INTEGRATING NATURAL LANGUAGE PROCESSING (NLP)
FOR ENHANCED STOCK MARKET PREDICTION
TRHOUGH TEXT AND NEWS DATA FUSION

BY

ONG DUN YI

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 2025

Ex

COPYRIGHT STATEMENT

This Final Year Project report 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 report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report 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.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor who has given me this bright opportunity to engage in researching and developing a model for stock market prediction project. It is my first step to establish a career in AI and fin-tech field.

Other than that, I would like to thanks to a very special community, open-source community, for all the contributors' profound knowledge, and for their willingness to share their knowledge to public selflessly for free. They are the ones who improve the global living quality by making the technology becoming better and accessible. Without them, AI technique would not be evolved to today's level. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This study addresses critical challenges in stock market forecasting by introducing FusionStockBERT, a transformer-based, multi-modal model that predicts both next-minute price movement direction (up/down) and expected return — the latter reframing traditional stock price regression into a more stable return regression task. Existing NLP methods in stock market prediction faced limitations from the context drift, extrapolation errors, out-of-vocab issue in lexicon, growth issue of dictionary size and the inability to capture complex textual semantics when being applied to the stock market movement prediction task. To address these issues, we proposed a self-supervised learning to further fine-tune the FinBERT (a pre-trained BERT model) directly on directional movement labels and augment its final [CLS] representation with engineered trading features via an intermediate neural fusion layer for downstream tasks.

Evaluated on minute-level Bloomberg news transcripts paired with trading data, FusionStockBERT achieves 80.53% accuracy on the development set and 71.42% on a held-out validation set for directional movement prediction—substantially outperforming both non-attention CNN baselines and majority-class baselines. In the return regression task, it delivers an MSE of 8.9×10^{-4} on the validation set, demonstrating competitive precision in estimating the directional movement's return. These results highlight that integrating fine-tuned transformer embeddings with structured market data provides a powerful, real-time tool for high-frequency trading decision support.

Area of Study: Natural Language Processing in Deep Learning

Keywords: Stock Market Movement Prediction, Integration of NLP in Stock Market Prediction, FinBERT Fine-tuning, Multi-Modal Data Fusion, Integration of BERT in Stock Market Directional Movement Prediction.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\oplus$	Element-wise Addition

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
CNN	Convolutional Neural Network
NN	Neural Network
LSTM	Long Short-Term Memory (A Type of Recurrent Neural Network)
MSE	Mean Squared Error
MCC	Matthews Correlation Cofficient

CHAPTER 1 INTRODUCTION

Stock market prediction remains one of the most challenging problems in quantitative finance. As prices respond not only to measurable indicators—such as trading volume, past returns, and macroeconomic statistics—but also to qualitative influences like investor sentiment, central bank communications, breaking news, and shifts in government policy. While traditional time-series models capture historical price dynamics, they often overlook the rich, real-time signals embedded in unstructured text.

In recent years, the integration of natural language processing (NLP) into financial forecasting has shown promise. For example, one study found that incorporating transcribed central bank speeches into a machine learning pipeline significantly improved the prediction of the market turbulence in next three-month [1]. Similarly, transformer-based language models pretrained on financial corpora—such as FinBERT—have demonstrated strong performance on sentiment classification tasks, outperforming lexicon-based methods across benchmarks like PhraseBank and AnalystTone [2]. Yet translating these advances into accurate, high-frequency predictions of price direction remains an open challenge.

At the same time, the emergence of high-frequency trading has heightened demand for ultra-short-term forecasts, where even minute-level price movements can yield substantial gains or losses. In our analysis, the existing approaches to stock market prediction fall into three broad categories—sentiment analysis, price regression, and directional movement—but each faces its own limitations in predicting the stock price movement. Sentiment models can suffer from context drift when applied to predict price movements; price-regression models struggle with extrapolating beyond historical ranges; and early movement-classification efforts rely on static dictionaries or non-attention neural network architectures that fail to capture nuanced linguistic context.

Against this backdrop, our work leveraged the power of transformer-based embeddings and multi-modal with data fusion to deliver robust directional movement and return predictions.

Our proposed model, FusionStockBERT, achieved a significant breakthrough in the stock market movement prediction task—attaining 80.53% accuracy on the development set and 71.42% accuracy on the validation set. The remainder of this chapter outlines the key problems and motivations that shaped our approach, followed by a clear presentation of the research objectives, project scope, contributions, and the organization of this report.

1.1 Problem Statement

In our analysis, the existing approaches to the stock market prediction can be categorized into three tasks, the sentiment analysis task, the price-regression task, and the directional movement classification task.

Sentiment analysis is an NLP sequence task focused on capturing human feelings, preferences, and opinions. A novel prior study in the stock market sentiment task is the FinBERT, a BERT model pretrained on financial corpora and fine-tuned for analyst-rating classification. It achieved a great performance in PhraseBank, FiQA and AnalystTone sentiment tasks [2]. However, because it is optimized for sentiment labels (positive, neutral, negative) rather than directional price movements (up, down), we argue that applying FinBERT directly to market-movement prediction risks context drift [3]. As analyst ratings reflect subjective judgments that vary by individual expertise and may not align with actual price changes. Without additional fine-tuning on movement-specific data, FinBERT's sentiment representations alone are unlikely to deliver reliable predictions of stock price direction.

While sentiment analysis focuses on interpreting human opinions and feelings from text, the stock price regression task aims to predict exact future prices—a challenge that has attracted extensive research and yielded generally strong results. Many approaches achieve low Mean Squared Error (MSE) and high R^2 by combining historical price series with auxiliary features. A notable example is the SA-LSTM [4], which augments an LSTM with emotion features extracted via an E-CNN and retrains on a rolling three-year window with a three-month stride , outperforming its peers in the stock price regression domain.

However, despite these strong results, such price-regression models suffer from a fundamental extrapolation limitation [5, 6]. Such that in the [7, 8] studies, they struggled to predict values that lie outside their training range. Since stocks can trade across vastly different price bands under varying market conditions—and because no asset has fixed bounds—a model trained on one price interval must be retrained continually to maintain its accuracy. For example, a model calibrated on Stock A's historical prices of 100–200 will likely mispredict Stock B, which might trade around 20, 500, or 1,000; and even Stock A itself may exceed its original training bounds in the future. This need for constant, large-scale retraining makes pure price regression impractical for robust, long-term forecasting.

By contrast, the stock market movement prediction task reframes forecasting as a directional classification problem — predicting whether the price will move up or down — rather than estimating exact values. This reframing inherently sidesteps the extrapolation issues seen in price regression, since classification depends on relative changes rather than absolute price levels. Unfortunately, the prior research in this area remains largely confined at the early stage of traditional NLP approach, and only a handful of studies explored the use of the deep learning approach.

Traditional NLP approaches to stock directional movement prediction have predominantly relied on dictionary-based techniques and related rule-based methods [1, 9]. In these approaches, sentiment lexicons such as the Loughran–McDonald financial dictionary or general-purpose tools like VADER are used to assign each word or phrase with a fixed numerical score (e.g., –5 to +5). These scores are then aggregated to form as an input feature vector for fitting the machine learning models. While these models are attractive for their interpretability and ease of implementation, they often overlook the complexity of natural language. As [9] cautioned, relying solely on unigrams cannot account for shifts in sentiment when words form phrases or when negations are present. Extending the lexicon to cover higher-order n-grams would mitigate this issue but it would also rapidly inflate both vocabulary size and the operational complexity of look-ups and maintenance.

Moreover, any term not explicitly listed in the lexicon—such as emerging jargon, company names, or ticker symbols—is effectively treated as out-of-vocabulary (OOV), resulting in information loss. Financial texts are rife with novel or domain-specific terms, and maintaining a comprehensive, up-to-date dictionary demands significant manual effort while slowing real-time inference. These constraints make static lexicons ill-suited to dynamic, high-frequency trading environments.

In the realm of deep learning approaches, a notable past study employed a non-attention CNN model to learn how word and event embeddings from financial news relate to market directional movement [10]. The integration of pretrained embeddings led to significant performance gains compared to traditional techniques. However, the CNN architecture itself has inherent limitations in capturing contextualized word relationships. As it lacks an attention mechanism [11], CNNs process words in isolation and cannot model how one word influences another across a sentence. Without the ability to dynamically attend to relevant parts of the input, the model extracts features solely based on static embedding representations, missing the subtle dependencies and contextual cues essential in complex financial language. This lack of contextual understanding restricts the CNN's capability in accurately interpreting complex text, ultimately limiting its performance in high-level NLP tasks.

1.2 Motivation

The limitations identified across sentiment analysis, price regression, and market movement prediction highlight the need for a unified, context-aware, and multi-modal framework—one capable of learning from both unstructured textual data and structured quantitative trading features.

Despite rapid advances in natural language processing, the stock market movement prediction task has yet to fully leverage the capabilities of modern transformer-based models. Transformers [12] have become the dominant sequential modelling approach across NLP tasks since their first release by surpassing the earlier models like ConvS2S [13], a CNN model with attention mechanisms. Their self-attention mechanism allows for deep contextual understanding by modelling relationships between all words in a sentence, regardless of distance. This makes them exceptionally well-suited to interpret complex and noisy text — something traditional dictionary methods and non-attention CNNs cannot achieve.

Moreover, transformer-based models employ sub-word tokenization, which enables them to handle new or unseen words without requiring an ever-expanding vocabulary. During pretraining, these models learn optimal ways to split words into smaller sub-word units—such as WordPiece's "in" + "##complete" for the word "incomplete." At inference time, an unseen word is decomposed into familiar sub-words, each of which the model already associates with meaningful embeddings. These embeddings are then combined and contextualized across the transformer's stacked attention layers to produce a representation that captures both the word's base meaning and its usage in context.

This approach not only mitigates the out-of-vocabulary problem but also offers a compact alternative to large static dictionaries. For example, if the training data contains examples like "inability," "incomprehension," and "incompatible," the model learns that the prefix "in-" often carries a negative connotation. When it later encounters "incomplete," it can infer its negative sentiment without ever having seen the full word before. In traditional dictionary-based methods, handling such variations would require explicitly listing each word and its possible n-grams, leading to unmanageable lexicon growth and maintenance challenges. Sub-word tokenization thus provides both robustness to unseen terms and efficiency in vocabulary size—key advantages for real-world NLP applications.

