    # 1️⃣ 建 lag variables
    # ========================
    # ✅ Y lag (1 到 p)
    for i in range(1, p + 1):
        df[f"Y_L{i}"] = df[y].shift(i)
    # ✅ X lag (1 到 q)
    # 每个 X 都做 lag
    for x in x_cols:
        for j in range(1, q + 1):
            df[f"{x}_L{j}"] = df[x].shift(j)

    # ========================
    # 2️⃣ 建差分 Δ
    # ========================
    # ΔY
    df["dY"] = df[y] - df["Y_L1"]
    # ΔX for each variable
    for x in x_cols:
        df[f"d{x}"] = df[x] - df[f"{x}_L1"]

    # ========================
    # 3️⃣ 建 ΔX lag (可以 remove specific lag)
    # ========================
    all_lags = list(range(q + 1))
    if remove_lags is not None:
        keep_lags = [l for l in all_lags if l not in remove_lags]
    else:
        keep_lags = all_lags
    for x in x_cols:
        for lag in keep_lags:
            df[f"d{x}_L{lag}"] = df[f"d{x}"].shift(lag)

    # ========================
    # 4️⃣ Long-run regression (拿 θ)
    # ========================
    lr_cols = []
    for x in x_cols:
        lr_cols.append(f"{x}_L1")
    lr_df = df.dropna(subset=["Y_L1"] + lr_cols).copy()
    X_lr = sm.add_constant(lr_df[lr_cols])
    y_lr = lr_df["Y_L1"]
    lr_model = OLS(y_lr, X_lr).fit()

    # ========================
    # 5️⃣ 建 ECT
    # ========================
    ect = df["Y_L1"].astype(float).copy()
    # subtract constant
    const = lr_model.params.get("const", 0)
    ect -= const

    for x in x_cols:
        theta = lr_model.params.get(f"{x}_L1", 0)
        ect -= theta * df[f"{x}_L1"]

    df["ECT"] = ect

    # ΔY lag
    for i in range(1, p):
        df[f"dY_L{i}"] = df["dY"].shift(i)

    # ========================
    # 6️⃣ ECM regression
    # ========================
    ecm_cols = ["ECT"]

    # ΔY lag
    ecm_cols += [f"dY_L{i}" for i in range(1, p)]

    # ΔX lag for each X
    for x in x_cols:
        for lag in keep_lags:
            ecm_cols.append(f"d{x}_L{lag}")

    ecm_df = df.dropna(subset=["dY"] + ecm_cols).copy()

    X_ecm = sm.add_constant(ecm_df[ecm_cols])
    y_ecm = ecm_df["dY"]

    ecm_model = OLS(y_ecm, X_ecm).fit()

    return {
        "success": True,
        "model": ecm_model
    }
```

Figure 5.2.13 The run_manual_ecm() function

5.2.3 Web-Based User Interface

The Web-Based User Interface is implemented using Streamlit in the app.py script. This component provides an interactive platform that allows users to configure the forecasting model, execute the ARDL pipeline, and visualize the results. It acts as the interface between user inputs and the underlying data preparation and modeling components.

5.2.3.1 System Execution

The forecasting system is executed using Streamlit, which provides a web-based interface for user interaction. The application is launched locally using the following command:

python -m streamlit run app.py

Once executed, the system opens in a web browser, allowing users to interact with the forecasting model and view results dynamically.

The sidebar component allows users to configure the forecasting model. Users can select the target bank, the GPT model used for sentiment generation, and the news source. These selections determine which dataset is loaded and which ARDL model is executed. The interface also provides optional controls for displaying training data in the forecast chart and for showing the full ARDL summary table.

```
# ============================
# Sidebar
# ============================
st.sidebar.header("Configuration")

bank = st.sidebar.selectbox(
    "Select Bank",
    ["Maybank", "Public Bank"]
)

model = st.sidebar.selectbox(
    "Select GPT Model",
    ["gpt-5", "gpt-5-mini", "gpt-5-nano"]
)

source_option = st.sidebar.selectbox(
    "Select Source",
    ["All", "The_Star", "i3Investor", "Free_Malaysia_Today", "Yahoo_Finance"]
)

show_training_data = st.sidebar.checkbox(
    "Show training data in forecast chart",
    value=True
)

show_full_ardl_table = st.sidebar.checkbox(
    "Show full ARDL summary table",
    value=False
)
```

Figure 5.2.14 The code for sidebar

In addition, lag removal testing is supported to evaluate simplified model structures. The system compares forecasting performance before and after removing selected lag variables, allowing users to assess whether model simplification improves prediction accuracy.

```python
# ============================
# Level 5: Error Correction Model
# ============================
st.subheader("⚙️ Error Correction Model (ECM)")

# remove lag selection
remove_lags = st.multiselect(
    "Remove Sentiment Lags",
    options=list(range(best_q + 1)),
    default=[]
)

with st.spinner("Running ECM..."):
    ecm_result = run_manual_ecm(
        train_df=train_df,
        p=best_p,
        q=best_q,
        x_cols=results["x_cols"],
        remove_lags=remove_lags
    )
```

Figure 5.2.15 The code for lag removal

5.3 System Operation (with Screenshot)

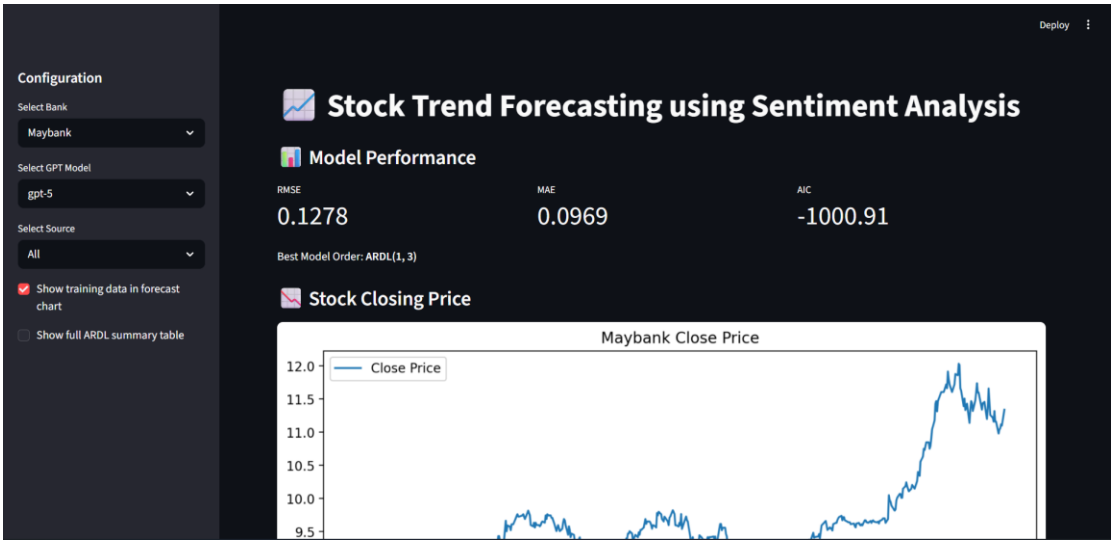


Figure 5.3.1 Main user interface showing configuration selection options

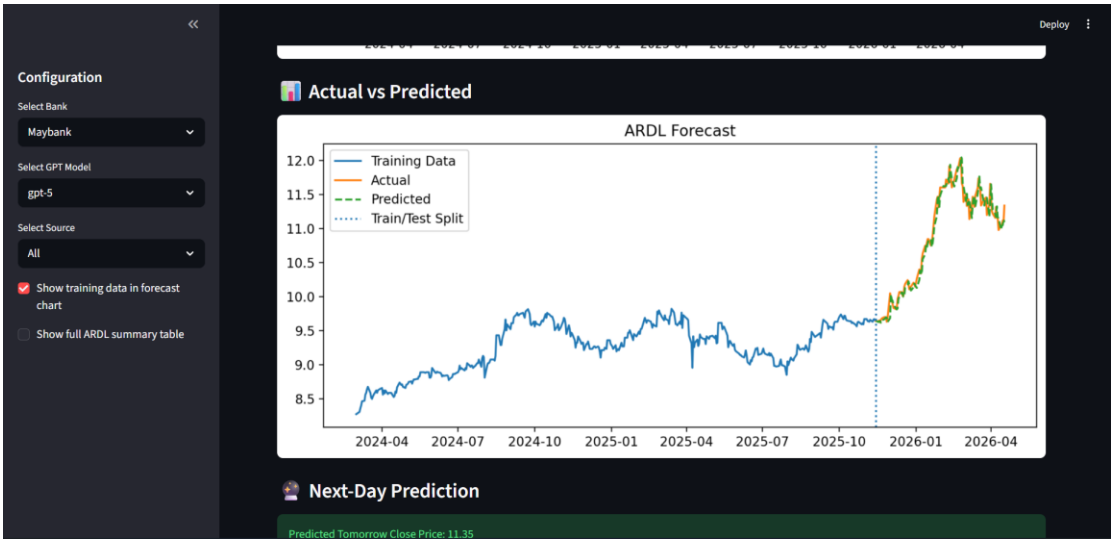


Figure 5.3.2 Demonstrating the forecasting performance of the ARDL model

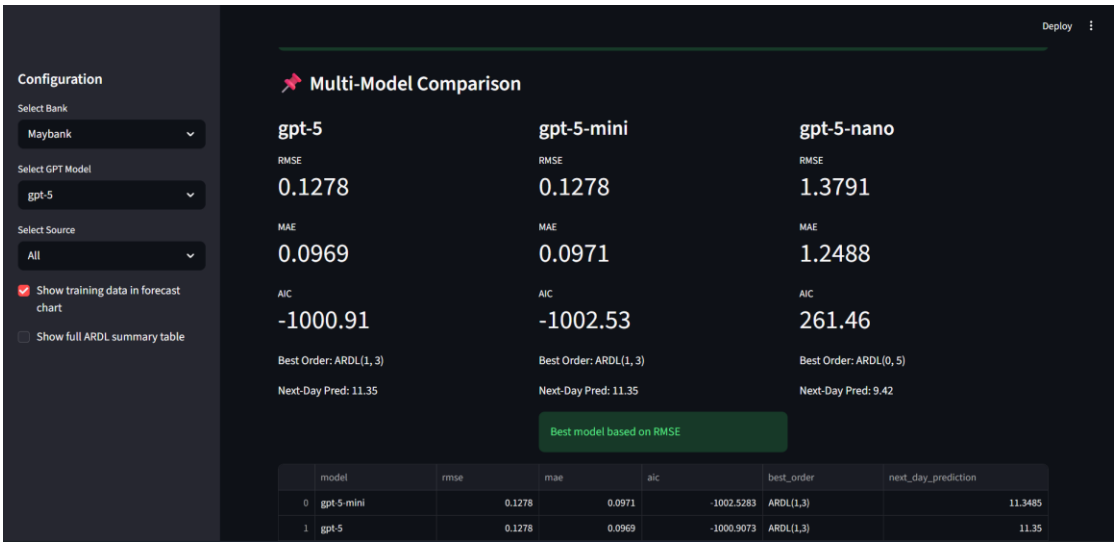


Figure 5.3.3 Model performance metrics used to evaluate forecasting accuracy

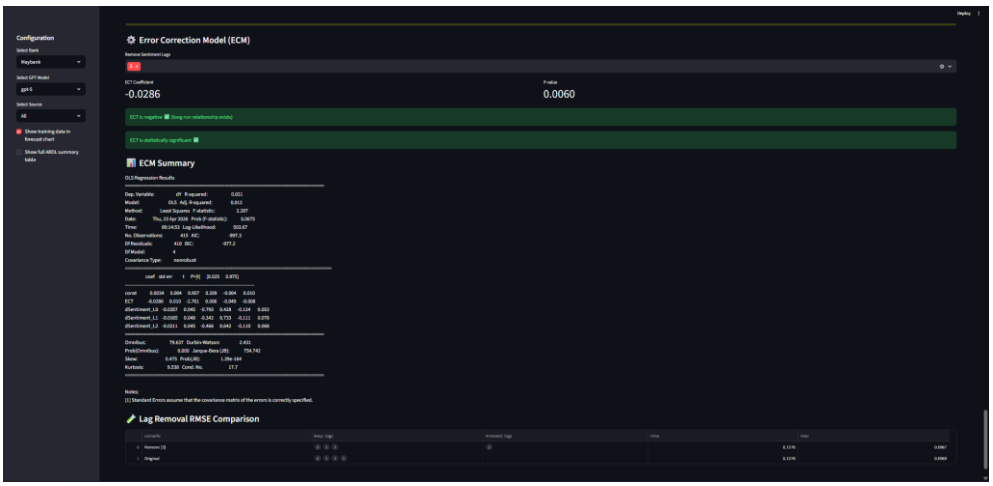


Figure 5.3.4 ECM analysis and lag removal testing results

5.4 Implementation Issues and Challenges

5.4.1 Data Availability and Quality Issues

One of the main challenges encountered in this project is the availability and quality of financial news data. The dataset is collected from multiple online sources, and the amount of news varies across different time periods.

In particular, there are periods where very few or no news articles are available, resulting in sentiment values that are either missing or close to zero. This creates an imbalance in the dataset and may affect the reliability of sentiment-based modeling.

To address this issue, missing sentiment values are handled by assigning neutral values (0), and multiple news sources are combined to increase data coverage. This helps improve the consistency of the dataset used for modeling.

5.4.2 Limited Impact of Sentiment on Stock Prices

Another challenge observed during implementation is that sentiment does not always have a strong or statistically significant impact on stock prices. In many cases, sentiment variables are found to be insignificant in the ARDL model.

This may be due to the complexity of financial markets, where stock prices are influenced by multiple factors beyond news sentiment, such as macroeconomic conditions, company performance, and investor behavior.

Despite this limitation, the model is still useful for exploring potential relationships and understanding delayed effects of sentiment on stock prices.

5.4.3 Complexity of Integrating Multiple Components

The system integrates multiple components, including data collection, sentiment analysis, ARDL modeling, ECM analysis, and user interface interaction. Managing the interaction between these components presents a challenge in terms of system design and implementation.

Ensuring that each component communicates correctly and produces consistent results requires careful structuring of the code and data flow.

This challenge is addressed by modularizing the system into separate components, such as data processing, modeling, and visualization modules. This improves maintainability and allows easier debugging and future extension of the system.

5.5 Concluding Remark

This chapter has presented the implementation of the stock trend forecasting system. The system is successfully developed using a combination of data preparation, ARDL modeling, and a web-based user interface. The hardware and software setup demonstrate that the system can be implemented using normal computing resources without requiring high-cost equipment.

The integration of multiple components, including automated data collection, sentiment analysis using GPT models, and econometric modeling, enables the system to perform one-step-ahead forecasting effectively. The inclusion of evaluation metrics, lag analysis, and Error Correction Model (ECM) further enhances the analytical capability of the system.

Despite several implementation challenges, such as data availability and quality issues and limited impact of sentiment on stock prices, appropriate solutions have been applied to ensure system reliability and performance. The developed user interface allows users to interact with the model, visualize results, and explore the impact of different configurations dynamically.

Overall, the implementation demonstrates that the proposed approach is feasible and practical for short-term stock trend forecasting using sentiment analysis. The system not only provides prediction results but also offers valuable insights into the relationship between news sentiment and stock price movements.

CHAPTER 6

SYSTEM RESULT AND DISCUSSION

6.1 System Result and Discussion

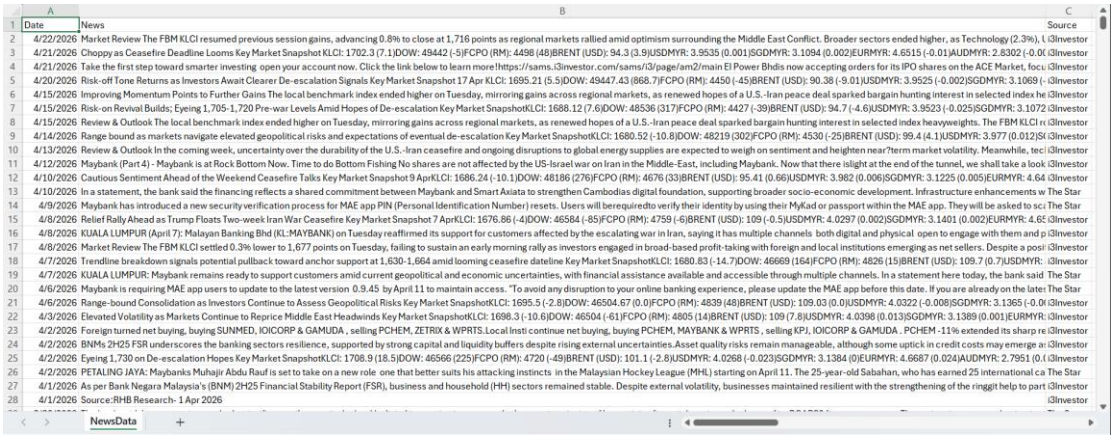
This section presents the testing results and performance evaluation of the proposed stock price forecasting system. The evaluation is conducted using two Malaysian banking stocks, namely Maybank and Public Bank, in order to assess the robustness and generalizability of the system across different datasets.

The dataset used in this study spans approximately two years, from 1 March 2024 to the current date in 2026. This period includes a sufficient number of financial news articles and stock price observations, covering both stable and volatile market conditions. Such a dataset allows the system to be evaluated under realistic scenarios and supports a more comprehensive analysis of model performance.

6.1.1 Result of News and Stock Data Collection

The News and Stock Data Preparation is responsible for collecting financial news data, preprocessing the collected data, and generating sentiment scores for use in the forecasting model. This module forms the foundation of the entire system, as the quality of the input data directly affects the performance of the ARDL model.

The data collection process gathers financial news articles related to Maybank and Public Bank from multiple sources, including The Star, Free Malaysia Today, i3Investor, and Yahoo Finance. The collected data is stored in CSV format and continuously updated through the automated pipeline.



	A	B	C
1	Date	News	Source
2	4/22/2026	Market Review The FBM KLCI resumed previous session gains, advancing 0.8% to close at 1,716 points as regional markets rallied amid optimism surrounding the Middle East Conflict. Broader sectors ended higher, as Technology (2.3%), L	i3Investor
3	4/21/2026	Choppy as Ceasefire Deadline Looms Key Market Snapshot KLCI: 1702.3 (7.1)DOW: 49442 (-5)FCPO (RM): 4498 (48)BRENT (USD): 94.3 (3.9)USDMYR: 3.9535 (0.001)SGDMYR: 3.1094 (0.002)EURMYR: 4.6515 (-0.01)AUDMYR: 2.8302 (-0.0(	i3Investor
4	4/21/2026	Take the first step toward smarter investing open your account now. Click the link below to learn more!https://sams.i3investor.com/sams/i3/page/am2/main El Power Bhdis now accepting orders for its IPO shares on the ACE Market, focu	i3Investor
5	4/20/2026	Risk-off Tone Returns as Investors Await Clearer De-escalation Signals Key Market Snapshot 17 Apr KLCI: 1695.21 (5.5)DOW: 49447.43 (868.7)FCPO (RM): 4450 (-45)BRENT (USD): 90.38 (-9.01)USDMYR: 3.9525 (-0.002)SGDMYR: 3.1069 (-	i3Investor
6	4/15/2026	Improving Momentum Points to Further Gains The local benchmark index ended higher on Tuesday, mirroring gains across regional markets, as renewed hopes of a U.S.-Iran peace deal sparked bargain hunting interest in selected index he	i3Investor
7	4/15/2026	Risk-on Revival Builds; Eyeing 1,705-1,720 Pre-war Levels Amid Hopes of De-escalation Key Market SnapshotKLCI: 1688.12 (7.6)DOW: 48536 (317)FCPO (RM): 4427 (-39)BRENT (USD): 94.7 (-4.6)USDMYR: 3.9523 (-0.025)SGDMYR: 3.1072	i3Investor
8	4/15/2026	Review & Outlook The local benchmark index ended higher on Tuesday, mirroring gains across regional markets, as renewed hopes of a U.S.-Iran peace deal sparked bargain hunting interest in selected index heavyweights. The FBM KLCI r	i3Investor
9	4/14/2026	Range bound as markets navigate elevated geopolitical risks and expectations of eventual de-escalation Key Market SnapshotKLCI: 1680.52 (-10.8)DOW: 48219 (302)FCPO (RM): 4530 (-25)BRENT (USD): 99.4 (4.1)USDMYR: 3.977 (0.012)S	i3Investor
10	4/13/2026	Review & Outlook In the coming week, uncertainty over the durability of the U.S.-Iran ceasefire and ongoing disruptions to global energy supplies are expected to weigh on sentiment and heighten near?term market volatility. Meanwhile, tec	i3Investor
11	4/12/2026	Maybank (Part 4) - Maybank is at Rock Bottom Now. Time to do Bottom Fishing No shares are not affected by the US-Israel war on Iran in the Middle-East, including Maybank. Now that there islight at the end of the tunnel, we shall take a look	i3Investor
12	4/10/2026	Cautious Sentiment Ahead of the Weekend Ceasefire Talks Key Market Snapshot 9 AprKLCI: 1686.24 (-10.1)DOW: 48186 (276)FCPO (RM): 4676 (33)BRENT (USD): 95.41 (0.66)USDMYR: 3.982 (0.006)SGDMYR: 3.1225 (0.005)EURMYR: 4.64	i3Investor
13	4/10/2026	In a statement, the bank said the financing reflects a shared commitment between Maybank and Smart Axiata to strengthen Cambodias digital foundation, supporting broader socio-economic development. Infrastructure enhancements w	The Star
14	4/9/2026	Maybank has introduced a new security verification process for MAE app PIN (Personal Identification Number) resets. Users will berequiredto verify their identity by using their MyKad or passport within the MAE app. They will be asked to sc	The Star
15	4/8/2026	Relief Rally Ahead as Trump Floats Two-week Iran War Ceasefire Key Market Snapshot 7 AprKLCI: 1676.86 (-4)DOW: 46584 (-85)FCPO (RM): 4759 (-6)BRENT (USD): 109 (-0.5)USDMYR: 4.0297 (0.002)SGDMYR: 3.1401 (0.002)EURMYR: 4.6	i3Investor