Motivated by the powerful technique and performance of transformer-based models in general NLP tasks, this research explores their untapped potential in high-frequency market movement prediction. We introduce **FusionStockBERT**, a novel model that combines a pre-trained encoder-based transformer (FinBERT) with a neural network fusion module. The FinBERT was further fine-tuned in this study to adapt its contextual language representations to the downstream task of predicting next-minute of stock directional movements and order return expectation.

FusionStockBERT is designed to address key challenges in the field:

- It mitigates context drift by fine-tuning FinBERT directly on movement labels instead of relying on sentiment outputs.
- It avoids extrapolation issues inherent in price regression by focusing on directional movement classification and order return expectation, the latter being a more stable regression target with a more bounded range than raw stock prices.
- It overcomes the out-of-vocabulary and the issue of large dictionary size by leveraging the sub-word tokenization technique that used in BERT model.
- It overcomes the contextual limitations of non-attention CNN and NN models by leveraging transformer-based language modelling alongside structured trading inputs.

The designed model operates in a multi-modal setup, fusing minute-level financial news speech with real-time trading features, to predict each next minute movement and return expectation. The full scope of this research spans from raw data mining and preprocessing to the fine-tuning and deployment of the custom transformer-based architecture—offering a complete, end-to-end solution for enhanced predictive performance in high-frequency trading environments.

1.3 Objectives

The primary objective of this research is to enhance the accuracy and reliability of stock market directional movement prediction by leveraging advanced natural language processing (NLP) techniques — specifically, through the use of a fine-tuned transformer-based model and multi-modal data fusion.

To achieve this, the study is guided by the following specific objectives:

(1) To Exceed the Baseline Accuracy Threshold.

The baseline accuracy is established and set to the percentage of sample size of the largest dominant class exist in the dataset. For example, if there is 3 A and 7 B samples in the dataset, the baseline accuracy of this classification is set to 70%. This baseline reflects the accuracy of a naive model that always predicts the majority class. A valid predictive model must outperform this threshold; otherwise, it indicates that the model has failed to learn meaningful patterns beyond random or biased guessing.

(2) To Outperform the Prior State-of-Art (Non-attention CNN model).

To demonstrate the superiority of the proposed FusionStockBERT model over earlier deep learning approaches — particularly the non-attention CNN architecture — by leveraging BERT's ability to capture contextual and semantic relationships in financial text, this objective is essential to show how much the performance gains introduced by transformer-encoder based modelling in this domain.

(3) To Validate the Predictive Relationship of the Dataset.

This objective is to evaluate whether a meaningful and generalizable relationship exists between the input features (textual and trading data) and the output labels (stock directional movement and return). Since this study relies on a self-collected, primary dataset, validation of the data's predictive integrity is crucial. This is assessed by monitoring the validation loss and accuracy trends as the training dataset size increases.

A dataset is considered to exhibit a valid predictive relationship when it satisfies both of the following conditions:

(1) As the training set grows, the validation loss must decrease alongside the training loss in the return regression task.
(2) While for the directional movement task, the validation accuracy must increase in parallel with the training accuracy as the training set grows.

1.4 Project Scope & Direction

The project scope and directions of this research would be focused on the NLP in the sub-field of the deep learning. Deep learning is an advance machine learning that is aimed to learn the extraction of useful features from the raw input itself. Such, it can minimize the needs of human or algorithms in extracting features for the prediction.

Deep learning has been proven to be powerful, as it has surpassed the human-level performance in various of the tasks. For an example in the image classification, it breaks through the human baseline errors in classifying the object in the images [14]. As it can extract the low-level feature and the features of the object at precise level that even human eye cannot see them, it learns smarter than human. While in the NLP task, the LLM breaks through the average human-level baseline in 2024 for the ARC and MMLU tasks [15].

This project aims to harness the power of Natural Language Processing (NLP) in the deep learning, to delve into the intricate world of market analysis. As stock markets are influenced not only by quantitative factors but also by qualitative aspects such as public opinion, Central Bank speeches, market news, and social media etc. Understanding and quantifying the features expressed in textual data can provide valuable insights into market dynamics.

1.5 Contributions

Leveraging multi-modal setup with the advanced NLP model, we aimed to train a model that can predict the next minute of the market movement with high accuracy.

The main contributions of this paper are as the following:

- Establishing an NLP dataset that contains each minute speech of Bloomberg Financial live news and trading data, contributing to any future research.
- Exploring the potentiality of advanced deep learning model in the market movement and return tasks at shorter interval -- minute.

- Investigating the effectiveness between fine-tuning and feature-based approach for the pre-trained BERT model at intermediate fusion stage.
- Investigating the effectiveness of using transformer model versus to the prior CNN model approach.

1.6 Report Organization

This report is organized into six chapters. Chapter 1 introduces the project by presenting the problem statement, background and motivation, scope, objectives, contributions, key achievements, and the structure of the report. Chapter 2 provides a literature review of existing approaches in the field, highlighting their strengths and limitations to establish the research gap. Chapter 3 outlines the system design, detailing the data mining process and the proposed model architecture, including its underlying mathematical formulation. Chapter 4 presents the implementation of the data mining system and the chosen hyperparameters and our dataset predictive relationship reliability test. Chapter 5 evaluates the model's performance and compares the results with those of prior state-of-the-art models. Finally, Chapter 6 summarizes the research findings and concludes the study.

CHAPTER 2 LITERATURE REVIEW

In recent years, the convergence of natural language processing (NLP) and financial markets has garnered substantial attention, with numerous studies exploring the predictive potential of textual data for forecasting market behaviour. NLP has been integrated into various stock market prediction tasks, including the sentiment analysis, stock price regression, and stock directional movement classification.

In this chapter, we present a structured review of prior work across these NLP tasks within the financial prediction domain. A key criterion for inclusion in this review is the presence of a proper validation set in the experimental evaluation. We observed that many existing studies overlook this critical aspect—an omission that undermines the credibility of their findings. By excluding a validation phase, even simple models such as random forests may appear to perform well by memorizing training data, thereby failing to demonstrate true generalization capability on unseen samples.

Accordingly, this chapter focuses exclusively on studies that adopt rigorous evaluation protocols, including held-out validation, to ensure that the reported model performance reflects real-world predictive utility.

2.1 Review of the State-of-the-Art Approach in Stock Market Sentiment Analysis

Sentiment prediction is an NLP sequence task concerned with identifying and interpreting human feelings, preferences, and opinions expressed in text. Within the context of financial markets, sentiment analysis focuses on interpreting factors such as analyst ratings, investor sentiment, and emotional indicators like fear and greed. These sentiment indicators are often used to infer the psychological state of the market and guide decision-making.

A prominent study in this area is the work by Yi Yang et al. (2020) titled "FinBERT: A Pretrained Language Model for Financial Communications" [2]. The authors introduced FinBERT, an encoder-transformer model (BERT) architecture, specifically tailored for financial sentiment tasks. They experimented with both cased and uncased on different financial corpora for pretraining and investigated how these differences affected the model's ability to capture the nuanced expressions found in expert financial commentary.

Following the pretraining phase, FinBERT was fine-tuned on several financial sentiment classification tasks through transfer learning. Their model, FinBERT-FinVocab, achieved strong performance across three major benchmarks:

- 87.2% accuracy on the Financial Phrase Bank,
- 84.4% accuracy on FiQA, and
- 88.7% accuracy on AnalystTone.

These results demonstrate the effectiveness of leveraging transformer-based models for sentiment analysis in the financial domain. The superior performance of FinBERT motivates our interest in exploring its application beyond sentiment analysis, specifically in the task of stock directional movement prediction. The integration of FinBERT as the foundation of our proposed model will be further elaborated in Chapters 3 and 4, where we discuss system design and implementation.

2.2 Review of the State-of-the-Art Approach in Stock Price Regression

Stock price prediction is commonly formulated as a regression task, where the objective is to forecast a stock's next closing price based on historical trends and auxiliary signals. Numerous studies have explored this area using a variety of machine learning and deep learning approaches [4, 7, 16]. However, this review focuses specifically on works that adopted a proper validation set in their evaluation—an essential component for assessing true generalization performance.

A notable contribution in this field is the SA-LSTM model, which integrates a moving window training strategy—a blend of traditional recurrent neural networks and modern deep learning techniques [4]. The authors used an E-CNN to extract sentiment features from the top 100 most-discussed comments on financial forums and microblogs. These sentiment features were combined with structured trading data extracted via a Denoising Autoencoder (DAE) and then fed into an LSTM for temporal sequence learning. The use of moving windows enabled the model to adapt continuously to changing price dynamics, helping to mitigate extrapolation issues inherent in stock price prediction. This dynamic training approach, combined with sentiment-rich textual inputs, allowed the SA-LSTM model to achieve a Mean Absolute Percentage Error (MAPE) of just 1.10%, outperforming several baseline models that reported significantly higher MAPE values (3.91, 11.69, 2.24, 10.24, 12.43, and 3.30), largely due to the absence of enriched textual features.

Another study [16] introduced the use of a BERT-based model to generate topic-aware sentiment scores, which were then used to improve downstream stock price prediction models. Unlike traditional dictionary-based methods such as VADER, their BERT-based sentiment analysis captured richer contextual information, resulting in better predictive performance. Their best model, CNN-LSTM(Topic), achieved strong results with an RMSE of 3.509, MAE of 2.94, R^2 of 0.708, and MAPE of 2.36.

Through our own comparison of CNN-LSTM [16] with the SA-LSTM approach in [4], we observed that the moving window re-training strategy plays a critical role in improving regression performance under conditions of high market uncertainty. As stock prices fluctuate across varying and unpredictable ranges, and without periodic retraining to accommodate such changes, models become more prone to extrapolation errors. This likely contributes to the higher MAPE observed in static-trained models, as they struggle to generalize when future prices fall outside the range of historical data.