16	4/8/2026	KUALA LUMPUR (April 7): Malayan Banking Bhd (KL:MAYBANK) on Tuesday reaffirmed its support for customers affected by the escalating war in Iran, saying it has multiple channels both digital and physical open to engage with them and p	i3Investor
17	4/8/2026	Market Review The FBM KLCI settled 0.3% lower to 1,677 points on Tuesday, failing to sustain an early morning rally as investors engaged in broad-based profit-taking with foreign and local institutions emerging as net sellers. Despite a posi	i3Investor
18	4/7/2026	Trendline breakdown signals potential pullback toward anchor support at 1,630-1,664 amid looming ceasefire dateline Key Market SnapshotKLCI: 1680.83 (-14.7)DOW: 46669 (164)FCPO (RM): 4826 (15)BRENT (USD): 109.7 (0.7)USDMYR:	i3Investor
19	4/7/2026	KUALA LUMPUR: Maybank remains ready to support customers amid current geopolitical and economic uncertainties, with financial assistance available and accessible through multiple channels. In a statement here today, the bank said	The Star
20	4/6/2026	Maybank is requiring MAE app users to update to the latest version 0.9.45 by April 11 to maintain access. "To avoid any disruption to your online banking experience, please update the MAE app before this date. If you are already on the late	The Star
21	4/6/2026	Range-bound Consolidation as Investors Continue to Assess Geopolitical Risks Key Market SnapshotKLCI: 1695.5 (-2.8)DOW: 46504.67 (0.0)FCPO (RM): 4839 (48)BRENT (USD): 109.03 (0.0)USDMYR: 4.0322 (-0.008)SGDMYR: 3.1365 (-0.0(	i3Investor
22	4/3/2026	Elevated Volatility as Markets Continue to Reprice Middle East Headwinds Key Market SnapshotKLCI: 1698.3 (-10.6)DOW: 46504 (-61)FCPO (RM): 4805 (14)BRENT (USD): 109 (7.8)USDMYR: 4.0398 (0.013)SGDMYR: 3.1389 (0.001)EURMYR:	i3Investor
23	4/2/2026	Foreign turned net buying, buying SUNMED, IOICORP & GAMUDA , selling PCHEM, ZETRIX & WPRTS.Local Insti continue net buying, buying PCHEM, MAYBANK & WPRTS , selling KPJ, IOICORP & GAMUDA . PCHEM -11% extended its sharp re	i3Investor
24	4/2/2026	BNMs 2H25 FSR underscores the banking sectors resilience, supported by strong capital and liquidity buffers despite rising external uncertainties.Asset quality risks remain manageable, although some uptick in credit costs may emerge a	i3Investor
25	4/2/2026	Eyeing 1,730 on De-escalation Hopes Key Market SnapshotKLCI: 1708.9 (18.5)DOW: 46566 (225)FCPO (RM): 4720 (-49)BRENT (USD): 101.1 (-2.8)USDMYR: 4.0268 (-0.023)SGDMYR: 3.1384 (0)EURMYR: 4.6687 (0.024)AUDMYR: 2.7951 (0.(	i3Investor
26	4/2/2026	PETALING JAYA: Maybanks Muhajir Abdu Rauf is set to take on a new role one that better suits his attacking instincts in the Malaysian Hockey League (MHL) starting on April 11. The 25-year-old Sabahan, who has earned 25 international ca	The Star
27	4/1/2026	As per Bank Negara Malaysia's (BNM) 2H25 Financial Stability Report (FSR), business and household (HH) sectors remained stable. Despite external volatility, businesses maintained resilient with the strengthening of the ringgit help to part	i3Investor
28	4/1/2026	Source:RHB Research- 1 Apr 2026	i3Investor

NewsData

Figure 6.1.1.1 News Data Collected in Github

A total of 1,104 news articles were collected across all sources and banks. The distribution of the collected news data is summarized as follows:

- Maybank
    - Free Malaysia Today: 39 articles
    - i3Investor: 334 articles
    - The Star: 217 articles
    - Yahoo Finance: 5 articles
    - Total: 595 articles
- Public Bank
    - Free Malaysia Today: 12 articles
    - i3Investor: 254 articles
    - The Star: 237 articles
    - Yahoo Finance: 6 articles
    - Total: 509 articles

This distribution shows that the majority of news articles are sourced from i3Investor and The Star, while Yahoo Finance contributes relatively fewer articles. The imbalance in the number of articles across sources may influence the sentiment aggregation and is considered during analysis.

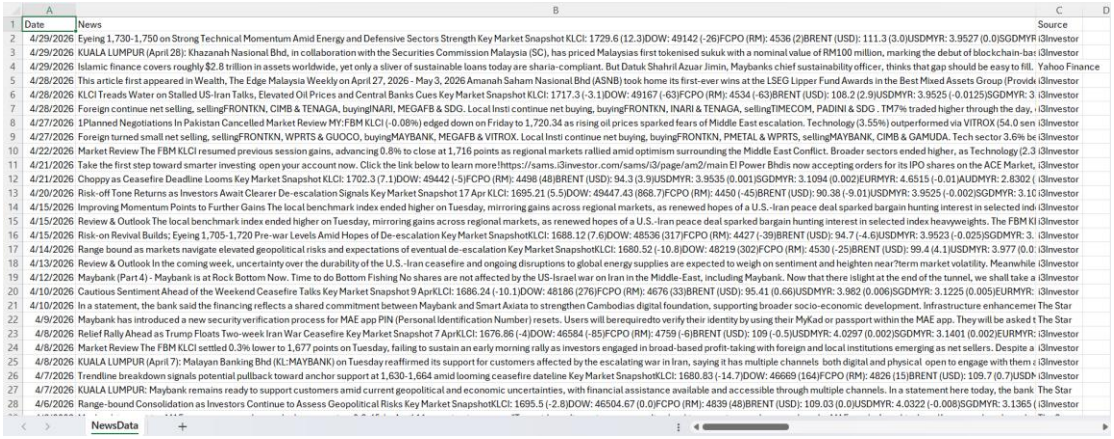
This indicates that sufficient data is available for sentiment analysis, which aims to perform sentiment analysis on financial news obtained from multiple sources. The use of multiple sources also ensures diversity in reporting styles and perspectives, which is important for capturing a more comprehensive representation of market sentiment.

6.1.2 Result of Data Preprocessing for News and Sentiment Analysis

After data collection, preprocessing is performed to clean and standardize the dataset. This step includes handling missing values, normalizing date formats, and merging news data from different sources into a unified structure. The preprocessing stage ensures that inconsistencies between different data sources are resolved before further analysis.

The preprocessing results demonstrate that the cleaned dataset is well-structured and suitable for analysis. As shown in the figure below, the merging process aligns all news articles into a consistent format, allowing sentiment analysis to be performed

across all sources. This step is essential because inconsistencies in raw data could otherwise introduce noise into the sentiment analysis process.



	A	B	C
1	Date	News	Source
2	4/29/2026	Eyeing 1,730-1,750 on Strong Technical Momentum Amid Energy and Defensive Sectors Strength Key Market Snapshot KLCI: 1729.6 (12.3)DOW: 49142 (-26)FCPO (RM): 4536 (2)BRENT (USD): 111.3 (3.0)USDMYR: 3.9527 (0.0)SGDMYR	i3Investor
3	4/29/2026	KUALA LUMPUR (April 28): Khazanah Nasional Bhd, in collaboration with the Securities Commission Malaysia (SC), has priced Malaysias first tokenised sukuk with a nominal value of RM100 million, marking the debut of blockchain-ba	i3Investor
4	4/29/2026	Islamic finance covers roughly $2.8 trillion in assets worldwide, yet only a sliver of sustainable loans today are sharia-compliant. But Datuk Shahril Azuar Jimin, Maybanks chief sustainability officer, thinks that gap should be easy to fill.	Yahoo Finance
5	4/28/2026	This article first appeared in Wealth, The Edge Malaysia Weekly on April 27, 2026 - May 3, 2026 Amanah Saham Nasional Bhd (ASNB) took home its first-ever wins at the LSEG Lipper Fund Awards in the Best Mixed Assets Group (Provid	i3Investor
6	4/28/2026	KLCI Treads Water on Stalled US-Iran Talks, Elevated Oil Prices and Central Banks Cues Key Market Snapshot KLCI: 1717.3 (-3.1)DOW: 49167 (-63)FCPO (RM): 4534 (-63)BRENT (USD): 108.2 (2.9)USDMYR: 3.9525 (-0.0125)SGDMYR: 3	i3Investor
7	4/28/2026	Foreign continue net selling, sellingFRONTKN, CIMB & TENAGA, buyingINARI, MEGAFB & SDG. Local Insti continue net buying, buyingFRONTKN, INARI & TENAGA, sellingTIMECOM, PADINI & SDG . TM7% traded higher through the day,	i3Investor
8	4/27/2026	1Planned Negotiations In Pakistan Cancelled Market Review MY:FBM KLCI (-0.08%) edged down on Friday to 1,720.34 as rising oil prices sparked fears of Middle East escalation. Technology (3.55%) outperformed via VITROX (54.0 sen	i3Investor
9	4/27/2026	Foreign turned small net selling, sellingFRONTKN, WPRTS & GUOCO, buyingMAYBANK, MEGAFB & VITROX. Local Insti continue net buying, buyingFRONTKN, PMETAL & WPRTS, sellingMAYBANK, CIMB & GAMUDA. Tech sector 3.6% be	i3Investor
10	4/22/2026	Market Review The FBM KLCI resumed previous session gains, advancing 0.8% to close at 1,716 points as regional markets rallied amid optimism surrounding the Middle East Conflict. Broader sectors ended higher, as Technology (2.3	i3Investor
11	4/21/2026	Take the first step toward smarter investing open your account now. Click the link below to learn more!https://sams.i3investor.com/sams/i3/page/am2/main El Power Bhdis now accepting orders for its IPO shares on the ACE Market,	i3Investor
12	4/21/2026	Choppy as Ceasefire Deadline Looms Key Market Snapshot KLCI: 1702.3 (7.1)DOW: 49442 (-5)FCPO (RM): 4498 (48)BRENT (USD): 94.3 (3.9)USDMYR: 3.9535 (0.001)SGDMYR: 3.1094 (0.002)EURMYR: 4.6515 (-0.01)AUDMYR: 2.8302 (	i3Investor
13	4/20/2026	Risk-off Tone Returns as Investors Await Clearer De-escalation Signals Key Market Snapshot 17 Apr KLCI: 1695.21 (5.5)DOW: 49447.43 (868.7)FCPO (RM): 4450 (-45)BRENT (USD): 90.38 (-9.01)USDMYR: 3.9525 (-0.002)SGDMYR: 3.1(	i3Investor
14	4/15/2026	Improving Momentum Points to Further Gains The local benchmark index ended higher on Tuesday, mirroring gains across regional markets, as renewed hopes of a U.S.-Iran peace deal sparked bargain hunting interest in selected indi	i3Investor
15	4/15/2026	Review & Outlook The local benchmark index ended higher on Tuesday, mirroring gains across regional markets, as renewed hopes of a U.S.-Iran peace deal sparked bargain hunting interest in selected index heavyweights. The FBM KI	i3Investor
16	4/15/2026	Risk-on Revival Builds; Eyeing 1,705-1,720 Pre-war Levels Amid Hopes of De-escalation Key Market SnapshotKLCI: 1688.12 (7.6)DOW: 48536 (317)FCPO (RM): 4427 (-39)BRENT (USD): 94.7 (-4.6)USDMYR: 3.9523 (-0.025)SGDMYR: 3.	i3Investor
17	4/14/2026	Range bound as markets navigate elevated geopolitical risks and expectations of eventual de-escalation Key Market SnapshotKLCI: 1680.52 (-10.8)DOW: 48219 (302)FCPO (RM): 4530 (-25)BRENT (USD): 99.4 (4.1)USDMYR: 3.977 (0.0	i3Investor
18	4/13/2026	Review & Outlook In the coming week, uncertainty over the durability of the U.S.-Iran ceasefire and ongoing disruptions to global energy supplies are expected to weigh on sentiment and heighten near?term market volatility. Meanwhile	i3Investor
19	4/12/2026	Maybank (Part 4) - Maybank is at Rock Bottom Now. Time to do Bottom Fishing No shares are not affected by the US-Israel war on Iran in the Middle-East, including Maybank. Now that there islight at the end of the tunnel, we shall take a	i3Investor
20	4/10/2026	Cautious Sentiment Ahead of the Weekend Ceasefire Talks Key Market Snapshot 9 AprKLCI: 1686.24 (-10.1)DOW: 48186 (276)FCPO (RM): 4676 (33)BRENT (USD): 95.41 (0.66)USDMYR: 3.982 (0.006)SGDMYR: 3.1225 (0.005)EURMYR:	i3Investor
21	4/10/2026	In a statement, the bank said the financing reflects a shared commitment between Maybank and Smart Axiata to strengthen Cambodias digital foundation, supporting broader socio-economic development. Infrastructure enhancemer	The Star
22	4/9/2026	Maybank has introduced a new security verification process for MAE app PIN (Personal Identification Number) resets. Users will berequiredto verify their identity by using their MyKad or passport within the MAE app. They will be asked t	The Star
23	4/8/2026	Relief Rally Ahead as Trump Floats Two-week Iran War Ceasefire Key Market Snapshot 7 AprKLCI: 1676.86 (-4)DOW: 46584 (-85)FCPO (RM): 4759 (-6)BRENT (USD): 109 (-0.5)USDMYR: 4.0297 (0.002)SGDMYR: 3.1401 (0.002)EURMYR:	i3Investor
24	4/8/2026	Market Review The FBM KLCI settled 0.3% lower to 1,677 points on Tuesday, failing to sustain an early morning rally as investors engaged in broad-based profit-taking with foreign and local institutions emerging as net sellers. Despite a	i3Investor
25	4/8/2026	KUALA LUMPUR (April 7): Malayan Banking Bhd (KL:MAYBANK) on Tuesday reaffirmed its support for customers affected by the escalating war in Iran, saying it has multiple channels both digital and physical open to engage with them	i3Investor
26	4/7/2026	Trendline breakdown signals potential pullback toward anchor support at 1,630-1,664 amid looming ceasefire dateline Key Market SnapshotKLCI: 1680.83 (-14.7)DOW: 46669 (164)FCPO (RM): 4826 (15)BRENT (USD): 109.7 (0.7)USDM	i3Investor
27	4/7/2026	KUALA LUMPUR: Maybank remains ready to support customers amid current geopolitical and economic uncertainties, with financial assistance available and accessible through multiple channels. In a statement here today, the bank	The Star
28	4/6/2026	Range-bound Consolidation as Investors Continue to Assess Geopolitical Risks Key Market SnapshotKLCI: 1695.5 (-2.8)DOW: 46504.67 (0.0)FCPO (RM): 4839 (48)BRENT (USD): 109.03 (0.0)USDMYR: 4.0322 (-0.008)SGDMYR: 3.1365 (	i3Investor

NewsData

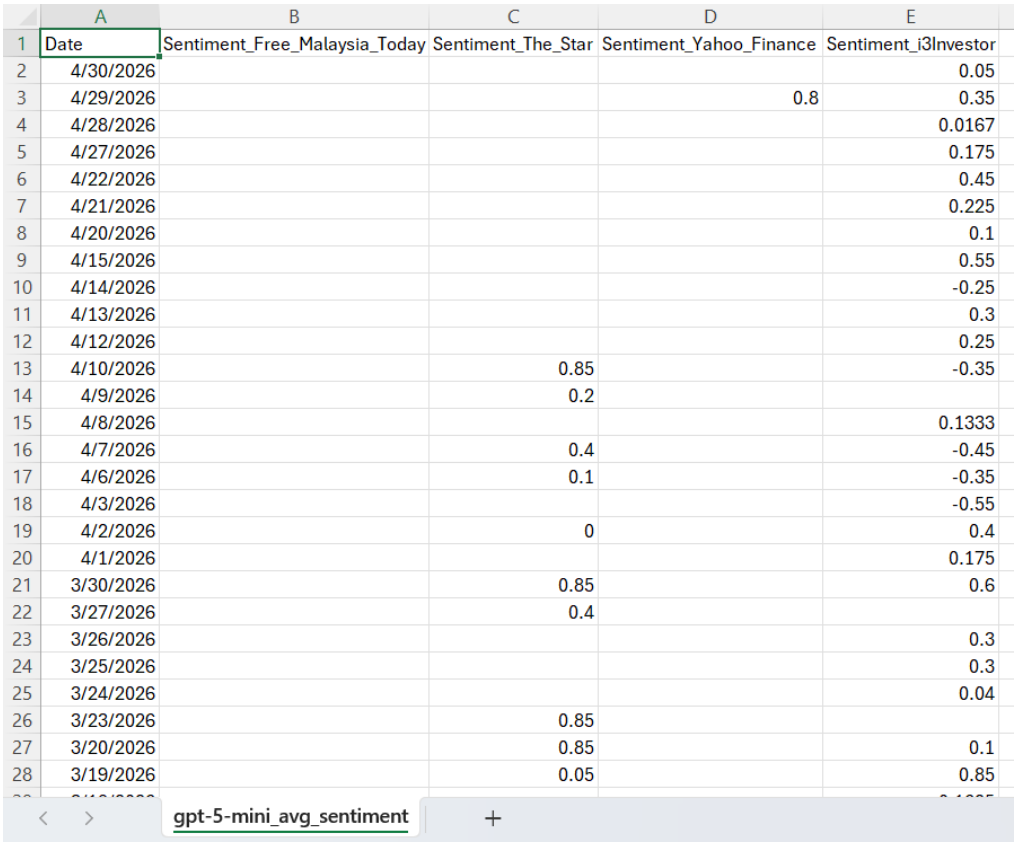
Figure 6.1.2.1 The news content after data cleaning and merging

Following preprocessing, sentiment analysis is performed using GPT models. Each news article is assigned a sentiment score within the range of -1 to 1, representing strongly negative to strongly positive sentiment. The figure below shows the sentiment scores are then aggregated on a daily basis and grouped by news source.

	A	B	C	D	E
1	Date	Sentiment_Free_Malaysia_Today	Sentiment_The_Star	Sentiment_Yahoo_Finance	Sentiment_i3Investor
2	4/30/2026				-0.1
3	4/29/2026			0.64	0.36
4	4/28/2026				-0.0333
5	4/27/2026				0.25
6	4/22/2026				0.2
7	4/21/2026				0.125
8	4/20/2026				0.2
9	4/15/2026				0.46
10	4/14/2026				-0.2
11	4/13/2026				0.2
12	4/12/2026				0.18
13	4/10/2026		0.74		-0.2
14	4/9/2026		0.2		
15	4/8/2026				0.0667
16	4/7/2026		0.3		-0.35
17	4/6/2026		0.1		-0.2
18	4/3/2026				-0.35
19	4/2/2026		0.1		0.3667
20	4/1/2026				0.27
21	3/30/2026		0.775		0.6
22	3/27/2026		0.7		
23	3/26/2026				0.4
24	3/25/2026				0.325
25	3/24/2026				0.08
26	3/23/2026		0.62		
27	3/20/2026		0.66		-0.1
28	3/19/2026		-0.025		0.9

gpt-5_avg_sentiment

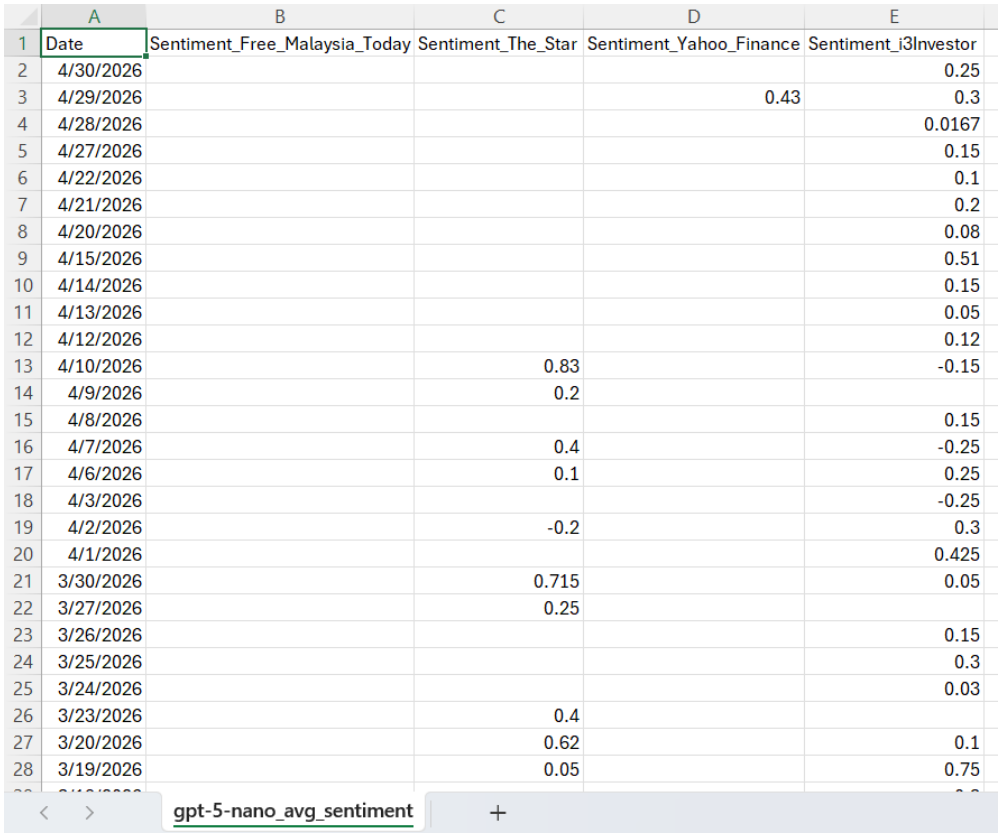
Figure 6.1.2.2 The sentiment analysis result for GPT-5



	A	B	C	D	E
1	Date	Sentiment_Free_Malaysia_Today	Sentiment_The_Star	Sentiment_Yahoo_Finance	Sentiment_i3Investor
2	4/30/2026				0.05
3	4/29/2026			0.8	0.35
4	4/28/2026				0.0167
5	4/27/2026				0.175
6	4/22/2026				0.45
7	4/21/2026				0.225
8	4/20/2026				0.1
9	4/15/2026				0.55
10	4/14/2026				-0.25
11	4/13/2026				0.3
12	4/12/2026				0.25
13	4/10/2026		0.85		-0.35
14	4/9/2026		0.2		
15	4/8/2026				0.1333
16	4/7/2026		0.4		-0.45
17	4/6/2026		0.1		-0.35
18	4/3/2026				-0.55
19	4/2/2026		0		0.4
20	4/1/2026				0.175
21	3/30/2026		0.85		0.6
22	3/27/2026		0.4		
23	3/26/2026				0.3
24	3/25/2026				0.3
25	3/24/2026				0.04
26	3/23/2026		0.85		
27	3/20/2026		0.85		0.1
28	3/19/2026		0.05		0.85

gpt-5-mini_avg_sentiment

Figure 6.1.2.3 The sentiment analysis result for GPT-5-mini



	A	B	C	D	E
1	Date	Sentiment_Free_Malaysia_Today	Sentiment_The_Star	Sentiment_Yahoo_Finance	Sentiment_i3Investor
2	4/30/2026				0.25
3	4/29/2026			0.43	0.3
4	4/28/2026				0.0167
5	4/27/2026				0.15
6	4/22/2026				0.1
7	4/21/2026				0.2
8	4/20/2026				0.08
9	4/15/2026				0.51
10	4/14/2026				0.15
11	4/13/2026				0.05
12	4/12/2026				0.12
13	4/10/2026		0.83		-0.15
14	4/9/2026		0.2		
15	4/8/2026				0.15
16	4/7/2026		0.4		-0.25
17	4/6/2026		0.1		0.25
18	4/3/2026				-0.25
19	4/2/2026		-0.2		0.3
20	4/1/2026				0.425
21	3/30/2026		0.715		0.05
22	3/27/2026		0.25		
23	3/26/2026				0.15
24	3/25/2026				0.3
25	3/24/2026				0.03
26	3/23/2026		0.4		
27	3/20/2026		0.62		0.1
28	3/19/2026		0.05		0.75

gpt-5-nano_avg_sentiment

Figure 6.1.2.4 The sentiment analysis result for GPT-5-nano

The sentiment analysis results show that sentiment varies over time and across different news sources. Some sources exhibit more stable sentiment trends, while others show higher variability. This variation indicates that different news platforms may emphasize different aspects of financial events, leading to differences in sentiment signals.

Figure 6.1.2.3 supports us to investigate the impact of different news sources on stock market movements. The presence of varying sentiment patterns suggests that the choice of news source can influence the sentiment input used in the forecasting model, which may subsequently affect ARDL model performance.

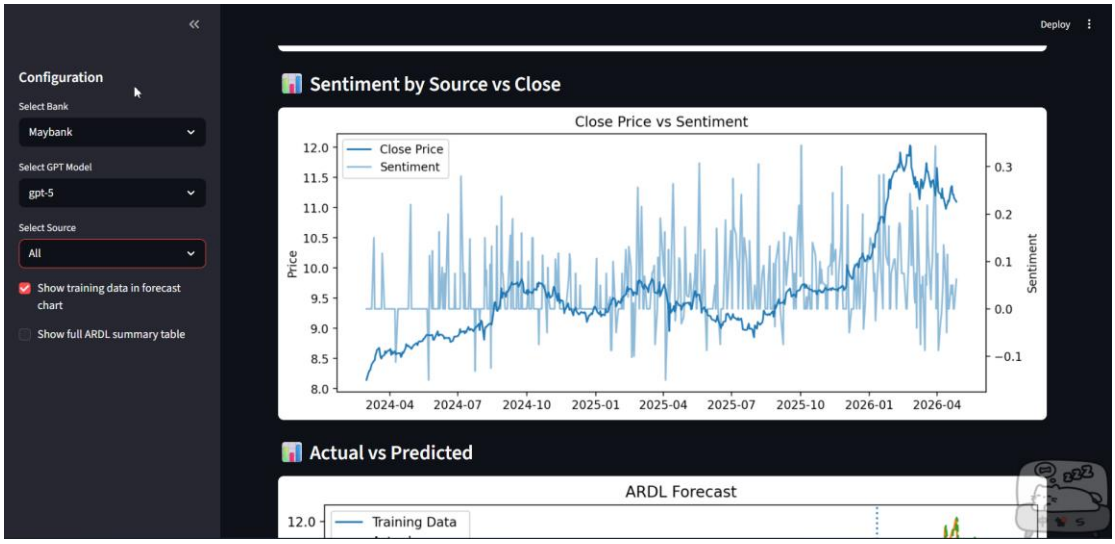


Figure 6.1.2.5 Relationship between sentiment by selected all source and close price

6.1.3 Result of Data Alignment

Finally, the News and Stock Data Preparation successfully produces a structured dataset that integrates stock price data with sentiment scores derived from multiple sources. This dataset serves as the input for the Stock Prediction and enables further analysis of the relationship between sentiment and stock price movements.

During the data alignment process, sentiment scores are synchronized with the corresponding stock closing prices based on the date. Since financial news is not published every day, some dates may not have available sentiment values. To ensure consistency and maintain a continuous time series, missing sentiment values are handled by filling them with 0, indicating a neutral sentiment. This approach prevents data gaps and allows the ARDL model to process the dataset without interruption while preserving the temporal alignment between sentiment and stock price data.

	A	B	C	D	E	F
1	Date	Close	Sentiment_Free_Malaysia_Today	Sentiment_The_Star	Sentiment_Yahoo_Finance	Sentiment_i3Investor
2	4/24/2026	11.16	0	0	0	0
3	4/23/2026	11.22	0	0	0	0
4	4/22/2026	11.22	0	0	0	0.2
5	4/21/2026		0	0	0	0.125
6	4/20/2026	11.34	0	0	0	0.2
7	4/17/2026	11.1	0	0	0	0
8	4/16/2026	11.12	0	0	0	0
9	4/15/2026	11.06	0	0	0	0.46
10	4/14/2026	11.04	0	0	0	-0.2
11	4/13/2026	10.98	0	0	0	0.2
12	4/10/2026	11.16	0	0.74	0	-0.2
13	4/9/2026	11.16	0	0.2	0	0
14	4/8/2026	11.32	0	0	0	0.0667
15	4/7/2026	11.16	0	0.3	0	-0.35
16	4/6/2026	11.22	0	0.1	0	-0.2
17	4/3/2026	11.26	0	0	0	-0.35
18	4/2/2026	11.46	0	0.1	0	0.3667
19	4/1/2026	11.66	0	0	0	0.27
20	3/31/2026	11.36	0	0	0	0
21	3/30/2026	11.2	0	0.775	0	0.6
22	3/27/2026	11.46	0	0.7	0	0
23	3/26/2026	11.44	0	0	0	0.4
24	3/25/2026	11.44	0	0	0	0.325
25	3/24/2026	11.34	0	0	0	0.08
26	3/20/2026	11.6	0	0.66	0	-0.1
27	3/19/2026	11.6	0	-0.025	0	0.9
28	3/18/2026	11.74	0	0	0	0.2
29	3/17/2026	11.6	0	0	0	0.29
30	3/16/2026	11.48	0	0	0	-0.35

gpt-5_And_Closing_Price

Figure 6.1.3.1 The final dataset that included stock closing price and sentiments from various sources

6.2 Stock Prediction Result

6.2.1 Maybank

6.2.1.1 ARDL Model

The Stock Prediction is applied to the Maybank dataset to evaluate the effectiveness of the ARDL model in forecasting next-day stock closing prices using sentiment-derived features. The model incorporates both lagged stock prices and sentiment variables to capture temporal dependencies and delayed market reactions.

The forecasting performance of the model is first evaluated using the actual versus predicted stock price comparison.

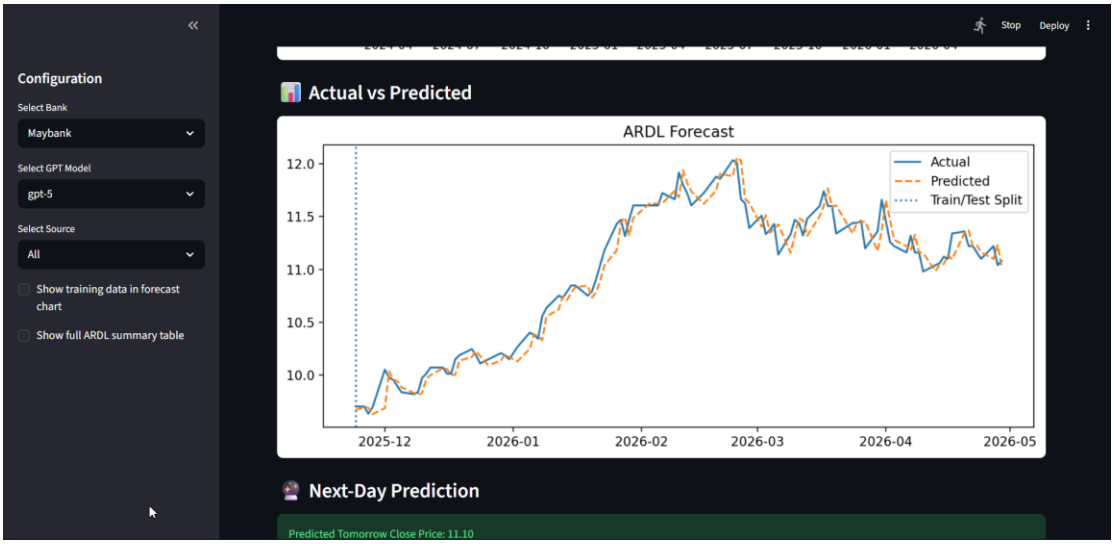


Figure 6.2.1.1.1 GPT-5 Actual vs Predicted Stock Price Comparison for Maybank

Based on the results, the ARDL model is able to closely follow the overall trend of Maybank stock prices. The predicted values align well with the actual values, particularly during stable periods. Quantitatively, the model achieves an RMSE of approximately 0.13 and an MAE of approximately 0.09, indicating a relatively low prediction error. The AIC value further confirms that the selected ARDL(1,3) model (means 1 lag for close price and 3 lags for sentiment) provides a good balance between model fit and complexity compared to other lag configurations.