Our concerns regarding extrapolation in stock price prediction are further supported by the findings in prior studies [7, 8]. In [7], the authors conducted a long-term study spanning from 2008 to 2020, incorporating NLP-driven sentiment into various models including ARIMA, GRU, CNN, LSTM, GRU with sentiment, FinBERT, and FinBERT with prediction layers. Despite the use of sophisticated architectures, the models reported consistently high RMSE values—ranging from 370.155 to 399.128—highlighting significant prediction errors. Another study [8], using a non-NLP approach, found that LSTM models struggled when predicting stock prices in the validation set that deviated from the historical ranges in training set—further illustrating the extrapolation limitation that hinders the effectiveness of regression-based stock prediction models.

Together, these findings underscore the challenges of stock price regression and highlight the need for alternative formulations—such as directional movement prediction—that are less susceptible to extrapolation and may generalize more effectively to future market conditions.

2.3 Review of the State-of-the-Art Approach in Stock Directional Movement Classification

The stock directional movement classification task, which aims to predict whether a stock's price will move upward or downward, offers a promising reformulation of price prediction by avoiding the extrapolation issues common in regression models. However, prior research in this area remains largely confined to the early stage of traditional NLP approaches, with only a limited number of studies exploring the capabilities of deep learning architectures.

One of the most notable traditional methods is the adaptive multi-dictionary approach proposed by Anastasios Petropoulos et. al (2021) [1]. Their methodology utilized multiple sentiment dictionaries to extract 105 sentiment features per document, derived primarily from central bank speeches. These features were then used to predict the likelihood of market turbulence in three-month horizon. Their best-performing model, XGBoost, achieved 85.77% training accuracy and a recall rate of 89%. However, based on our own calculation from the reported confusion matrix, we noticed the model had a low precision of approximately 0.45, which is presumably due to class imbalance caused by the underrepresentation of turbulence events. On the validation set, the model's accuracy dropped to 66.47%, with a recall of 0.91 and a precision of just 0.1289 for the turbulence class. Despite the low precision, the consistently high recall across training and validation sets suggests a meaningful predictive relationship between central bank speeches and subsequent market turbulence.

Another traditional NLP approach is the combination of Naïve Bayes (NB) with Latent Dirichlet Allocation (LDA) for topic modelling [17]. LDA was applied to extract topics from both news headlines and article bodies, which were then used as sparse feature vectors for the NB classifier. The model achieved 49.4% accuracy in predicting stock price direction and 55.6% accuracy for market volatility. These results suggest near-random prediction in directional movement, highlighting the limitations of relying solely on topic-level features. This simplistic representation fails to capture sentiment polarity or contextual nuances. For instance, articles titled *"War is stopped"* and *"War is started"* would likely share the same topic — *"war"* — despite conveying opposite sentiments, leading to misleading or insufficient input representations.

In contrast, deep learning approaches to this task remain relatively scarce. One of the few contributions employed a non-attention CNN and neural network (NN) architecture to process event embeddings extracted from financial news [9]. Their proposed event embeddings helped models to improve performance compared to those models that used basic word-sum embeddings [18]. Their event embeddings were formed by fitting token embeddings, that were initially trained using a skip-gram, to an NTN network. The best model in this study, EB-CNN, achieved 65.08% accuracy in daily stock movement prediction on the training set — outperforming the WB-CNN (61.73%), and EB-NN (62.84%) over WB-NN (60.25%). This result showed that the embeddings that were learned through self-supervise performed better, where the prefix-EB models that were trained on event embeddings outperformed the prefix-WB models that were trained on the traditional basic word-sum embeddings.

In summary, although traditional NLP and early deep learning methods have laid foundational work in this domain, their reliance on rigid feature extraction and lack of contextual understanding restrict their potential. These limitations point to the need for more advanced sequence models, such as transformer-based models, which offer dynamic attention capabilities and deeper contextual insight—an avenue this research aims to explore.

CHAPTER 3 SYSTEM DESIGN & MODEL ARCHITECTURE

In this chapter, we explained our system design for the data mining process and the proposed model architecture.

3.1 System Hardware Requirement

The hardware for powering the system in this project, involved a computer device and a cloud service as shown in Table 3.1.1 and 3.1.2. The computer device is used for the process of data collection, EDA and model architecture building, model evaluation. While the cloud service is purposely used for fine-tuning the BERT model.

Table 3.1.1 Specifications of laptop

Description	Specifications
Model	IdeaPad Gaming 3 15ACH6
Processor	AMD R5 5600H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 3050 4GB
Memory	16GB DDR4 RAM
Storage	512GB SATA SSDs

Table 3.1.2 Specifications of cloud service

Description	Specifications
Model	P100
Processor	1 GPU, 4 vCPUs
RAM (GiB)	16 GPU, 29 CPU
Service Provider	Kaggle
Price/Hour	Free

3.2 System Design for Data Pipeline

The data pipeline system is designed to consist of two primary components: a speech collector and a trading data collector as shown in Table 3.2.1. Both components operate on a 1-minute interval. The speech collector is responsible for capturing live audio streams from Bloomberg's financial news, transcribing the audio into text using OpenAI's open-source Whisper model (small.en) [19], and storing the transcribed content in JSON format. The choice of the small.en Whisper model was based on a balance between performance and system resource constraints. In the Wall Street Journal (WSJ) English transcription task, small.en achieved a low Word Error Rate (WER) of 3.3%, which is comparable to the medium.en model and only 0.2% higher than the large Whisper model [20]. Given this minimal performance gap and the significant speed advantage, the small.en model was selected as the optimal choice for efficient and accurate transcription within our available computational capacity. While for the trading data collector, it played a role in connecting to the MooMoo API [21] to collect the real-time trading data of SPY, QQQ and DIA ETFs, and storing them into CSV file. These ETFs are represented to track of the S&P500, NASDAQ100, and Dow Jones indexes.

Table 3.2.1 Information of Data Pipeline Collectors

Collector	Source	Target Data	Interval	Timeframe	File Format	Cost
Live News Speech	YouTube: Bloomberg	Live Speech	1-minute	9:00 – 16:00 US/Eastern	Json	Free
Trading Data	MooMoo API	SPY, QQQ, DIA ETFs	1-minute	9:30 – 16:00 US/Eastern	CSV	$180

3.3 Model Architecture Design with Mathematical Formulation

In this project, our proposed FusionStockBERT model architecture integrates a pretrained BERT model – FinBERT with a Neural Network fusion on top of its last hidden state of (CLS) tokens as shown in Fig. 3.3.1. The integration of pre-trained model is to enable a better weight initialization. This allows our model to reuse any possible features, that were learned previously, are also found to be useful in the current task. Besides that, the core objective of the fusion design is to allow the model to process and learn from two distinct modalities: contextualized language features derived from market news and quantitative trading features. By applying an intermediate fusion technique, the model leverages both modalities simultaneously, enabling the network to co-adapt and refine the (CLS) representation at downstream layers for enhancing the sequence prediction in the downstream tasks.

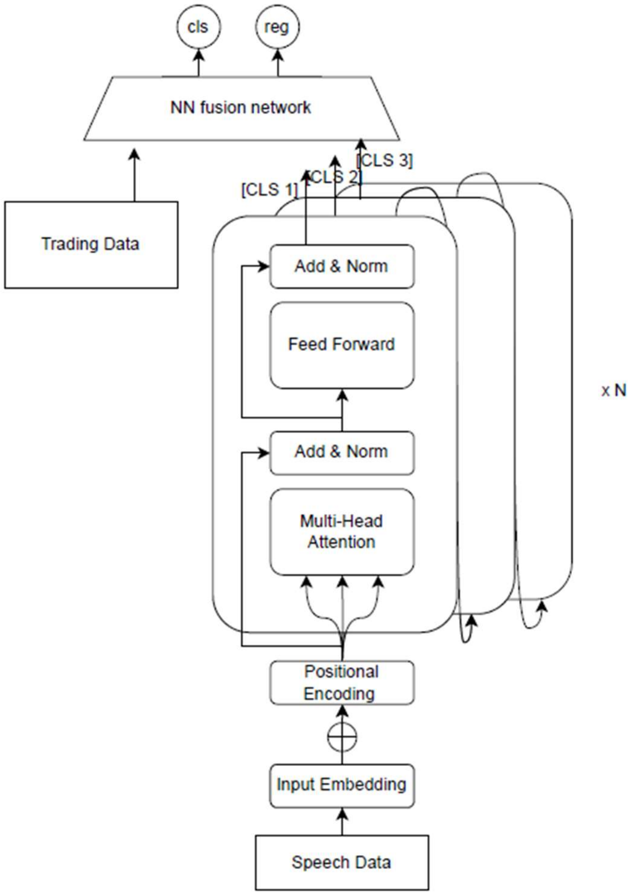


Fig. 3.3.1 Bottom-up diagram - FusionStockBERT Architecture

The inner structure of the BERT model consists of a word embeddings, a positional embeddings and N=12 stacked sequential encoders with multi-head self-attention mechanisms. Unlike the original encoder in the transformer architecture, we follow the BERT model approach that allows the positional embeddings to be trainable [22]. Besides that, we enlarge the positional embeddings to contain up to 1024 tokens, it is to allow the model to be able to operate on larger number of input tokens. As our large number of input tokens per minute is 917, without enlargement, the original number 512 of positional embeddings would fail to process the input data. While for the number of heads and input token dimensions, we followed to the same configuration as in pre-trained model -- FinBERT [23].

For the NN fusion model that extended on top of the BERT model, it is a residual network that consists of a dense layer with 0.01 negative slope setting of Leaky ReLU activation function [24] for feature extraction, a dense layer for residual mapping, and a final dense output layer for predicting the next minute movement and the next minute return expectation. The utilization of the residual technique [25] and Leaky ReLU is because of we want the model to have fully active neurons that can be tuned across all regions, removing the saturated regions as shown in Fig. 3.3.2. Such that, this allows the model to slowly judge for the signals to be negative region and cumulating as a heavy judge when the number of batches being fitted for learning increased over time and being found that most of the samples have the similar preference of direction that penalty the signals into negative region.