Although minor deviations can be observed during periods of higher volatility (2026-01 until 2026-03), the model successfully captures the direction of price movements. This demonstrates that the ARDL model is capable of generating reliable short-term forecasts for the Maybank dataset.

Figure 6.2.1.1.2 GPT-5 ARDL Result for Maybank

6.2.1.2 GPT Models Result Comparison

To quantitatively evaluate the model performance, RMSE, MAE, and AIC are used as evaluation metrics.

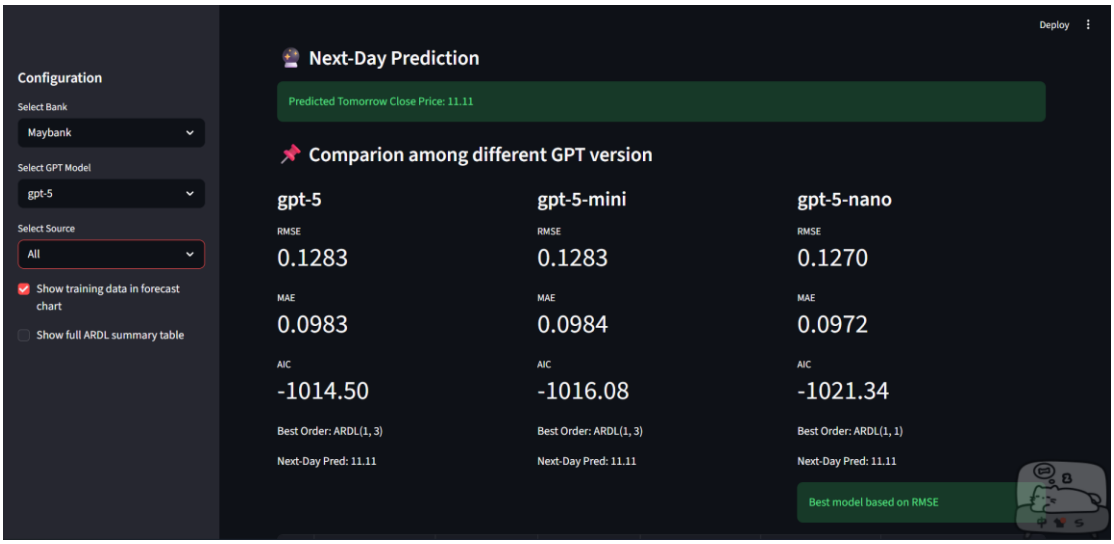


Figure 6.2.1.2 Performance Comparison among different GPT model for Maybank

The results show that all of the GPT models produce similar forecasting performance, with RMSE values around 0.12 and MAE values around 0.09. These relatively low error values indicate that the model is able to produce accurate predictions. The AIC values also suggest that the selected ARDL model achieves a good balance between model fit and complexity.

An important observation from the experimental results is the instability of the GPT-5-nano model in sentiment-based stock prediction. Unlike GPT-5 and GPT-5-mini, which tend to produce consistent ARDL lag structures, GPT-5-nano exhibits significant variation in both lag selection and model performance when the dataset is slightly updated.

	source	term	coef	p_value	significant
0	Sentiment	Sentiment.L0	0.6777	0.0143	Yes
1	Sentiment	Sentiment.L1	0.5801	0.0386	Yes
2	Sentiment	Sentiment.L2	0.5754	0.0387	Yes
3	Sentiment	Sentiment.L3	0.5712	0.0401	Yes
4	Sentiment	Sentiment.L4	0.5313	0.0592	No
5	Sentiment	Sentiment.L5	0.5715	0.0399	Yes

Figure 6.2.1.3 ARDL(0,5) when used GPT-5-nano's dataset

In earlier results, GPT-5-nano selects an ARDL(0,5) structure, indicating that the model does not rely on the autoregressive component of stock prices but instead depends heavily on multiple lagged sentiment variables. In this configuration, several sentiment lag terms are statistically significant, suggesting a strong influence of sentiment on stock price movements.

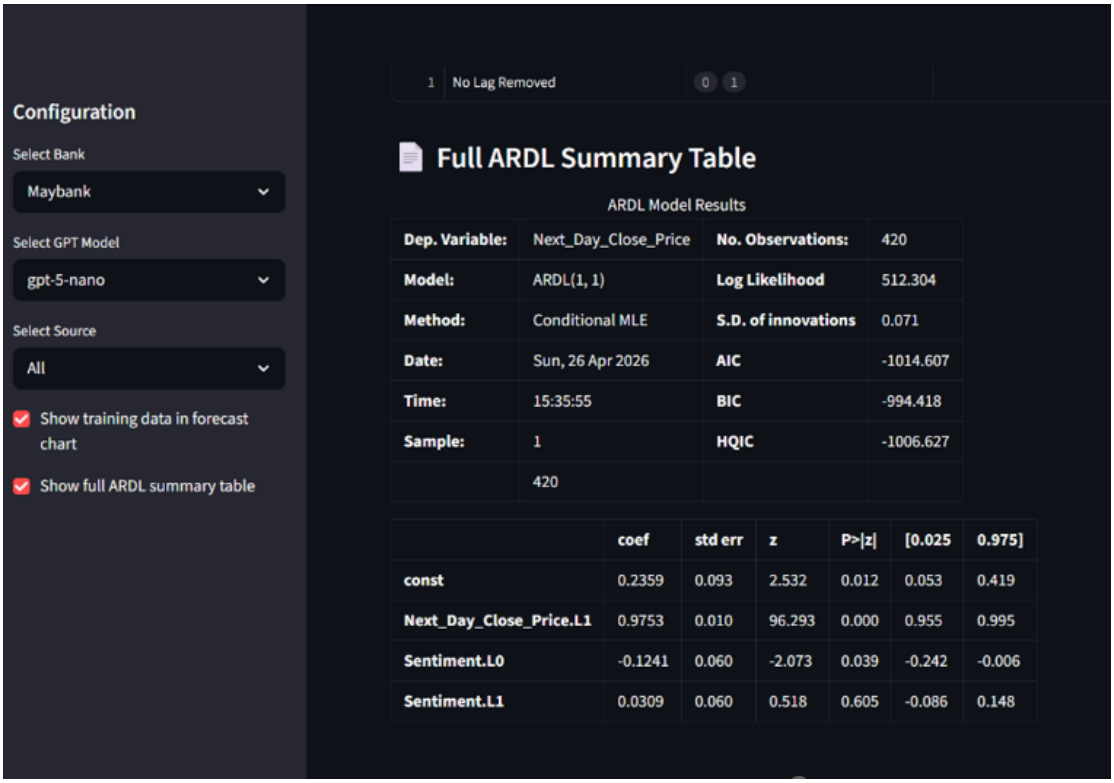


Figure 6.2.1.4 ARDL(1,1) when used GPT-5-nano's dataset

However, in later results, GPT-5-nano selects a different lag structure, such as ARDL(1,1), and the overall model performance changes accordingly. This shift occurs even though the dataset is only slightly updated, indicating that the model is highly sensitive to small variations in input data.

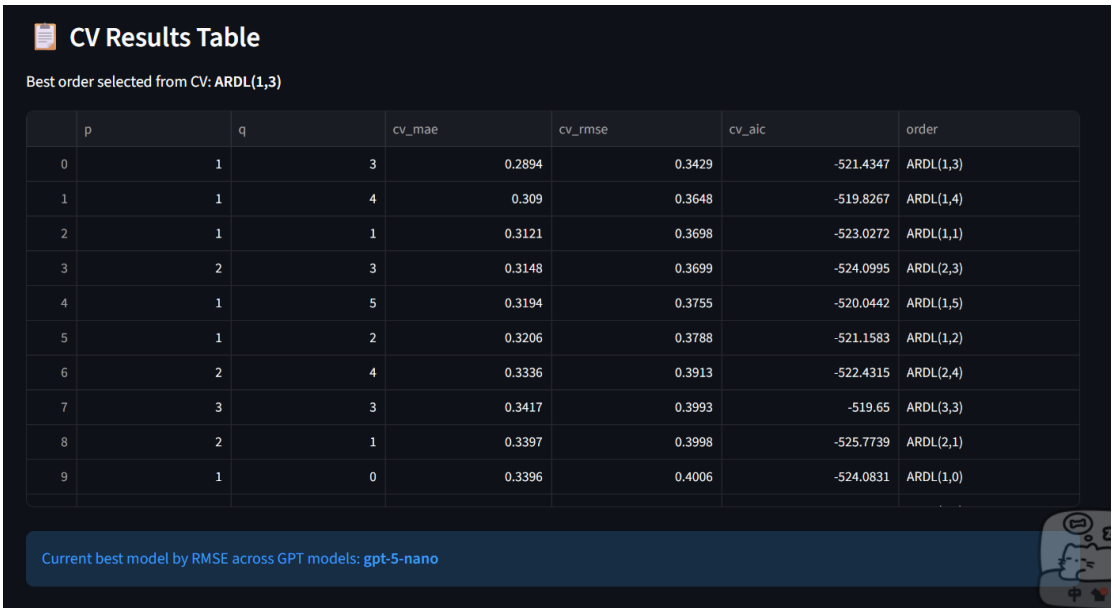
This instability suggests that GPT-5-nano produces less consistent sentiment signals compared to larger models. As a result, the ARDL model built on top of GPT-5-nano sentiment is more volatile in terms of lag selection, statistical significance, and forecasting performance.

From a modeling perspective, this finding highlights the importance of robustness in sentiment generation. While GPT-5-nano may occasionally produce competitive or even superior results, its lack of stability makes it less reliable for

consistent forecasting. Therefore, larger models such as GPT-5 and GPT-5-mini may be more suitable for practical applications where stability is required.

6.2.1.3 Lag Selection Test

In terms of model structure, both GPT-5 and GPT-5-mini consistently select an ARDL(1,3) configuration. This indicates that the model relies on one lag of the stock price and up to three lags of sentiment variables to capture the relationship between sentiment and price movement.



	p	q	cv_mae	cv_rmse	cv_aic	order
0	1	3	0.2894	0.3429	-521.4347	ARDL(1,3)
1	1	4	0.309	0.3648	-519.8267	ARDL(1,4)
2	1	1	0.3121	0.3698	-523.0272	ARDL(1,1)
3	2	3	0.3148	0.3699	-524.0995	ARDL(2,3)
4	1	5	0.3194	0.3755	-520.0442	ARDL(1,5)
5	1	2	0.3206	0.3788	-521.1583	ARDL(1,2)
6	2	4	0.3336	0.3913	-522.4315	ARDL(2,4)
7	3	3	0.3417	0.3993	-519.65	ARDL(3,3)
8	2	1	0.3397	0.3998	-525.7739	ARDL(2,1)
9	1	0	0.3396	0.4006	-524.0831	ARDL(1,0)

Figure 6.2.1.5 GPT-5 Lag Selection Results for Maybank

The lag selection results highlight that different lag combinations produce varying levels of performance, and the ARDL(1,3) model is selected as the optimal configuration based on cross-validation with lowest RMSE. This supports the methodology described in Chapter 3, where expanding-window cross-validation is used to determine the optimal lag order.

6.2.1.4 Error Correction Model

In addition to ARDL forecasting, the Error Correction Model (ECM) is constructed to further examine the dynamic relationship between sentiment and stock price movements.

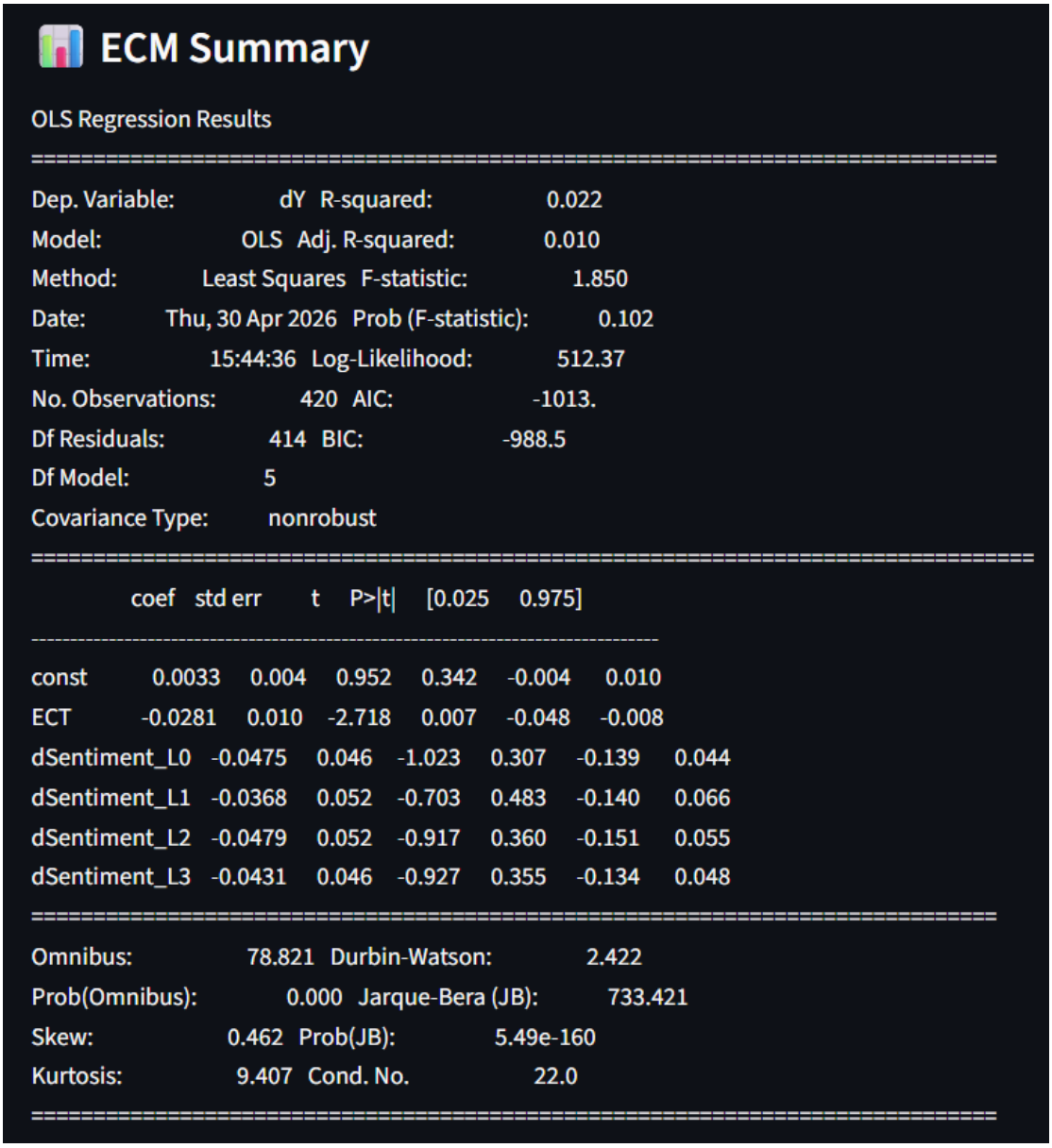


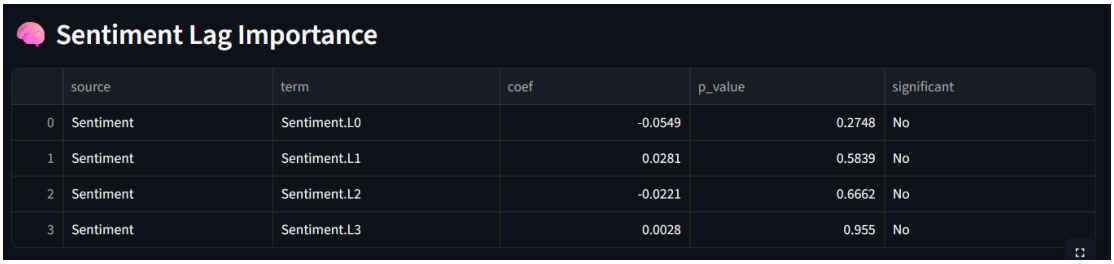
Figure 6.2.1.6 Error Correction Model for Maybank GPT 5

The ECM result provides the coefficient and p-value of the error correction term. A negative error correction coefficient indicates that the model has an adjustment

mechanism toward long-run equilibrium, while the p-value is used to determine whether this adjustment is statistically significant, if p-value < 0.05 means significant; otherwise means not significant. Based on the result, the ECM analysis helps evaluate whether deviations between sentiment and stock price are corrected over time. However, the interpretation depends on whether the selected ARDL lag structure satisfies the ECM requirement and whether the error correction term is statistically significant.

6.2.1.5 Lag Removal Test

Further analysis is conducted to evaluate the statistical significance of sentiment variables within the ARDL model.



Sentiment Lag Importance

	source	term	coef	p_value	significant
0	Sentiment	Sentiment.L0	-0.0549	0.2748	No
1	Sentiment	Sentiment.L1	0.0281	0.5839	No
2	Sentiment	Sentiment.L2	-0.0221	0.6662	No
3	Sentiment	Sentiment.L3	0.0028	0.955	No

Figure 6.2.1.7 GPT-5 Sentiment Lags Significance for Maybank

The estimated results show that the coefficient for Sentiment.L1 is 0.012 with a p-value of 0.412, Sentiment.L2 has a coefficient of -0.008 with a p-value of 0.537, and Sentiment.L3 has a coefficient of 0.015 with a p-value of 0.289. All sentiment lag variables have p-values greater than 0.05, indicating that none of the sentiment terms are statistically significant at the 5% significance level.

This suggests that, in the case of Maybank, sentiment does not provide strong direct explanatory power for stock price movements. However, the inclusion of sentiment variables still contributes to the overall model structure and may capture delayed or indirect effects.

Lag removal testing is also performed to evaluate whether simplifying the sentiment lag structure improves forecasting performance. The lag removal result compares the original ARDL model with the modified model after selected sentiment lags are removed. In this case, we try to remove the lag who has the highest p-value. The comparison is based on RMSE and MAE. If the modified model produces lower RMSE or MAE, it suggests that removing less useful lag terms may improve prediction accuracy. If the error values increase after lag removal, it indicates that the removed lag terms still contribute useful information to the forecasting model.

In addition, the validity of the Error Correction Model (ECM) must also be considered. Specifically, the Error Correction Term (ECT) is expected to be negative and statistically significant. A negative ECT indicates that the system converges back to the long-run equilibrium after short-term deviations. If the ECT becomes positive or insignificant after lag removal, it suggests that the long-run relationship may no longer be stable, even if short-term prediction accuracy improves.

This analysis is important because not all sentiment lags contribute equally to stock prediction. Some lag terms may introduce noise instead of improving forecasting accuracy. Therefore, ECM and lag removal testing provide additional insight beyond basic prediction accuracy and help improve the interpretability of the ARDL model.

Another important finding is the impact of lag removal on model performance. In the ARDL(1,3) configuration for Maybank using GPT-5, multiple sentiment lag terms are initially included in the model. However, not all lag terms contribute equally to forecasting accuracy.

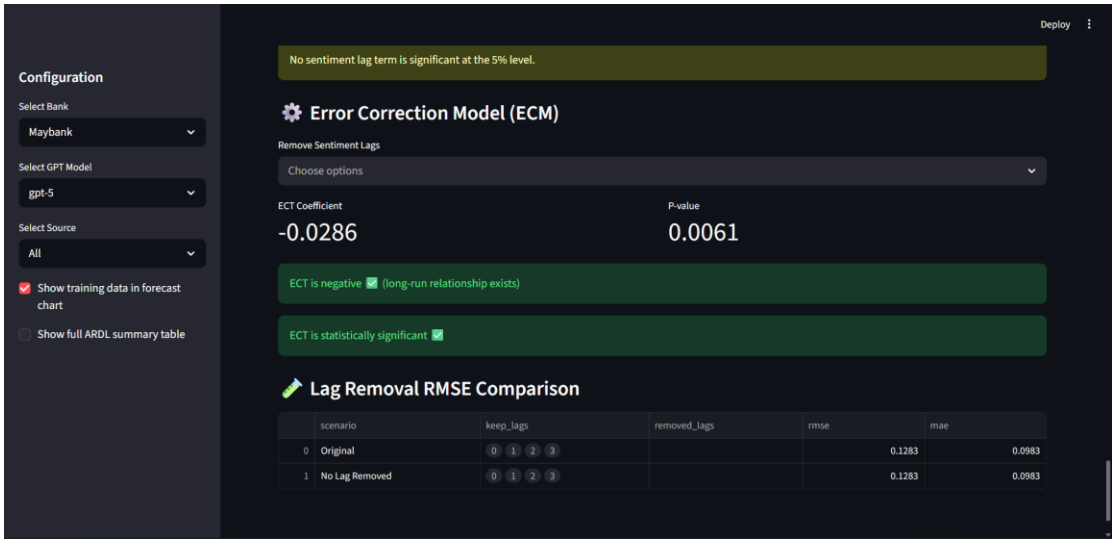


Figure 6.2.1.8 ECM when no lag removed

The original model includes all sentiment lag terms, but some of these terms are not statistically significant. This suggests that certain lag variables may introduce noise instead of improving prediction accuracy.

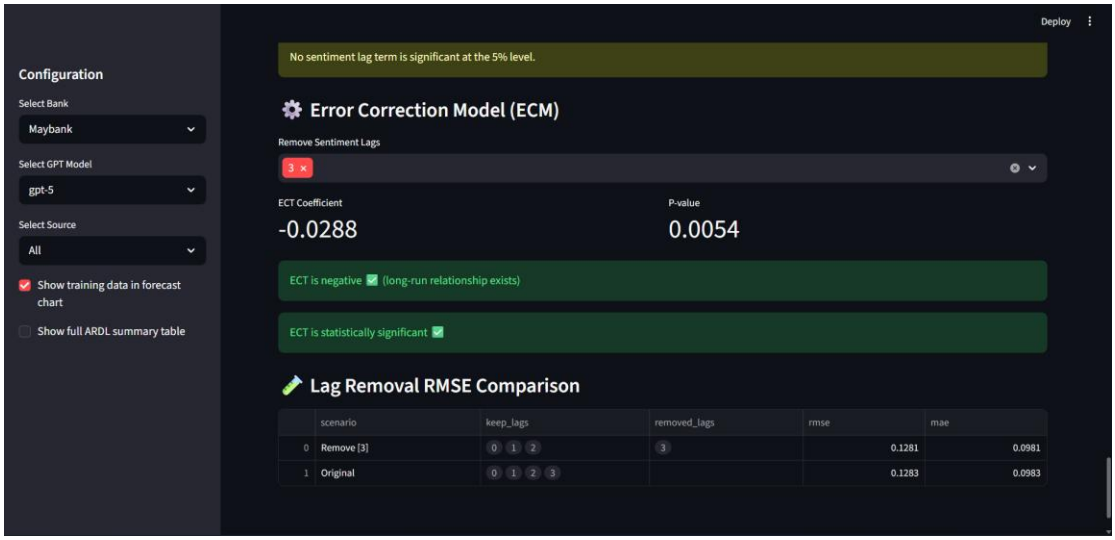


Figure 6.2.1.9 ECM when L3 removed

After removing the sentiment lag with the highest p-value (L3), the model performance shows slight improvement. This indicates that the removed lag may not provide useful information and may negatively affect the model when included.

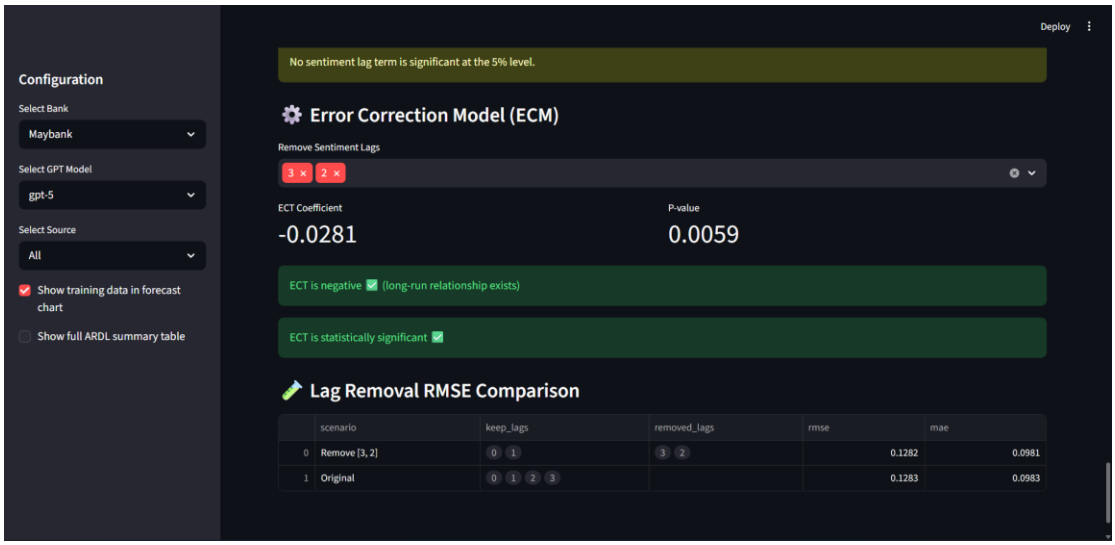


Figure 6.2.1.10 ECM when L3 and L2 removed

Further lag removal was also tested to examine whether additional simplification could improve the model. However, after removing both L3 and L2, the ECM result becomes slightly weaker compared with removing only L3. Specifically, the ECT changes from -0.0288 to -0.0281, while the p-value increases from 0.0054 to 0.0059. Although the ECT remains negative and statistically significant, the adjustment speed toward long-run equilibrium becomes slightly slower. Therefore, L2 is retained in the model, and only L3 is removed, as this provides a better balance between

forecasting performance and long-run equilibrium stability.This finding suggests that the optimal lag structure is not necessarily the most complex one. Including too many lag terms may introduce unnecessary complexity and reduce model generalization. Therefore, lag removal testing is an important step to identify the most effective model configuration.

Overall, this result highlights the importance of model simplification and supports the use of lag analysis as part of the ARDL modeling process.

6.2.2 Public Bank

6.2.2.1 ARDL Model

The Stock Prediction is applied to the Public Bank dataset to evaluate the effectiveness of the ARDL model in forecasting next-day stock closing prices using sentiment-derived features. Unlike the Maybank case, the selected ARDL model for Public Bank exhibits a simpler structure, where sentiment variables are not included in the final model.

The forecasting performance of the model is first evaluated using the actual versus predicted stock price comparison.

Figure 6.2.2.1 GPT-5 Actual vs Predicted Stock Price Comparison for Public Bank

Based on the results, the ARDL model is able to closely follow the overall trend of Public Bank stock prices. The predicted values align well with the actual values, particularly during stable periods. Quantitatively, the model achieves an RMSE of approximately 0.05 and an MAE of approximately 0.04, indicating a relatively low prediction error. The AIC value further confirms that the selected ARDL(1,0) model

(means selected 1 lag for close price and no lag for sentiment) provides a good balance between model fit and complexity compared to other lag configurations.

Although minor deviations can be observed during periods of higher volatility that has been mention previously (2026-01 until 2026-03), the model successfully captures the direction of price movements. This demonstrates that the ARDL model is capable of generating reliable short-term forecasts for the Maybank dataset.

6.2.2.2 GPT Models Result Comparison

To quantitatively evaluate the model performance, RMSE, MAE, and AIC are used as evaluation metrics.

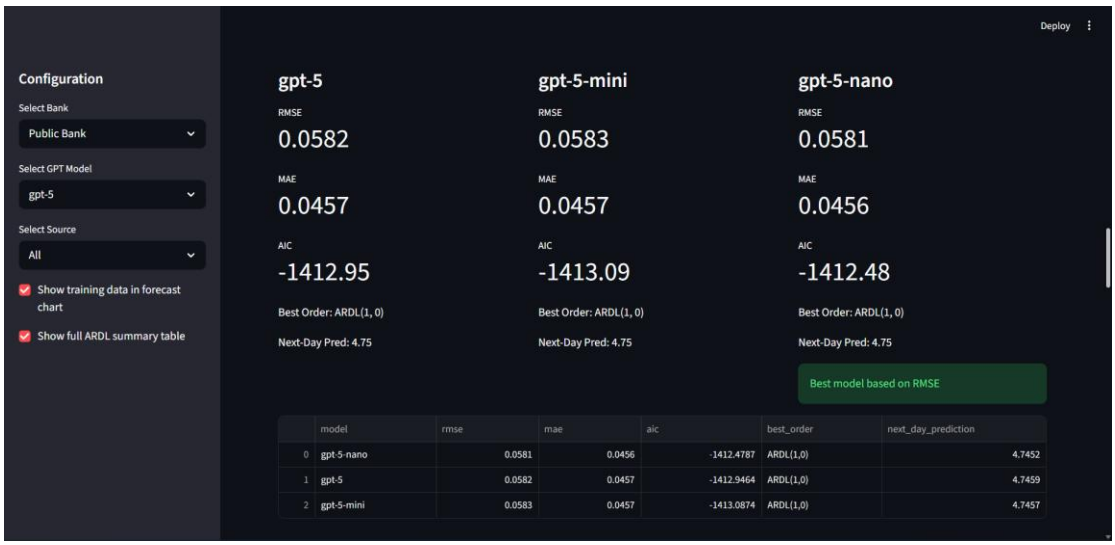
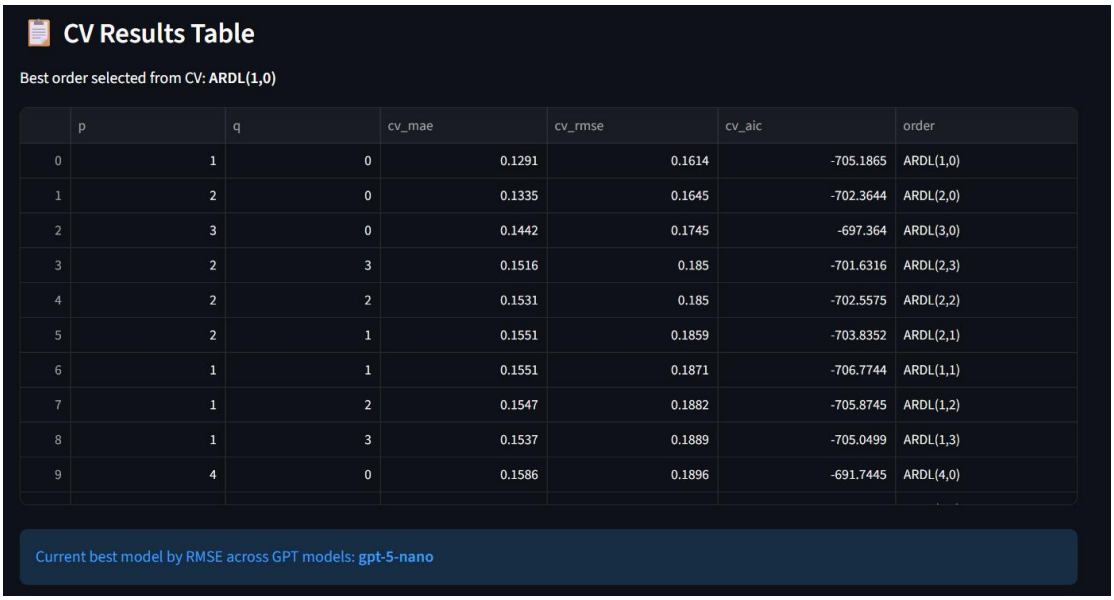


Figure 6.2.2.2 Performance Comparison among different GPT model for Public Bank

The results show that all of the GPT models produce similar forecasting performance, with RMSE values around 0.12 and MAE values around 0.09. These relatively low error values indicate that the model is able to produce accurate predictions. The AIC values also suggest that the selected ARDL model achieves a good balance between model fit and complexity.

6.2.2.3 Lag Selection Test

In terms of model structure, all GPT models consistently select an ARDL(1,0) configuration.



CV Results Table

Best order selected from CV: ARDL(1,0)

	p	q	cv_mae	cv_rmse	cv_aic	order
0	1	0	0.1291	0.1614	-705.1865	ARDL(1,0)
1	2	0	0.1335	0.1645	-702.3644	ARDL(2,0)
2	3	0	0.1442	0.1745	-697.364	ARDL(3,0)
3	2	3	0.1516	0.185	-701.6316	ARDL(2,3)
4	2	2	0.1531	0.185	-702.5575	ARDL(2,2)
5	2	1	0.1551	0.1859	-703.8352	ARDL(2,1)
6	1	1	0.1551	0.1871	-706.7744	ARDL(1,1)
7	1	2	0.1547	0.1882	-705.8745	ARDL(1,2)
8	1	3	0.1537	0.1889	-705.0499	ARDL(1,3)
9	4	0	0.1586	0.1896	-691.7445	ARDL(4,0)

Current best model by RMSE across GPT models: gpt-5-nano

Figure 6.2.2.3 GPT-5 Lag Selection Results for Public Bank

This indicates that the model relies only on one lag of the stock price, while no lagged sentiment variables are included. The selection of ARDL(1,0) suggests that sentiment does not provide additional explanatory power for the Public Bank dataset in the current configuration.

This result is consistent across all GPT models, further confirming that sentiment information does not contribute significantly to the model performance for this particular stock.

6.2.2.4 Error Correction Model and Lag removal

Unlike the Maybank case, the Error Correction Model (ECM) cannot be constructed for Public Bank.

As discussed in Chapter 3, ECM requires at least one lag in both the dependent variable and the explanatory variable. In this case, the selected ARDL(1,0) model has no lagged sentiment variable (q = 0), which violates the necessary condition for ECM construction. Therefore, no ECM analysis is performed.

Since no sentiment lags are included in the selected ARDL model, statistical significance testing and lag removal analysis are not applicable for the Public Bank case.

The absence of sentiment variables in the final model suggests that sentiment does not have a statistically significant contribution to explain stock price movements for Public Bank within the scope of this study. This differs from the Maybank results, where sentiment variables are included but often found to be statistically insignificant.

This highlights the Public Bank stock change does not rely on news sentiment, where the absence of significant sentiment lag effects prevents further investigation of long-run equilibrium relationships between sentiment and stock prices.

6.2.3 Web-Based User Interface

The Web-Based User Interface is developed to provide an interactive platform for users to explore the forecasting results and analytical outputs generated by the system. The interface integrates data processing, sentiment analysis, and ARDL modeling into a single dashboard, allowing users to easily configure inputs and visualize outputs.

As the figure below, the interface provides several configuration options, including the selection of bank (Maybank or Public Bank), GPT model (GPT-5, GPT-5-mini, GPT-5-nano), and news source (individual sources or combined dataset). These options allow users to dynamically adjust the input parameters and observe how different configurations affect the forecasting results.

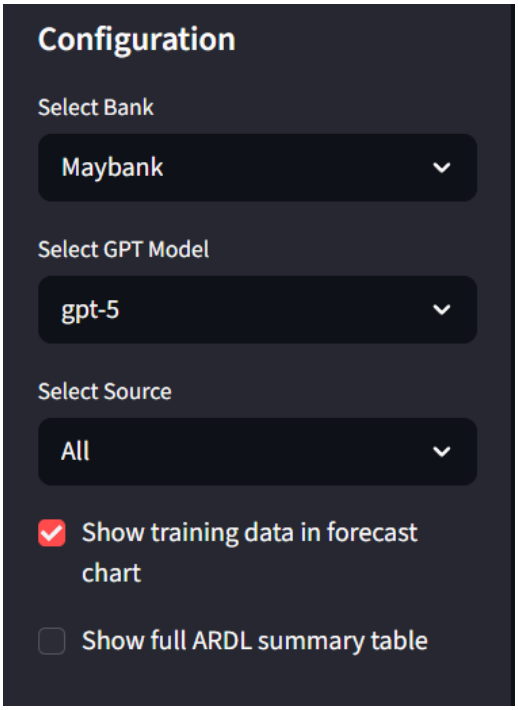


Figure 6.2.3.1 Configuration Panel of the Web-Based Interface

In addition to model configuration, the interface includes optional display settings such as showing training data in the forecast chart and displaying the full ARDL summary table. These options enhance the flexibility of the system by allowing users to control the level of detail presented in the analysis.

Once a configuration is selected, the system generates multiple outputs within the same interface. These include stock closing price trends, sentiment trends, and the ARDL forecasting results. The integration of these components enables users to compare actual and predicted values, observe sentiment patterns, and analyze model performance in a unified view.

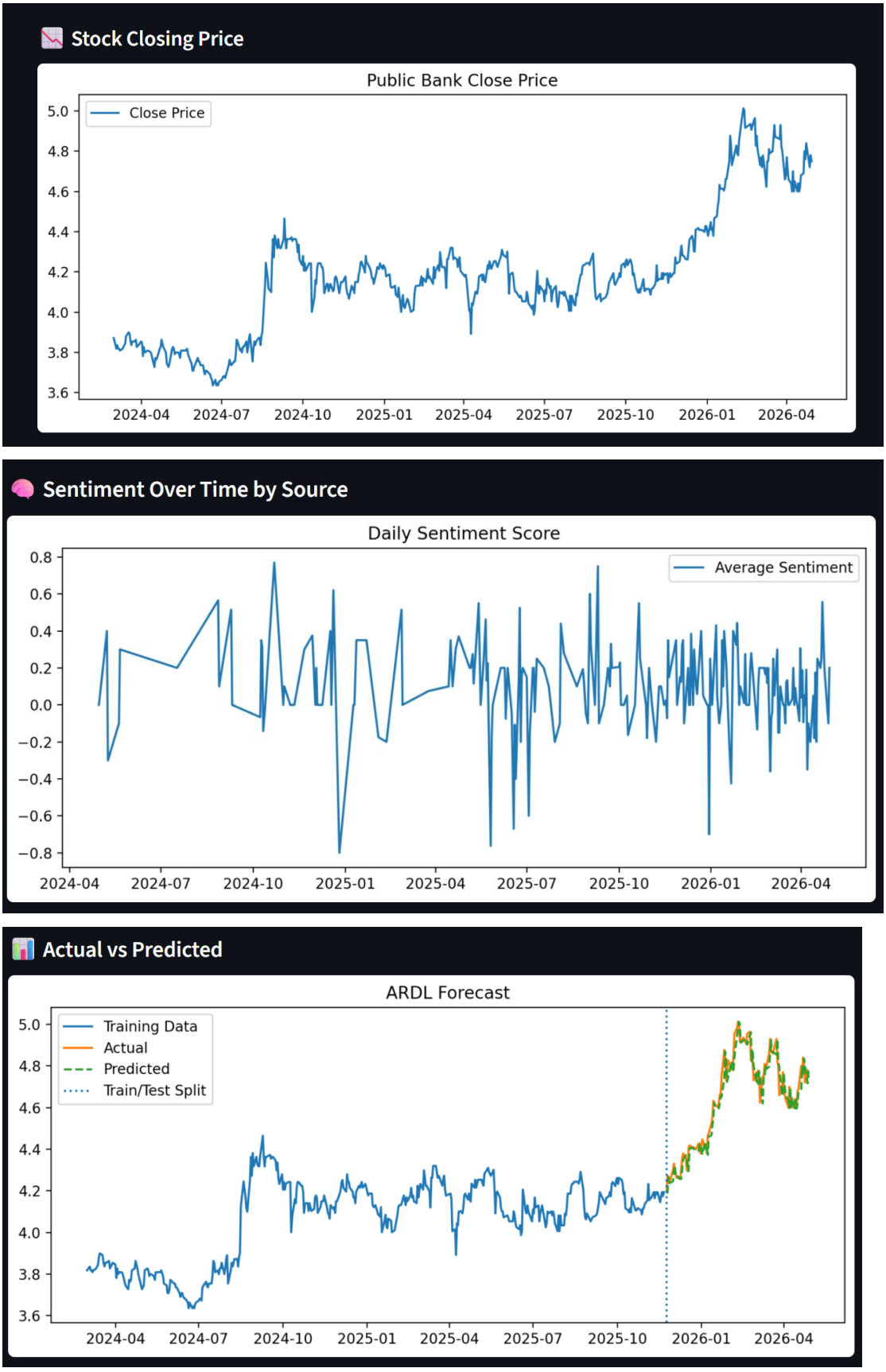


Figure 6.2.3.2 Stock Closing Price Trends, Sentiment Trends, and the forecasting results in the UI

Overall, the Web-Based User Interface enhances the interpretability and accessibility of the forecasting system by transforming complex analytical outputs into an interactive and user-friendly format. This allows users to better understand the relationship between sentiment and stock price movements without requiring in-depth technical knowledge.

6.3 Impact of Different News Sources

The system allows sentiment data to be generated from different news sources, including The Star, Yahoo Finance, Free Malaysia Today, and i3Investor, as well as a combination of all sources. This provides an opportunity to observe how different sources influence forecasting results.

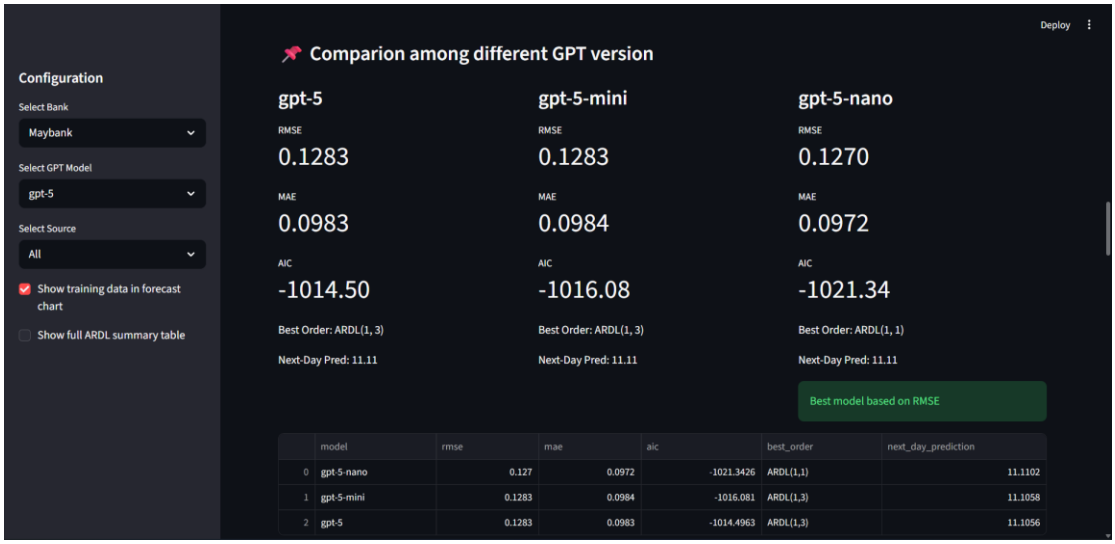


Figure 6.3.1 Performance Metrics when news from all sources are included

When all news sources are combined, the model produces a balanced sentiment signal that reflects a mixture of perspectives. This often results in stable forecasting performance.

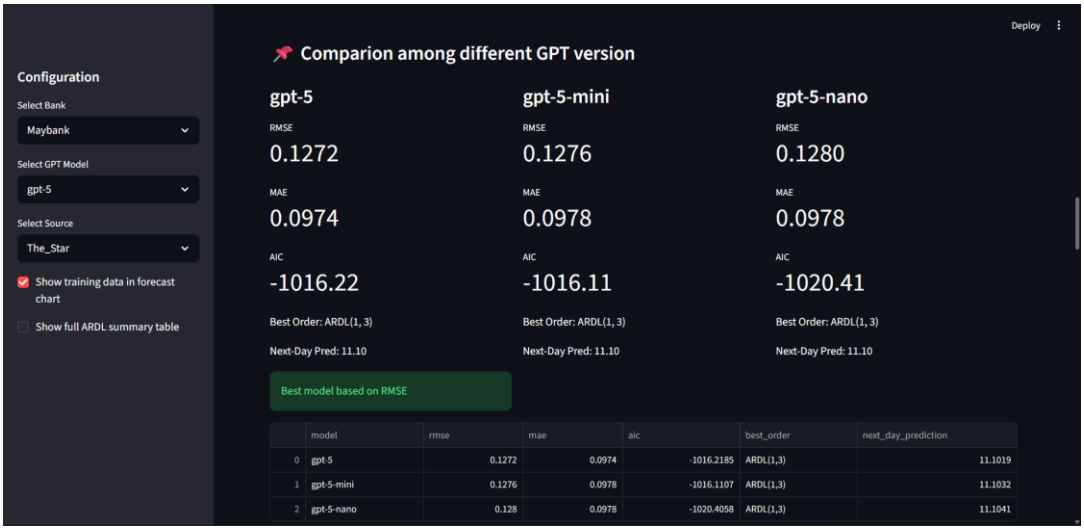


Figure 6.3.2 Performance Metrics when only news from The Star are included

When using only The Star as the sentiment source, the forecasting results show slight variation in RMSE, MAE, and lag selection. This suggests that individual news sources may capture specific aspects of market sentiment.

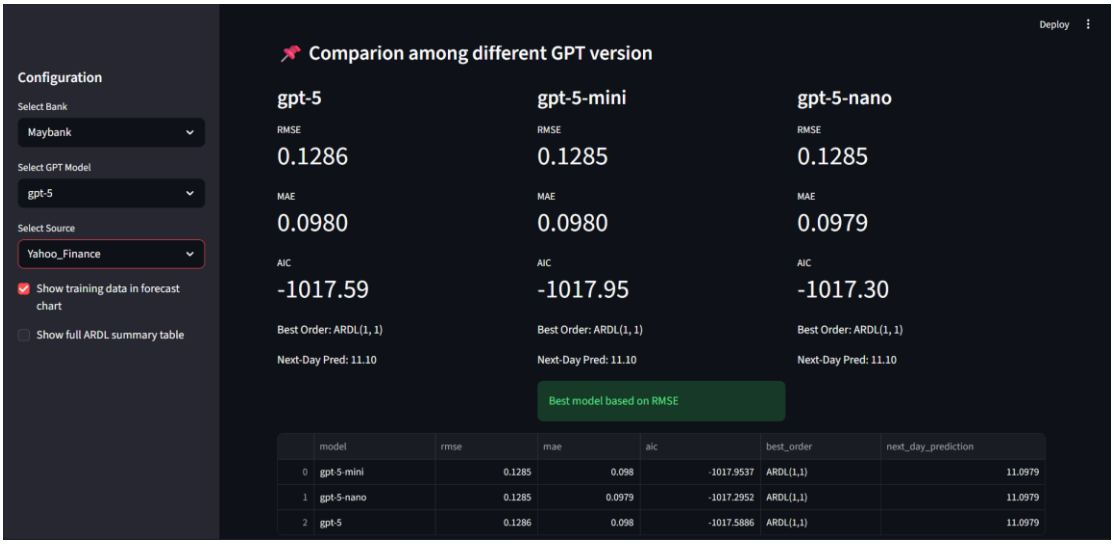


Figure 6.3.3 Performance Metrics when only news from Yahoo Finance are included

Similarly, using Yahoo Finance produces different prediction results and model configurations. This indicates that financial-focused news sources may provide sentiment signals that differ from general news platforms.

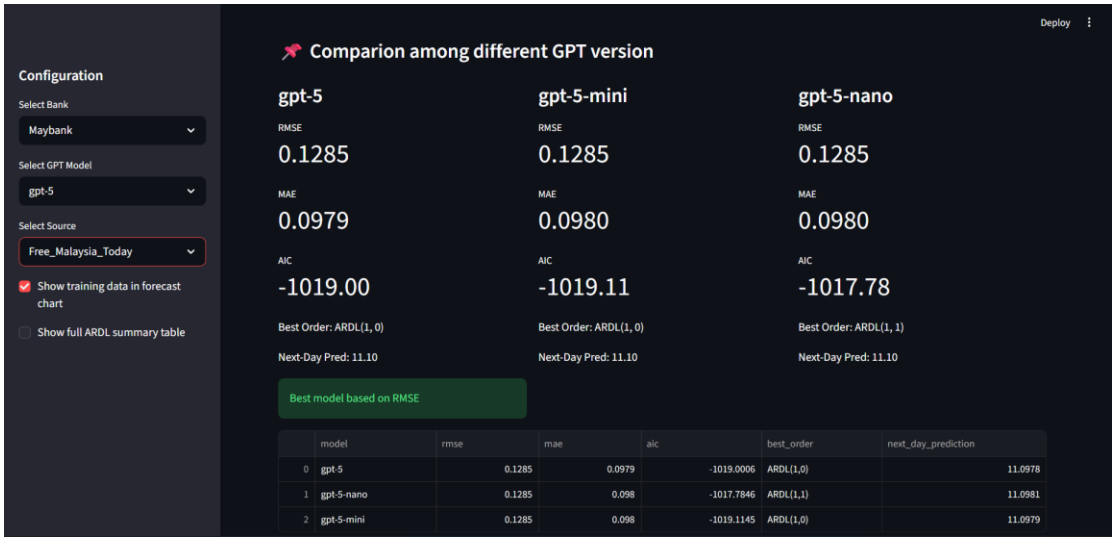


Figure 6.3.4 Performance Metrics when only news from Free Malaysia Today are included

The results from Free Malaysia Today also show variation in model performance and lag structure. This further supports the observation that different sources contribute different sentiment characteristics.

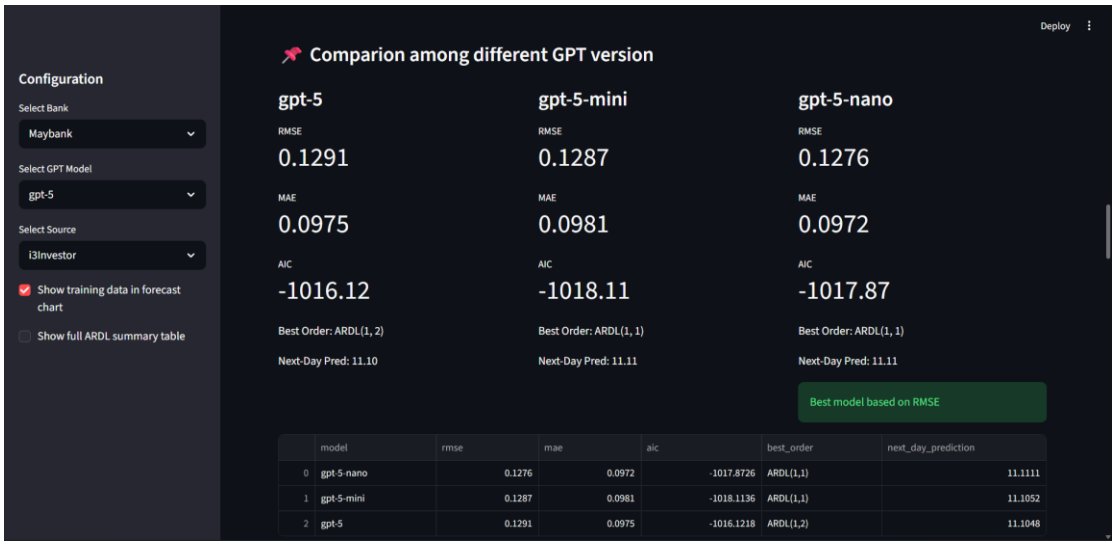


Figure 6.3.5 Performance Metrics when only news from i3Investor are included

For i3Investor, which is more investor-oriented, the sentiment signals may reflect market opinions rather than formal news reporting, leading to different forecasting behavior.