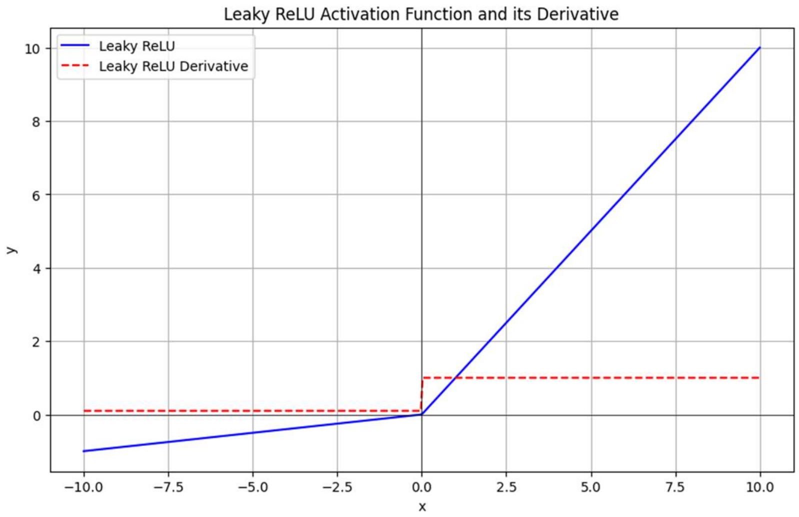


Fig. 3.3.2 Leaky ReLU and its derivative (negative slope = 0.1, for ease of viewing)

The feature extraction layer in the NN fusion model has a in_features of 768 + 11 and a out_features of 1000, and its functionality is to extract non-linear features via activation from the input features that was dynamic constructed by concatenating the last hidden-states of CLS token that has 768 of dimensions with another 11 quantitative trading features. Before the quantitative features being concatenated, they are applied with a dropout rate of 0.1 in training, as it is to help the model to simulate the possibility of encountering the loss of some quantitative data during the real-life events, exploiting the model to learn more from other available features.

The feature extraction layer can be denoted as:

$$x_{concat} = Concat(\ cls,\ Dropout_{0.1}(x_{trading})\ ) \quad (1)$$

$$y_{resf} = F_a(x_{concat}, \{W_i\}) \quad (2)$$

Such, y_{resf} is the output that has a shape of 1000 dimensions, cls is the last hidden-state of CLS token that has shape of 768 dimensions, $x_{trading}$ has 11 dimensions, W_i tensor has shape of (779, 1000), and F_a is a Leaky ReLU activation function.

The output from the feature extraction layer is then fed into a dense layer for residual mapping back into 768 + 11 features and performing the element-wise skip connection addition. After residual skip connection is performed on the 768+11 features, these strengthened features are then passed into the last dense layer to map to the downstream outputs – next minute movement and next minute return expectation.

The remaining processes after the feature extraction layer can be denoted into:

$$y_{ds} = L_2(\ L_1(y_{resf}\ , W_j) \oplus x_{concat}\ , W_{cls}) \quad (3)$$

Such, y_{ds} is the downstream output that has shape of 2 dimensions which first dimension is for next minute movement classification and the second dimension is for the next minute return regression prediction. The L_2 and L_1 are both a simple linear function without activation function attached. For the L_1 it has a W_j shape of (1000, 779) that is responsible to map the 1000 feature dimensions into 779 feature dimensions for the $\oplus$ element-wise addition on the

x_{concat} from Equation (1). The L_2 is then responsible to linear project the updated features into 2 dimension for the downstream tasks, such its W_{cls} has a shape of (779, 2).

The input flow to our FusionStockBERT model can be viewed as follow:

1. Minute of news speech being tokenized into input token ids.
2. Input token ids are padded.
3. Input token ids with its padding info are then fed into the BERT model.
4. BERT model generates the last hidden-state of the CLS token.
5. The last hidden-state of CLS token is concatenated with 11 trading features.
6. Concatenated features are then fed into the feature extraction layer in NN fusion layer.
7. The non-linear features extracted are then residual mapped to offset and strengthen the value of input features.
8. The strengthened features are then linear mapped into two downstream tasks – the Next Minute Movement and the Next Minute Return Expectation.

CHAPTER 4 IMPLEMENTATION

4.1 Data Preparation

In this research, we had successfully obtained a total 122004 minutes of trading data and 42401 thousands minutes of news speech data from the active trading days within the range from 18th July 2024 to 23rd Jan 2025 via our data mining system. Within 122004 thousands minutes of trading data, 42401 are belonging to SPY ETF, 42400 are belonging to QQQ ETF, and 37203 are belonging to DIA ETF. The attributes of trading dataset and speech dataset are shown in Tab.4.1.1 and Tab. 4.1.2.

Table 4.1.1 Attributes of Trading Data

Column Name	Description	Data Types
code	The listed code for the ETFs	String
update_time	The record time (9:30:00 - 9:30:58 is 9:31)	String
last_price	Closing price for each minute	Float
open_price	Open_price for the trading day	Float
high_price	The latest highest price reached	Float
low_price	The latest lowest price reached	Float
prev_close_price	The previous trading day’s close price	Float
volume	The latest cumulated trading volume	Int
turnover	The latest cumulated trading turnover	Float
highest52weeks_price	Latest earnings per share	Float
lowest52weeks_price	Latest price per earnings ratio	Float
pre_price	Current day’s premarket close price	Float

Table 4.1.2 Attributes of Speech Data

Column Name	Description	Data Types
record_on	The record time (9:30:00 - 9:30:58 is 9:31)	String
minute_speech	The news speech in this minute	String

The initial collected speech and trading data are being pre-processed before fitting to the model. In the pre-processing, a self-supervised labelling is performed, such that the y labels for the next minute movement and next minute return are computed for each day and each ETF from using their last price attribute, the formula for computing next minute movement and return can be denoted as Eqn. (5) and (6).

$$NextReturn(y_{t+1},\ y_t) = \left(\frac{y_{t+1}-y_t}{y_t}\right) \tag{4}$$

$$nextMinReturn(ret) = abs(ret) \tag{5}$$

$$nextMinMovement(ret) = \begin{cases} 1, & ret \geq 0 \\ 0, & ret < 0 \end{cases} \tag{6}$$

In Eqn. (4), it is to compute the next minute return without absolute, the y_{t+1} is the ETF price at $t+1$ next minute and y_t is the ETF price at t current minute. For the Eqn. (5), it takes in the next return generated by Eqn. (4) and applies absolute as a next minute return label for each minute. This Next Minute Return is to act as an order return expectation in next minute when the (Long or Short) order follows the same with the market movement. For the Eqn. (6), it takes in the next return generated by Eqn. (4) and assign a next movement label for each minute. Note that, for the last minute in each end of the trading day is set to a default 0 value, as in this project we focused on the regular trading time.

After computed the labels, the initial attributes in trading dataset are then gone through a simple transformation process into the ratio and changes rate. Such initial attributes of the trading data, shown in Table 4.1.1, are being transformed into Table 4.1.3.

Table 4.1.3 Transformed Attributes of Trading Data

Column Name	Description	Data Types
code	The listed code for the ETFs	String
update_time	The record time (9:30:00 - 9:30:58 is 9:31)	String
last_price_ChangeRate	Change rate of current price from last minute	Float
volume_ChangeRate	Change rate of current volume from last minute	Float
low_price_ChangeRate	Change rate of current low price from last minute	Float
high_price_ChangeRate	Change rate of current high price from last minute	Float
lowest52weeks_price_ChangeRate	Change rate of lowest 52 weeks price from last minute	Float
highest52weeks_price_ChangeRate	Change rate of highest 52 weeks price from last minute	Float
ratio_last_price:open_price	Ratio of last price to open price	Float
ratio_open_price:prev_close_price	Ratio of open price to previous day's close price	Float
ratio_open_price:pre_price	Ratio of open price to premarket close price	Float
ratio_last_price:lowest52weeks_price	Ratio of last price to lowest 52 weeks price	Float
ratio_last_price:highest52weeks_price	Ratio of last price to highest 52 weeks price	Float

The formula for ratio and change rate are denoted into Equation (7) and (8) respectively:

$$ratio(a_t,\ b_t) = (a_t - b_t)/b_t \qquad (7)$$

$$ChangeRate(a_t,\ a_{t-1}) = (a_t - a_{t-1})/a_{t-1} \qquad (8)$$

Such, in the ratio transformation Eqn. (7), it is to compute the difference ratio between different variables a_t and b_t at each minute time t . While for the change rate transformation Eqn. (8), it is to compute the changes rate of each a_t variable at time t with its last-minute time $t-1$. Note that, the change rate the first minute of each new trading day are set to a default 0 value.

The ratio and change rate transformation are introduced as an overcome of no self-attention applied to the trading features at NN fusion layer. The implementation of transformation algorithm for computing the ratio space relationship among the trading features is due to the numerical features are hard to form a ratio feature space in simple NN network if there is no attention mechanism applied.

After the transformation was performed on the trading features, the trading features are then performed a (M:1) joins with the speech dataset on the record time to form into a single dataset. After joined, all the ETF tickers in each minute would have the same corresponding news speech. To inject the model to learn which news speech can benefit to the prediction for which ETFs, we proposed to concatenate their ETF ticker name to the beginning of each minute news speech, as shown in Fig. 4.1.1.

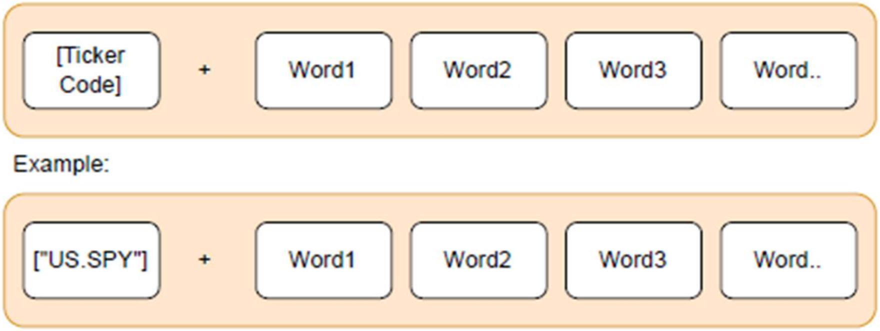


Fig. 4.1.1 Pre-processing on Minute News Speech Attribute

After completing the pre-processing stage, we performed a re-packaging step to finalize the essential input features required by our model. This was done to eliminate the need for any additional pre-processing during the model's forward pass. We utilized the finbert-tone version of the BERT tokenizer [23], with padding enabled, to pre-generate tokenized inputs for each minute of transcribed speech. The tokenization process produced three key features for each minute of news speech: input_ids, token_type_ids, and attention_mask. These pre-generated features were then stored and used directly during model inference, serving as inputs to the BERT component. For the trading data inputs to the neural network fusion layer, we selected 11 transformed trading features as illustrated in Table 4.1.3. A summary of the final input features fitting to the model is shown in Table 4.1.4.

Table 4.1.4 Input Features to the Models

Column Name	Description	Data Types
input_ids	Id that corresponds to the token embeddings in BERT	List
token_type_ids	Id that corresponds to the sequence embeddings in BERT	List
attention_mask	Masks that tell if the BERT needs to compute attention for the token position.	List
last_price_ChangeRate	Change rate of current price from last minute	Float
volume_ChangeRate	Change rate of current volume from last minute	Float
low_price_ChangeRate	Change rate of current low price from last minute	Float
high_price_ChangeRate	Change rate of current high price from last minute	Float
lowest52weeks_price_ChangeRate	Change rate of lowest 52 weeks price from last minute	Float
highest52weeks_price_ChangeRate	Change rate of highest 52 weeks price from last minute	Float
ratio_last_price:open_price	Ratio of last price to open price	Float
ratio_open_price:prev_close_price	Ratio of open price to previous day's close price	Float
ratio_open_price:pre_price	Ratio of open price to premarket close price	Float
ratio_last_price:lowest52weeks_price	Ratio of last price to lowest 52 weeks price	Float
ratio_last_price:highest52weeks_price	Ratio of last price to highest 52 weeks price	Float

4.2 Model Training Approach

This 4.2 subsection outlines our model training strategy, including details on the hyperparameter configurations.

In this research, we adopted two model training approaches: the feature-based approach and the fine-tuning approach. In the feature-based approach, all parameters of the pre-trained BERT model – FinBERT are frozen and remain untrainable in our model during training, while only the parameters in the neural network (NN) fusion layer are trainable. In contrast, the fine-tuning approach allows all parameters in both the pre-trained BERT model and the NN fusion layer to be trainable, enabling the model to update and learn the language representations on our specific task.

The dataset prepared in Section 4.1 was stratified and split based on a combination of the *Next Movement Label* and *ETF Tickers*. 80% of the data was randomly assigned to the training set, while the remaining 20% formed the validation set. Stratification ensured that all label distribution across all tickers was preserved in training and validation set, . Two data loaders were constructed to handle batching, with each loader processing 10 batches for the training and validation sets, respectively.

Prior to the full-scale model training, a preliminary hyperparameter tuning phase was conducted on a smaller subset of the dataset to determine configurations that would most effectively support the model's learning dynamics. Based on this analysis, we selected the AdamW optimizer [26] and the OneCycleLR learning rate scheduler [27] for both training approaches. The AdamW optimizer was configured with betas = (0.98, 0.99) and weight_decay = 0.01. For the OneCycleLR scheduler, we set max_lr = 3.6e-5, pct_start = 0.07, anneal_strategy = "cos", and re-initialized the schedule at the start of each epoch. This re-initialization allows the learning rate to explore optimal ranges more effectively, enabling faster convergence when the model identifies a new steep optimization direction. Lastly the number of epochs for training the models is set to 11.

4.3 Dataset Predictive Relationship Reliability Test

Following the selection of optimal hyperparameters, we conducted a dataset predictive relationship reliability test to validate the relevance of our NLP dataset to the downstream tasks—namely, next-minute movement classification and next-minute return regression. This test involved training the model for the first 3 epochs using varying training set sizes, where each subset sample size was defined as a multiple *k* of 10000 samples. The objective of this approach was important as it would determine whether the dataset contains input signals that are meaningfully associated with the target labels, thereby indicating the feasibility of our proposed modelling direction.

Encouragingly, the results from this test revealed a positive trend starting at *k* = 6 onwards for both downstream tasks, as illustrated in Fig. 4.3.1. Specifically, the validation MSE loss consistently decreased in tandem with the training loss k=2 and validation accuracy increased at k=6 onwards when the number of training samples enlarged each time. This trend indicates the presence of recurring patterns in the dataset that the model was able to exploit for learning. Consequently, this provides preliminary evidence that our NLP dataset contains informative features relevant to predicting the target labels, supporting the viability of our approach.

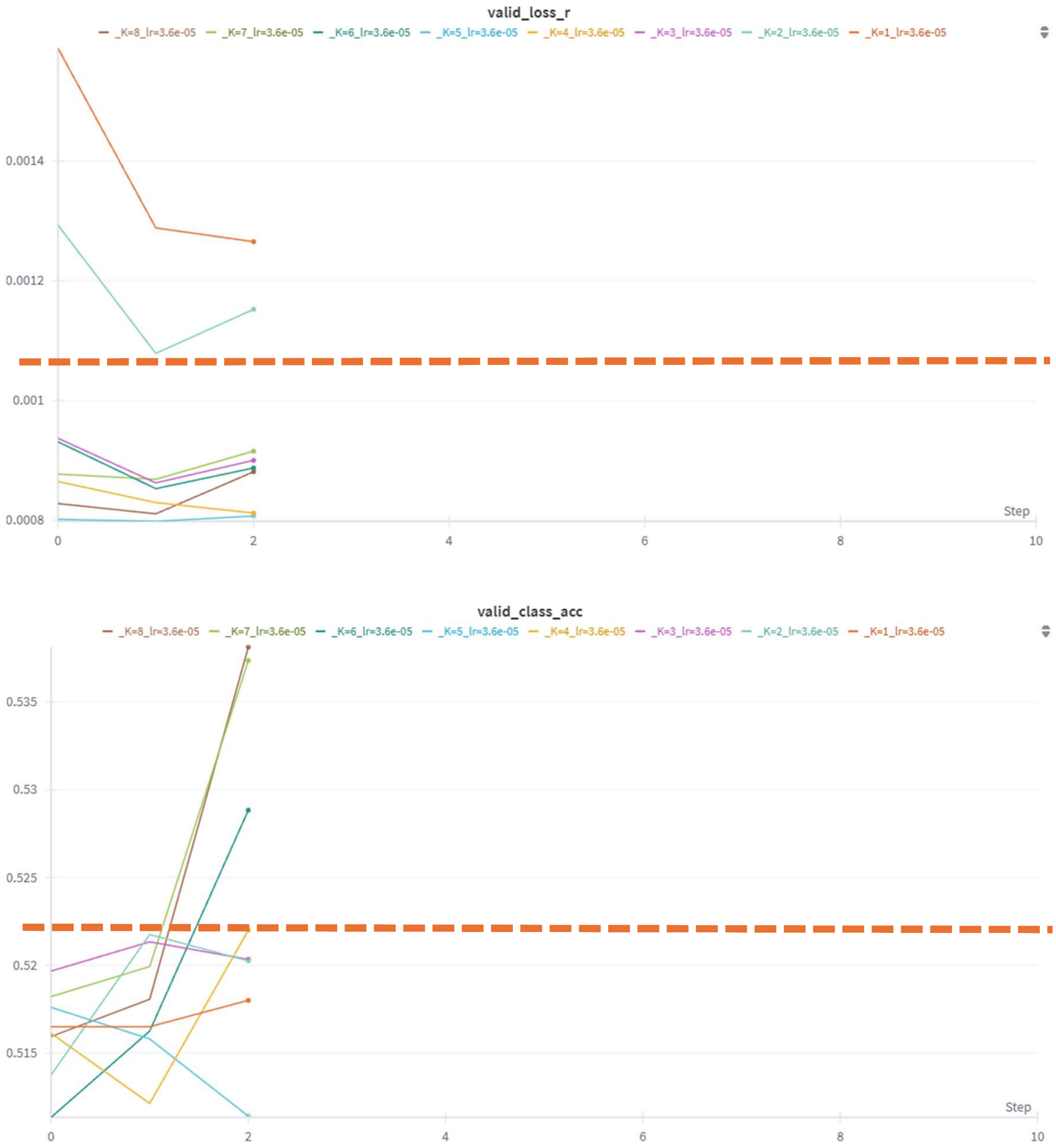


Fig. 4.3.1 Validation MSE & Accuracy from Relationship Test on K of 10000 samples

The outcomes from our dataset reliability validation test, along with a well-defined model architecture and optimized hyperparameters, establishes a strong foundation for model training. With these preparations in place, we proceeded to train both the feature-based and fine-tuning variants of the FusionStockBERT model. The following chapter presents a comprehensive evaluation of the model's performance across classification and regression tasks, and benchmarked against prior state-of-the-art approaches.

CHAPTER 5 EVALUATION AND FINDINGS

This chapter presents the evaluation of our proposed model across three phases: (1) overall accuracy performance, (2) class-specific performance analysis, and (3) comparative analysis with prior works. The evaluation focuses on both classification and regression aspects of the next-minute market prediction task. For classification task, we employ accuracy as a general performance indicator, and further assess model behaviour at the class level using recall, precision, and F1 score. These metrics provide a more granular understanding of how well the model distinguishes between different market movement directions. While for the regression task, we employ the Mean Squared Error (MSE) as a performance indicator to understand how precise the model is in giving the return expectation.

5.1 Overview of the Accuracy Performance

In overview of accuracy performance, we performed a baseline accuracy achievement evaluation to our models on the validation set. Our minimum accuracy baseline in our dataset for the next minute movement label is 51.6%. This minimum accuracy baseline is determined by the percentage of the dominant class resides in our dataset as shown in Fig. 5.1.1, it is to evaluate if a model had learned anything from the dataset and if it would perform better than a random guesser.