Overall, the results indicate that different news sources produce varying sentiment signals, which in turn influence ARDL model performance and lag selection. Based on the observed results, it can be seen that the performance differences across news sources are

relatively small, with RMSE and MAE values remaining within a close range. This suggests that no single news source consistently outperforms the others in terms of forecasting accuracy.

However, certain tendencies can still be observed. When all news sources are combined, the model tends to produce more stable results, as the sentiment signal reflects a broader range of perspectives. In contrast, using individual sources may lead to slight variations in performance and lag selection, indicating that each source captures different aspects of market sentiment.

Despite these observations, the differences are not sufficiently large or consistent to conclude that one specific news source is superior. Therefore, this study does not identify a single best-performing source, but instead highlights that combining multiple sources may provide a more balanced and stable sentiment signal.

6.4 Objectives Evaluation

The first objective, which is to perform sentiment analysis on financial news data, is successfully achieved. The system is able to collect news data from multiple sources, including Yahoo Finance, The Star, Free Malaysia Today, and i3Investor, and generate sentiment scores using different GPT models (GPT-5, GPT-5-mini, and GPT-5-nano). The generated sentiment scores are standardized on a continuous scale from -1 to 1 and successfully integrated into the data pipeline. This demonstrates that unstructured textual data can be effectively transformed into structured numerical features for further analysis.

The second objective, which is to implement ARDL regression for predicting stock trends, is also successfully achieved. The ARDL model is able to produce reliable short-term forecasts, as indicated by low RMSE and MAE values and the close alignment between predicted and actual stock prices. In addition, the model supports lag selection and dynamic analysis, allowing the system to capture temporal dependencies between stock prices and sentiment variables. The implementation of ARDL confirms its suitability as a forecasting approach for time-series data in this study.

The third objective, which is to evaluate the performance of different GPT models on stock prediction, is successfully achieved. The experimental results show that different GPT models produce slightly different sentiment scores, which in turn influence lag selection and model structure in certain cases. However, the overall

forecasting performance remains relatively consistent across all models, indicating that while sentiment generation varies across models, its impact on prediction accuracy is limited. This finding provides valuable insight into the role of sentiment models in financial forecasting systems.

The fourth objective, which is to develop a web-based user interface for stock prediction based on news sentiment, is successfully achieved. The developed interface allows users to configure input parameters, including bank selection, GPT model, and news source, and dynamically visualize forecasting results. The system successfully integrates front-end interaction with back-end data processing and modeling components, demonstrating a complete end-to-end implementation of the proposed system.

Overall, all research objectives are achieved. The system demonstrates strong capability in data processing, forecasting, and system integration. However, the findings also reveal that the contribution of sentiment variables to stock prediction is limited in certain scenarios, particularly when sentiment lags are not selected or are statistically insignificant. This suggests that while sentiment analysis provides additional contextual information, its effectiveness in stock prediction may depend on factors such as data characteristics, lag structure, and model configuration.

6.5 Concluding Remark

This chapter evaluates the performance of the proposed sentiment-based stock forecasting system and discusses the key findings derived from the experimental results. The system demonstrates strong forecasting capability in terms of prediction accuracy, as indicated by low RMSE and MAE values and the close alignment between predicted and actual stock prices.

However, the results also reveal that sentiment variables are generally not statistically significant in many cases, suggesting that their direct explanatory power in predicting short-term stock price movements is limited. This indicates that stock price dynamics may be influenced more strongly by historical price patterns than by sentiment signals alone.

In addition, the analysis highlights several important insights. The results show that smaller language models, such as GPT-5-nano, can produce unstable outcomes, with variations in lag selection, statistical significance, and prediction performance

when the dataset is slightly updated. This suggests that model robustness is an important consideration when using LLM-based sentiment analysis.

Furthermore, lag removal experiments demonstrate that simplifying the ARDL model can sometimes improve forecasting performance, indicating that not all lag variables contribute equally and that excessive model complexity may introduce noise. The analysis of different news sources also shows that sentiment signals vary across sources, although no single source consistently outperforms others.

Overall, the evaluation confirms that the proposed system is effective as a forecasting tool and provides valuable insights into the role of sentiment in financial prediction. At the same time, the findings highlight the limitations and challenges of using sentiment analysis, emphasizing the need for careful model design, robust sentiment generation, and consideration of additional factors in stock market forecasting.

CHAPTER 7

CONCLUSION AND RECOMMENDATION

7.1 Conclusion

This project presents a sentiment-based stock trend forecasting system that integrates financial news analysis with time-series modeling. The system begins by collecting financial news data from multiple sources, including Yahoo Finance, The Star, Free Malaysia Today, and i3Investor. The collected textual data is then processed using GPT-based models (GPT-5, GPT-5-mini, and GPT-5-nano) to generate sentiment scores on a continuous scale ranging from -1 to 1. These sentiment scores are aligned with daily stock closing prices and used as input features for the forecasting model.

The forecasting component is implemented using the Autoregressive Distributed Lag (ARDL) model, which captures both the temporal dynamics of stock prices and the lagged effects of sentiment variables. Expanding-window cross-validation is applied to determine the optimal lag structure, and the model performance is evaluated using RMSE, MAE, and AIC. In addition, an Error Correction Model (ECM) is constructed in applicable cases to analyze the long-run relationship between sentiment and stock prices. A web-based user interface is also developed to allow users to interactively explore the forecasting results and model outputs.

The experimental results reveal several important findings regarding the relationship between textual sentiment and financial time series. One of the key findings is that the effectiveness of sentiment-based forecasting is highly dependent on the characteristics of the generated sentiment data rather than the forecasting model itself. Although all three GPT models used in this study (GPT-5, GPT-5-mini, and GPT-5-nano) share the same Transformer-based architecture, they produce significantly different sentiment outputs due to differences in parameter size, training data richness, and optimization objectives , which is consistent with findings in prior research on large language models [17].

In particular, larger models such as GPT-5 demonstrate stronger capability in capturing nuanced financial language, including contextual cues such as contrast (“but”), uncertainty, and mixed sentiment expressions, which has also been observed in previous studies on large-scale language models [18]. As a result, the generated

sentiment series exhibits higher variability and richer informational content. In contrast, smaller models such as GPT-5-nano tend to produce more neutral and compressed sentiment outputs, resulting in lower variance and reduced signal strength.

These differences in sentiment generation lead to distinct statistical properties in the sentiment time series, which directly affect the performance of the ARDL model. Since ARDL relies on variations in explanatory variables to detect relationships, sentiment series with higher variance and information content are more likely to produce meaningful lag structures and stronger predictive performance. Conversely, sentiment series with low variability may result in weaker explanatory power and different lag selections.