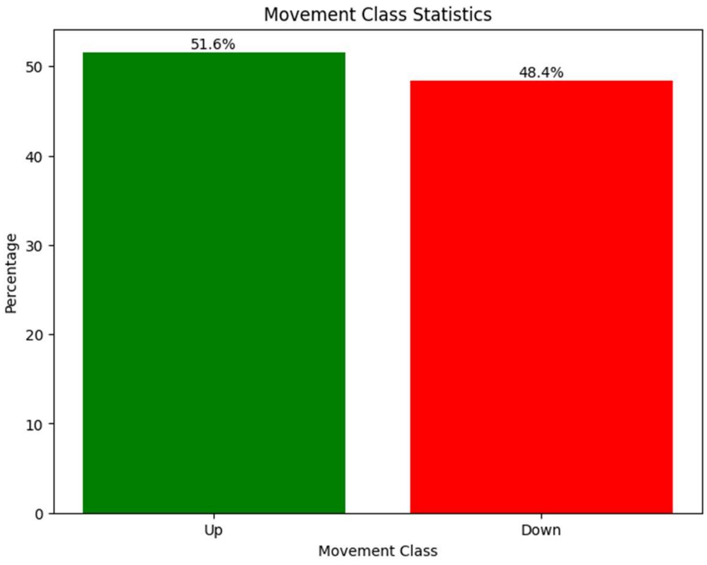


Fig. 5.1.1 Statistics: Number of Samples Per Label

The feature-based FusionStockBERT model resulted a validation accuracy of 52.32% in the next-minute movement classification task and a validation Mean Squared Error (MSE) of 0.97e-3 in the next-minute return regression task. As illustrated in Fig. 5.1.2 and Fig.5.1.3, this model's accuracy in movement task steadily increased over the training epochs but plateaued from epoch 6 onward, where both validation MSE loss and Classification Accuracy showed no further improvement. Despite achieving a relatively low MSE in the return prediction, the model's overall performance in direction classification was considered inadequate, as it only barely meets the minimum accuracy baseline of 51.6% Consequently, the model's inability to reliably predict the movement direction renders it less effective for informing trading decisions, even if return estimates appear numerically sound.

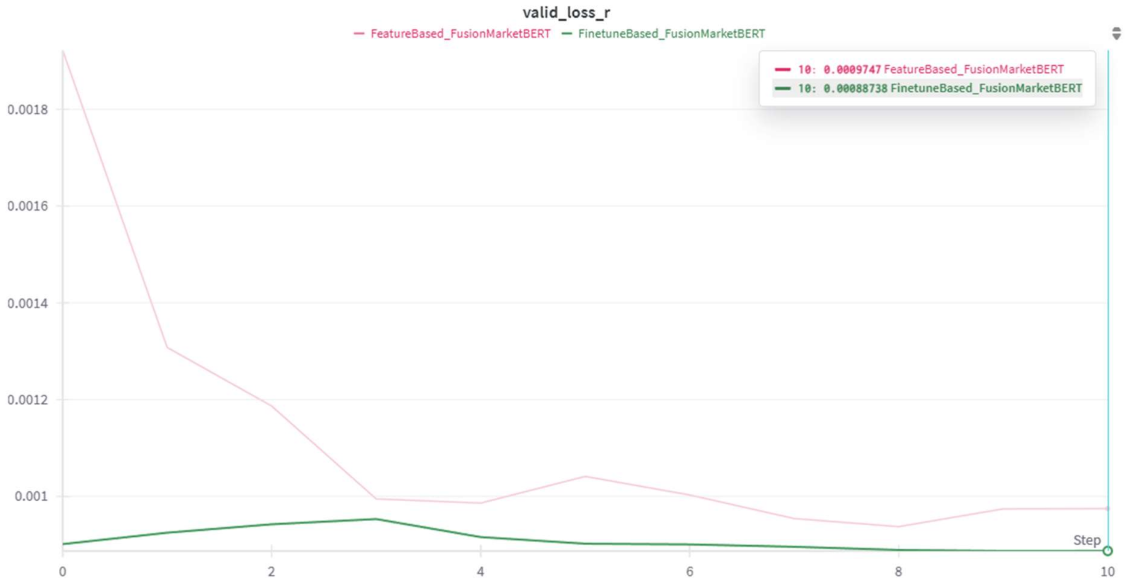


Fig. 5.1.2 Validation MSE of Feature-based & Fine-tune Based Model, over epoch.

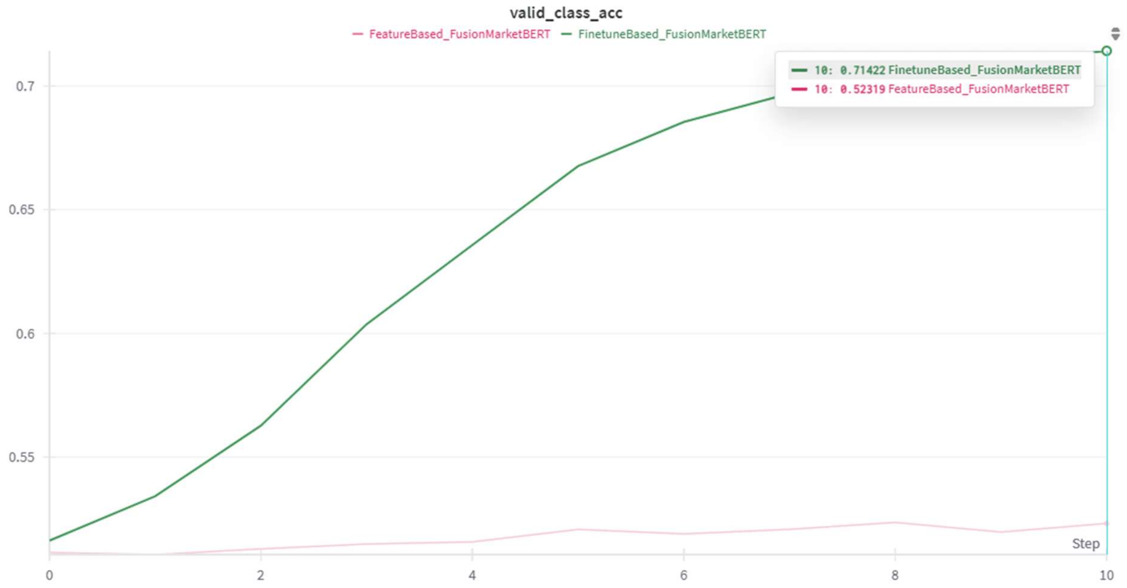


Fig. 5.1.3 Validation Accuracy of Feature-based & Fine-tune Based Model, over epoch.

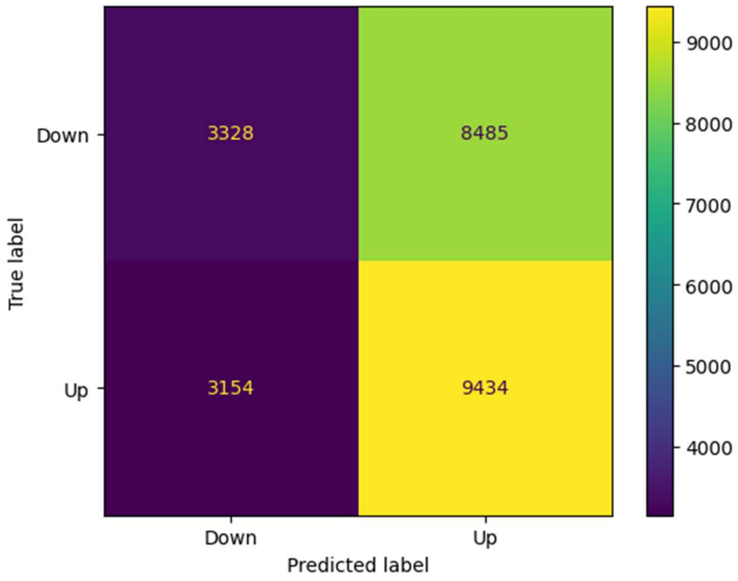
Meanwhile, the **fine-tuning-based FusionStockBERT model** achieved a significantly higher validation accuracy of 71.42% in the next-minute movement classification task and a lower validation Mean Squared Error (MSE) of 0.89e-3 in the next-minute return regression task compared to the feature-based model. This findings showed that the fine-tuning allowed the BERT model to more effectively capture the relationship between the market news and stock behaviour, leading to noticeably improved performance in both classification of stock movement and return regression tasks. Moreover, the model hugely surpassed the 51.6% accuracy baseline for the next-minute movement prediction, indicating its potential to offer more reliable and actionable insights for real-time trading decisions.

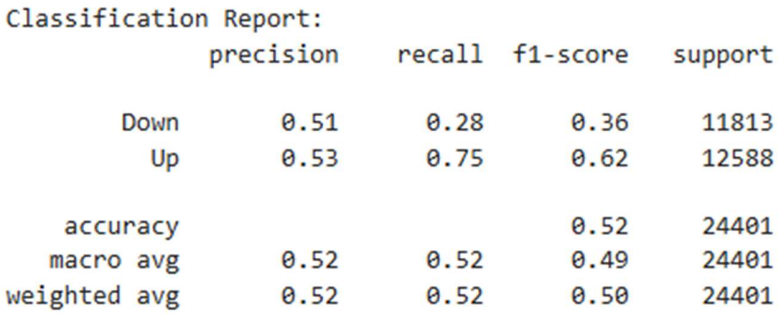
In our analysis, the poor performance of the feature-based model may be attributed to the mismatch between the natural language relationships learned from the analyst ratings task and the requirements of short-interval stock market movement prediction. Analyst ratings often reflect subjective opinions influenced by individual backgrounds and may not align well with fast-changing market conditions. Without fine-tuning, the model likely struggles to extract meaningful insights from the noisy, minute-level news speech, resulting in weak predictive performance for short-term market movements.

5.2 Class-specific Level Evaluation

In the class-specific level evaluation, we performed an analysis to the recall rate, precision rate, and F1 score of the models on validation set, such is to evaluate the performance of the models in predicting the next-minute movement class label.

Feature-based FusionStockBERT: In Fig 5.2.1, it showed the confusion matrix result and classification report of the feature-based FusionStockBERT on validation set, the Y=1 is placed with (Up) label and the Y=0 is placed with (Down) label . The feature-based FusionStockBERT model obtained a final recall rate of 0.75 and 0.28 for the positive and negative stock movement label respectively. The low recall rate in negative movement labels indicates that the model does not know how to distinguish negative movement label (Down) from positive movement label (Up), causing the negative movement label are being frequently predicted to positive movement class. This is also a reason that it results the precision of positive movement class to be very small. Among the predicted positive movement labels, there are only 53% of the samples being predicted correctly and other 47% are actual belonging to the negative movement labels.

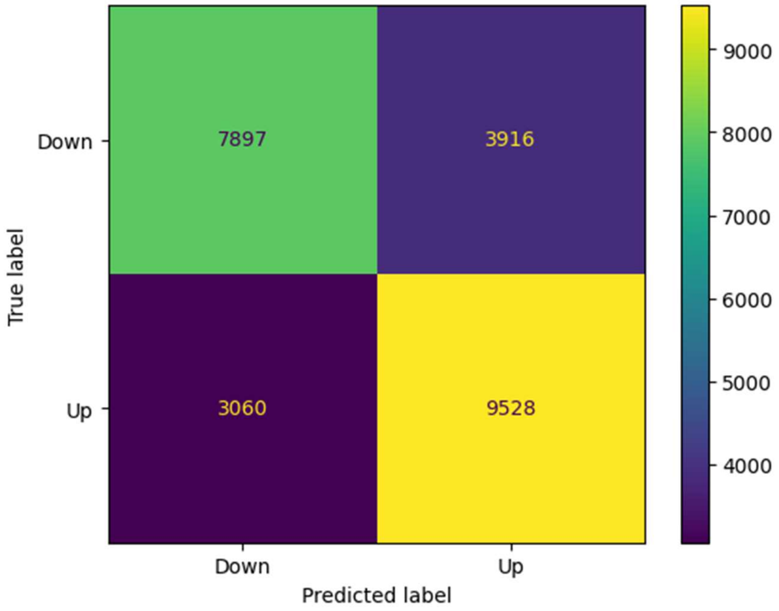


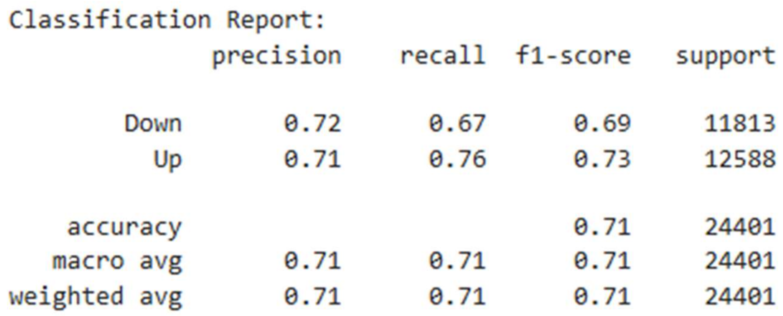


```
Classification Report:
              precision    recall  f1-score   support

        Down       0.51      0.28      0.36     11813
          Up       0.53      0.75      0.62     12588

    accuracy                           0.52     24401
   macro avg       0.52      0.52      0.49     24401
weighted avg       0.52      0.52      0.50     24401
```

Fig. 5.2.1 Confusion Matrix & Classification Report of Feature-based FusionStockBERT on Validation Set

Fine-tuning Based FusionStockBERT: As shown in Fig. 5.2.2, the confusion matrix result and classification report of the fine-tuning based FusionStockBERT on the validation set demonstrates significantly higher recall and precision rates compared to the feature-based model (Fig. 5.2.1). Specifically, it achieved a recall rate of 0.76 for the positive stock movement label and 0.67 for the negative label, along with a precision rate of 0.71 and 0.72 for the positive and negative labels respectively. The higher recall rate implies that the model is now having a lower false prediction for each of the classes, especially for the negative movement label, resulting the precision rate for each predicted classes increased.





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

        Down       0.72      0.67      0.69     11813
          Up       0.71      0.76      0.73     12588