Another important observation is that the impact of sentiment on stock prices is generally weak and often statistically insignificant across different lag structures. In many cases, sentiment variables are found to be insignificant regardless of whether they are included at the current period or at lagged intervals. For example, in the case of Maybank using the GPT-5 model with all news sources, the selected ARDL(1,3) model shows that all sentiment lag coefficients have p-values greater than 0.05, indicating a lack of statistical significance.

This finding implies that the relationship between news sentiment and stock prices may be weaker than expected, or that sentiment alone is insufficient to explain short-term price fluctuations. This observation is consistent with prior studies which suggest that sentiment variables may not always exhibit strong statistical significance in financial forecasting models [19]. It also reflects the complexity of financial markets, where stock prices are influenced by multiple factors beyond textual sentiment, such as macroeconomic conditions, firm fundamentals, and investor behavior.

For Public Bank, the ARDL model results further support the observation that sentiment has limited influence on stock price movements. Across different configurations, the selected model is consistently ARDL(1,0), indicating that no sentiment lag is included in the final specification.

This suggests that sentiment variables are not selected as relevant predictors within the ARDL framework for Public Bank, as they do not improve model performance during the lag selection process. In contrast to Maybank, where sentiment variables are included but found to be statistically insignificant, the Public Bank results indicate that sentiment is excluded entirely due to its lack of contribution to the model.

Furthermore, the Error Correction Model (ECM) analysis shows that the Error Correction Term (ECT) is negative and statistically significant (e.g., ECT ≈ -0.0288, p-value < 0.01), indicating the presence of a long-run equilibrium relationship between the variables. The magnitude of the ECT suggests that approximately 2.8% of the deviation from equilibrium is corrected in each period, implying a relatively slow adjustment process. However, the adjustment speed is relatively slow, suggesting that the relationship between sentiment and stock prices is relatively weak and influenced by other external factors.

Overall, the findings suggest that sentiment analysis can provide useful signals for short-term forecasting, but its effectiveness depends heavily on the quality and variability of the sentiment data. The study highlights the importance of model selection, lag structure analysis, and careful interpretation of sentiment signals in financial forecasting applications.

7.2 Recommendation

Based on the findings of this study, several recommendations are proposed for future improvement and research.

Firstly, future work should focus on improving the quality and variability of sentiment data. Since the effectiveness of the forecasting model is highly dependent on the statistical properties of the sentiment series, more advanced sentiment generation approaches should be explored. This may include combining multiple models, applying domain-specific fine-tuning, or incorporating additional textual features to enhance information richness.

Secondly, further analysis can be conducted on the statistical properties of sentiment time series. Techniques such as variance analysis, autocorrelation analysis, and information entropy measurement can be used to better understand how sentiment signals influence model performance. This can provide deeper insights into the relationship between sentiment and financial data.

Thirdly, the forecasting framework can be extended by integrating alternative modeling approaches. While ARDL provides a strong baseline for linear time-series analysis, future studies may explore nonlinear models such as machine learning or deep learning approaches to capture more complex relationships.

In addition, the system can be enhanced by incorporating additional explanatory variables, such as trading volume, technical indicators, or macroeconomic factors. This can help improve the robustness and predictive performance of the model.

Finally, the system can be further developed into a scalable and real-time application. Deploying the system in a cloud environment and integrating live data streams would allow continuous forecasting and broader practical usage.

REFERENCES

[1] M. Baker and J. Wurgler, “Investor Sentiment in the Stock Market,” Journal of Economic Perspectives, vol. 21, no. 2, pp. 129–151, Apr. 2007, doi: https://doi.org/10.1257/jep.21.2.129.

[2] P. Tetlock, “Giving Content to Investor Sentiment: The Role of Media in the Stock Market,” The Journal of Finance, vol. 62, no. 3, pp. 1139–1168, May 2007, doi: https://doi.org/10.1111/j.1540-6261.2007.01232.x.

[3] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” Journal of Computational Science, vol. 2, no. 1, pp. 1–8, Mar. 2011.

[4] T. B. Brown et al., “Language Models Are Few-Shot Learners,” arxiv.org, vol. 4, no. 33, May 2020, Available: https://arxiv.org/abs/2005.14165

[5] Q. Xiao and B. Ihnaini, "Stock trend prediction using sentiment analysis," in PeerJ Computer Science, Mar. 2023, vol. 9, p. e1293, doi: https://doi.org/10.7717/peerj-cs.1293.

[6] K. Joshi, B. H. N, and J. Rao, "Stock Trend Prediction Using News Sentiment Analysis," in International Journal of Computer Science and Information Technology, vol. 8, no. 3, Jun. 2016, pp. 67–76, doi: http://dx.doi.org/10.5121/ijcsit.2016.8306.

[7] S. Paul and S. Vishnoi, "Real-Time Stock Trend Prediction via Sentiment Analysis of News Article," in SSRN Electronic Journal, 2020, doi: https://dx.doi.org/10.2139/ssrn.3753015.

[8] S. V. Kolasani and R. Assaf, "Predicting Stock Movement Using Sentiment Analysis of Twitter Feed with Neural Networks," in Journal of Data Analysis and Information Processing, vol. 08, no. 04, 2020, pp. 309–319, doi: https://doi.org/10.4236/jdaip.2020.84018.

[8, Chakraborty et al. 1-6] Chakraborty, P., Pria, U.S., Rony, M.R.A.H. and Majumdar, M.A. (2017) Predicting Stock Movement Using Sentiment Analysis of Twitter Feed. 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), Himeji, 1-3 September 2017, 1-6. doi: https://doi.org/10.1109/ICIEV.2017.8338584.

[9] C. Shao and X. Chen, "Deep-Learning-Based Financial Message Sentiment Classification in Business Management," in Computational Intelligence and Neuroscience, vol. 2022, Jul. 2022, pp. 1–9, doi: https://doi.org/10.1155/2022/3888675.

[10] W. Kang, X. Yuan, X. Zhang, Y. Chen, and J. Li, "ChatGPT-based Sentiment Analysis and Risk Prediction in the Bitcoin Market," in Procedia Computer Science, vol. 242, Aug. 2024, pp. 211–218, doi: https://doi.org/10.1016/j.procs.2024.08.258.

[11] Jan Ole Krugmann and J. Hartmann, "Sentiment Analysis in the Age of Generative AI," in Customer Needs and Solutions, Mar. 2024, vol. 11, no. 1, doi: https://doi.org/10.1007/s40547-024-00143-4.

[12] W. Zhang, Y. Deng, B. Liu, S. J. Pan, and L. Bing, "Sentiment Analysis in the Era of Large Language Models: A Reality Check," in arXiv.org, May 2023, doi: https://doi.org/10.48550/arXiv.2305.15005.

[13] T. Thogesan, A. Nugaliyadde, and K. W. Wong, "Integration of Explainable AI Techniques with Large Language Models for Enhanced Interpretability for Sentiment Analysis," in arXiv.org, Apr 2025, doi: https://doi.org/10.48550/arXiv.2503.11948

[14] Bashtage. (Apr. 2025). Autoregressive Distributed Lag (ARDL) models - statsmodels 0.15.0 (+617). Presented at GitHub. [Online]. Available: https://www.statsmodels.org/devel/examples/notebooks/generated/autoregressive_distributed_lag.html

[15] A. Alawajee, M. T. Ismail, and W. Alanazi. (Dec 2023). Assessing the Effect of Investor Sentiments on Housing and Stock Returns in Saudi Arabia: Application of Nonlinear ARDL Modelling. Presented at Kurdish Studies. [Online]. Available: https://kurdishstudies.net/menu-script/index.php/KS/article/view/1094

[16] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008, Available: https://www.cs.cornell.edu/home/llee/omsa/omsa.pdf

[17]J. Wei et al., “Emergent Abilities of Large Language Models.” Available: https://arxiv.org/pdf/2206.07682

[18]J. Kaplan et al., “Scaling Laws for Neural Language Models,” 2020. Available: https://arxiv.org/pdf/2001.08361

[19]C. Kearney and S. Liu, “Textual sentiment in finance: A survey of methods and models,” International Review of Financial Analysis, vol. 33, pp. 171–185, May 2014, doi: https://doi.org/10.1016/j.irfa.2014.02.006.

Poster

Using Sentiment Analysis to Forecast Short-Term Stock Trend

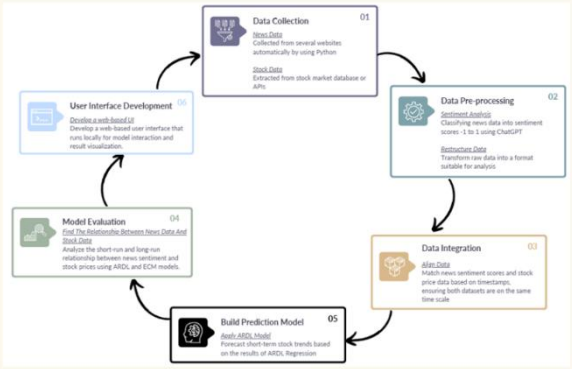
1 Introduction

This project is utilizing ChatGPT to classify and quantify sentiment from financial news and combining these sentiments with historical stock data.

2 Research Objectives

- To perform sentiment analysis on financial news data.
- To implement ARDL regression for predicting stock trends.
- To evaluate the performance of different GPT models on stock prediction.
- To develop web-based UI for stock prediction based on news.

3 System Development Flow



4 Conclusion

While these objectives represent the core outcomes of this study, FYP1 has primarily focused on building the foundation to achieve the objectives in FYP2, though none of the objectives have been achieved

sss