    accuracy                           0.71     24401
   macro avg       0.71      0.71      0.71     24401
weighted avg       0.71      0.71      0.71     24401
```

Fig. 5.2.2 Confusion Matrix & Classification Report of Fine-tuning based FusionStockBERT on Validation Set

After fine-tuned the BERT in our FusionStockBERT model, among the predicted positive movement labels, there are now only having 29% of false positive samples which is a huge 18% improvement from the 47% in feature-based model. And the recall rate in the negative movement ("Down")

These improvements are largely attributed to the fine-tuning process, which enables the pre-trained BERT model to adapt its internal representations to the specific requirements of our task. By allowing the model to update its parameters, fine-tuning enhances its ability to filter noise and extract relevant features from minute-level news speech, resulting in more accurate predictions for each stock movements.

5.3 Performance Comparison to Prior Works

In this 5.3 subsection, we compare the performance of our model against previous studies. In the performance comparison, we added an additional metrics – Matthews Correlation Cofficient (MCC), that was used by prior studies, to have a more detailed comparison. As illustrated in Tab. 5.3.1 and 5.3.2, the fine-tuned FusionStockBERT demonstrates a substantial advancement and generalization in the stock market movement prediction task. Despite the inherent noise in minute-level news speech, our model sustains robust predictive performance and surpasses the accuracy of prior works that employed CNN-based approaches.

Table 5.3.1 Model Performance Comparison on Training Set in Stock Market Movement Task ('-' means NA)

Works -	Xiao Ding et al. [10]				(This paper - FusionStockBERT)	
Model -	WB-NN	EB-NN	WB-CNN	EB-CNN	Feature-based	Finetune-based
Accuracy	60.25%	62.84%	61.73%	65.08%	52.88%	**80.53%**
MCC	0.1958	0.3472	0.2147	0.4357	0.0959	**0.7106**
MSE Return	-	-	-	-	1.04e-3	**0.96e-3**

Table 5.3.2 Model Performance Comparison on Validation Set in Stock Market Movement Task ('-' means NA)

Works -	Xiao Ding et al. [10]				(This paper - FusionStockBERT)	
Model -	WB-NN	EB-NN	WB-CNN	EB-CNN	Feature-based	Finetune-based
Accuracy	-	-	-	64.21%	52.32%	**71.42%**
MCC	-	-	-	0.40	0.0352	**0.4274**
MSE Return	-	-	-	-	0.97e-3	**0.89e-3**

Our fine-tuned FusionStockBERT model achieved an 80.53% of accuracy and 0.7106 of MCC on the training set, significantly outperforming the best previous work -- EB-CNN model that reported an accuracy result of 65.08% and MCC of 0.4357. On the validation set, FusionStockBERT retained strong generalization capability, achieving 71.24% accuracy and 0.4274 of MCC compared to the 64.12% accuracy and 0.40 of MCC reported in EB-CNN. This notable improvement highlights the advantage of leveraging contextualized language representations through BERT and integrating multi-modal fusion techniques for fine-grained temporal prediction in the stock market domain.

CHAPTER 6 CONCLUSION & RECOMMENDATIONS

In this study, we demonstrated that a transformer-based architecture—fine-tuned FinBERT—can be highly effective for stock market movement prediction when combined with quantitative trading features through a multi-modal fusion approach. Our proposed model, FusionStockBERT, excelled at capturing useful time-sensitive signals that present in the noisy minute-level financial news. Crucially, fine-tuning FinBERT on next-minute up/down movement labels—rather than treating it as a frozen feature extractor—yielded a 19.1% uplift in classification accuracy and reduced return-regression error by 0.9×10^{-4} MSE on the validation set compared to a feature-based variant, underscoring the value of task-specific adaptation in financial NLP.

By combining contextualized language embeddings with structured market data, FusionStockBERT was able to generalize effectively to unseen data, confirming the reliability of our minute-level news and trading dataset. The model's success highlights two key insights: (1), transformer-based representations offer substantial advantages over traditional NLP and non-attention deep-learning methods; and (2), multi-modal fusion of news and numeric inputs provides a robust framework for real-time, high-frequency forecasting.

For future work, we recommend enhancing temporal modeling—perhaps by stacking additional recurrent layers atop the model outputs—to better capture dependencies beyond single-minute intervals. Additionally, applying this framework to other asset classes (e.g., commodities, cryptocurrencies) and extending prediction horizons to hourly or daily intervals would test its flexibility and scalability.

REFERENCES

[1] A. Petropoulos and V. Siakoulis, “Can central bank speeches predict financial market turbulence? Evidence from an adaptive NLP sentiment index analysis using XGBoost machine learning technique,” *Central Bank Review*, vol. 21, no. 4, pp. 141–153, Dec. 2021, doi: 10.1016/j.cbrev.2021.12.002.

[2] Y. Yang, M. C. S. Uy, and A. Huang, “FinBERT: a pretrained language model for financial communications,” *arXiv.org*, Jun. 15, 2020. https://arxiv.org/abs/2006.08097

[3] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, “Characterizing concept drift,” *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, Apr. 2016, doi: 10.1007/s10618-015-0448-4.

[4] Y. Zhao and G. Yang, “Deep Learning-based Integrated Framework for stock price movement prediction,” *Applied Soft Computing*, vol. 133, p. 109921, Dec. 2022, doi: 10.1016/j.asoc.2022.109921.

[5] “Extrapolation Using Regression - Challenges and Solutions - Google Search.” https://www.google.com/search?q=Extrapolation+Using+Regression+-+Challenges+and+Solutions&oq=Extrapolation+Using+Regression+-+Challenges+and+Solutions&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQRRg80gEIMTA5MmowajeoAgCwAgA&sourceid=chrome&ie=UTF-8

[6] T. Hengl, M. Nussbaum, M. N. Wright, G. B. M. Heuvelink, and B. Gräler, “Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables,” *PeerJ*, vol. 6, p. e5518, Aug. 2018, doi: 10.7717/peerj.5518.

[7] K. Puh and M. B. Babac, “Predicting stock market using natural language processing,” *American Journal of Business*, vol. 38, no. 2, pp. 41–61, Apr. 2023, doi: 10.1108/ajb-08-2022-0124.

[8] A. Moghar and M. Hamiche, “Stock market prediction using LSTM Recurrent Neural network,” *Procedia Computer Science*, vol. 170, pp. 1168–1173, Jan. 2020, doi: 10.1016/j.procs.2020.03.049.

[9] “Predicting the effects of news sentiments on the stock market,” *IEEE Conference Publication | IEEE Xplore*, Dec. 01, 2018. https://ieeexplore.ieee.org/document/8621884

[10] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” *International Conference on Artificial Intelligence*, pp. 2327–2333, Jul. 2015, [Online]. Available: http://ir.hit.edu.cn/~xding/docs/Deep%20Learning%20for%20Event-Driven%20Stock%20Prediction%20slides.pdf

[11] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv (Cornell University)*, Jan. 2015, [Online]. Available: https://arxiv.org/pdf/1409.0473

[12] A. Vaswani *et al.*, “Attention is All you Need,” *arXiv (Cornell University)*, vol. 30, pp. 5998–6008, Jun. 2017, [Online]. Available: https://arxiv.org/pdf/1706.03762v5

[13] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” *International Conference on Machine Learning*, pp. 1243–1252, May 2017, [Online]. Available: http://proceedings.mlr.press/v70/gehring17a/gehring17a.pdf

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing Human-Level performance on ImageNet classification,” Dec. 2015. doi: 10.1109/iccv.2015.123.

[15] “Open LLM Leaderboard - a Hugging Face Space by open-llm-leaderboard-old.” https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard

[16] E. Zhu and J. Yen, “BERTopic-Driven Stock market Predictions: Unraveling sentiment insights,” *arXiv.org*, Apr. 02, 2024. https://arxiv.org/abs/2404.02053

[17] A. Atkins, M. Niranjan, and E. Gerding, “Financial news predicts stock market volatility better than close price,” *The Journal of Finance and Data Science*, vol. 4, no. 2, pp. 120–137, Feb. 2018, doi: 10.1016/j.jfds.2018.02.002.

[18] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Using Structured Events to Predict Stock Price Movement: An Empirical Investigation,” Jan. 2014. doi: 10.3115/v1/d14-1148.

[19] Openai, “GitHub - openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision,” *GitHub*. https://github.com/openai/whisper

[20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via Large-Scale Weak Supervision,” *arXiv.org*, Dec. 06, 2022. https://arxiv.org/abs/2212.04356

[21] “OpenAPI Introduction | moomoo API Doc v9.2.” https://openapi.moomoo.com/moomoo-api-doc/en/intro/intro.html

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv.org*, Oct. 11, 2018. https://arxiv.org/abs/1810.04805

[23] A. H. Huang, H. Wang, and Y. Yang, “FinBERT: A Large Language Model for Extracting Information from Financial Text*,” *Contemporary Accounting Research*, vol. 40, no. 2, pp. 806–841, Sep. 2022, doi: 10.1111/1911-3846.12832.

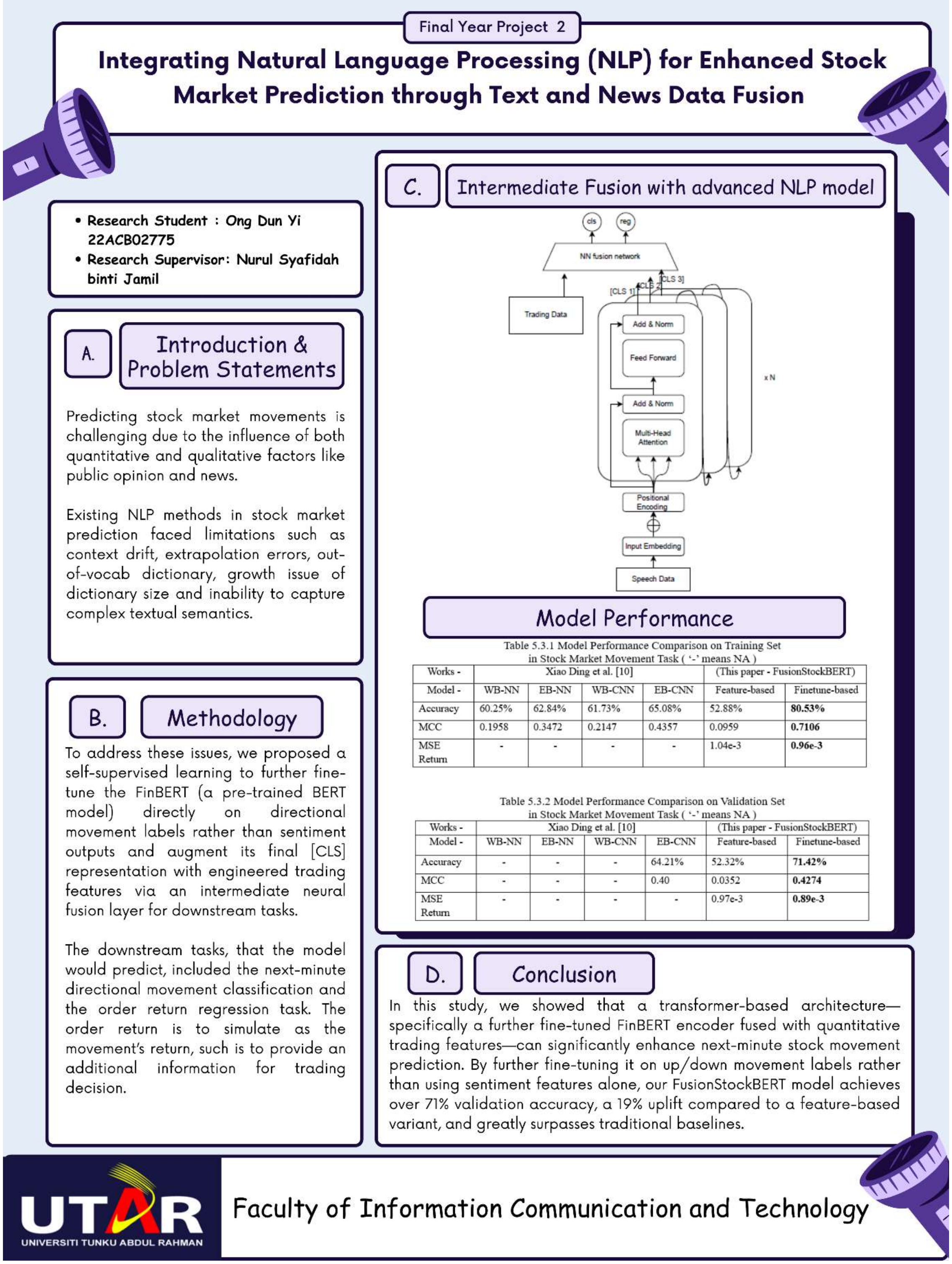
[24] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv.org*, May 05, 2015. https://arxiv.org/abs/1505.00853

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv.org*, Dec. 10, 2015. https://arxiv.org/abs/1512.03385

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv.org*, Nov. 14, 2017. https://arxiv.org/abs/1711.05101

[27] L. N. Smith and N. Topin, "Super-Convergence: very fast training of neural networks using large learning rates," *arXiv.org*, Aug. 23, 2017. https://arxiv.org/abs/1708.07120

POSTER



Final Year Project 2

Integrating Natural Language Processing (NLP) for Enhanced Stock Market Prediction through Text and News Data Fusion

- Research Student : Ong Dun Yi 22ACB02775
- Research Supervisor: Nurul Syafidah binti Jamil

A. Introduction & Problem Statements

Predicting stock market movements is challenging due to the influence of both quantitative and qualitative factors like public opinion and news.

Existing NLP methods in stock market prediction faced limitations such as context drift, extrapolation errors, out-of-vocab dictionary, growth issue of dictionary size and inability to capture complex textual semantics.

B. Methodology

To address these issues, we proposed a self-supervised learning to further fine-tune the FinBERT (a pre-trained BERT model) directly on directional movement labels rather than sentiment outputs and augment its final [CLS] representation with engineered trading features via an intermediate neural fusion layer for downstream tasks.

The downstream tasks, that the model would predict, included the next-minute directional movement classification and the order return regression task. The order return is to simulate as the movement's return, such is to provide an additional information for trading decision.

C. Intermediate Fusion with advanced NLP model

Model Performance

Table 5.3.1 Model Performance Comparison on Training Set in Stock Market Movement Task ('-' means NA)

Works -	Xiao Ding et al. [10]				(This paper - FusionStockBERT)	
Model -	WB-NN	EB-NN	WB-CNN	EB-CNN	Feature-based	Finetune-based
Accuracy	60.25%	62.84%	61.73%	65.08%	52.88%	**80.53%**
MCC	0.1958	0.3472	0.2147	0.4357	0.0959	**0.7106**
MSE Return	-	-	-	-	1.04e-3	**0.96e-3**

Table 5.3.2 Model Performance Comparison on Validation Set in Stock Market Movement Task ('-' means NA)

Works -	Xiao Ding et al. [10]				(This paper - FusionStockBERT)	
Model -	WB-NN	EB-NN	WB-CNN	EB-CNN	Feature-based	Finetune-based
Accuracy	-	-	-	64.21%	52.32%	**71.42%**
MCC	-	-	-	0.40	0.0352	**0.4274**
MSE Return	-	-	-	-	0.97e-3	**0.89e-3**

D. Conclusion

In this study, we showed that a transformer-based architecture—specifically a further fine-tuned FinBERT encoder fused with quantitative trading features—can significantly enhance next-minute stock movement prediction. By further fine-tuning it on up/down movement labels rather than using sentiment features alone, our FusionStockBERT model achieves over 71% validation accuracy, a 19% uplift compared to a feature-based variant, and greatly surpasses traditional baselines.

UTAR UNIVERSITI TUNKU ABDUL RAHMAN

Faculty of Information Communication and Technology