Forex Trading Signal Extraction with Deep Learning Models

LING QI

M. Phil

THE UNIVERSITY OF SYDNEY

Supervisor: Dr. Josiah Poon
Associate Supervisor: Dr. Matloob Khushi

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

Faculty of Engineering
The University of Sydney
Australia

30 June 2023

Abstract

The emergence of artificial intelligence (AI) technology has made deep learning models popular in the prediction of financial trading. An automated system that could correctly predict prices or trends will help investors generate considerable profit with minimum risk. Leading banks and trading firms, such as Westpac, CBA, Macquarie Bank, and Bloomberg, are investing money and human resources to leverage the advancements of such transformative technology. Researchers have also studied the exchange rate market by adopting this cutting-edge technology. This thesis focuses on creating advanced deep learning models to predict the forex market as correctly as possible and to discover trading strategies powered by AI.

In this thesis, three distinct deep learning models were proposed: an event-driven LSTM model, an Attention-based VGG16 (MHATTN-VGG16), and a pre-trained model (TradingBERT). The introduced three models can facilitate better-informed investment strategies via correct signal extraction and price forecast in forex trading. These models generate trading signals based on historical price data and derived features. The outputs of these models show promising improvements in price and trend prediction accuracy compared to baseline models. The enhanced prediction results offer traders valuable insights, enabling them to make trading decisions based on the most predicted signals within the prediction horizon.

The first model proposed is a Long Short-Term Memory (LSTM) focusing on the prediction of the retracement points, which indicate a change in trading trend direction. The accurate prediction of these retracement points can provide an optimal entry point and lead to a successful trading strategy. The training data was selected based on a sequence of events derived from the Elliott Wave. The standard LSTM, supplemented by unique noise reduction in the training data, shows significant improvement in predicting retracement points compared to the baseline models, namely GRU and RNN. Furthermore, this study conducts experiments to determine the optimal number of timesteps needed to identify an underlying price trend. The results demonstrate significant potential for building a robotic trading platform.

The second model proposed is a Multi-Head Attention VGG model, designed to forecast the maximum and minimum price movements in forex trading chart images on a scale. Accurate prediction of these price movement scales can support investment decisions. The method proposed in this paper, MHATTN-VGG16, uses VGG16 as a base, incorporating a multi-head attention mechanism to capture global and local information in the image. It also employs two-dimensional positional encoding to guide the model on where and what to focus on within the image. The experimental results demonstrate that the MHATTN-VGG16 architecture has the ability to effectively classify financial chart images.

The third novel system proposed employs a pre-trained model based on the BERT architecture. This model is built upon four essential design principles: transforming trading price data points into normalized embeddings; constructing masked sequences for self-supervised learning; developing seven self-learning tasks for sequential data; and employing the transformer's encoder mechanism for training. The experimental results indicate that the features generated from the pre-trained model are strong contenders for TA indicators, demonstrating the model's ability to extract meaningful signals from financial price data. This study lays the groundwork for using pre-trained models in financial trading and introduces a method to convert continuous price data into a set of categorized elements, thus enabling the embedding to utilize the success of the BERT architecture.

My research in this thesis particularly emphasizes deep learning models such as LSTM, BiLSTM, GRU, CNN, Transformer, and supervised and unsupervised learning methods. When thoughtfully applied and combined, the studies reveal how specific deep learning techniques can successfully augment the decision-making process in forex trading. This research project provides valuable contributions to the scientific community in its findings and innovative approach to utilizing deep learning in algorithmic trading.

Acknowledgements

I would first like to thank my supervisors, Dr. Matloob Khushi and Dr. Josiah Poon, for their supervision and guidance. I am grateful for their mentorship; I would not be able to complete this thesis without their suggestion, advice, and support. During this journey, I have learned many things from them, for example, what is a novel system and how to develop a novel system; throughout this journey, I have been fortunate enough to benefit from their wisdom, patience, and dedication, which have helped me tremendously.

Also, I would like to thank Dr. Saqib Ayyubi. Dr. Ayyubi was my capstone project tutor in the Master of Data Science; his instruction has not only enhanced my knowledge but also inspired me to strive for excellence in my work.

Statement of originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

[redacted] June 12, 2023

Student: Ling Qi Date

Authorship attribution statement

Chapter 5 - "Event-Driven LSTM For Forex Price Prediction" of this thesis is published as [1]. I designed the study, built the models, analysed the results, and wrote the drafts of the MS.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

[redacted] June 12, 2023

Student: Ling Qi Date

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

[redacted]

Supervisor: Dr. Josiah Poon Date

[redacted] 13 June 2023

Supervisor: Dr. Matloob Khushi Date

Publications

The investigations conducted in this thesis have resulted in the creation of the following publications:

- Ling Qi, Matloob Khushi, and Josiah Poon. Event-driven LSTM for forex price prediction. Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE): 16-18 December 2020; Virtual, pages 1-6.
- Ling Qi, Matloob Khushi, and Josiah Poon. TradingBERT: A Pre-trained Model for Financial Trading Prediction. Currently under review.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This Chapter aims to provide an expansive overview of this thesis. The opening section uncovers the motivations behind this research, offering a brief yet comprehensive introduction to financial markets and a critical examination of the latest applications of Artificial Intelligence (AI) in financial trading. It discusses the predictability of financial trading time series data, setting the stage for scientific inquiries to follow.

Following the motivational groundwork, the subsequent section introduces the core research questions of the thesis. The structure of this thesis is then outlined, followed by an analysis of the contributions, which will be discussed at the end of this chapter.

1.1 Motivations

Forex trading, also known as foreign exchange or currency trading, is a global market where participants buy, sell, and exchange currencies. With a daily trading volume of over $6 trillion, the forex market is the world's largest and most liquid financial market. Investors, banks, corporations, and governments participate in this decentralized market, which operates 24 hours a day, five days a week.

Forex market, due to its high liquidity, has a massive amount of historical data available for analysis, making it an ideal platform for implementing Machine Learning techniques. The OHLC (Open, High, Low, Close) prices, which provide comprehensive information about price fluctuations during a specific time frame. The market mainly comprises "Cross Currency" pairs such as AUD/USD and EUR/GBP, which create diverse market structures based on currency combinations through trades taking place in ratios determined by the pairing.

On the other hand, two primary methods are used to analyse and forecast market trends: fundamental analysis[2] and technical analysis[3]. Fundamental analysis focuses on evaluating a currency's intrinsic value by considering economic indicators such as interest rates, inflation, gross domestic product (GDP), political factors, and global events. This type of analysis helps traders to understand the overall health of a country's economy and the potential impact on currency prices.

Technical analysis[4] is a methodology that studies historical price data and market trends to predict future price movements. Technical analysts use tools like chart patterns, trend lines, and technical indicators to identify potential trade opportunities. Simple moving average (SMA)[5], exponential moving average (EMA)[6], and relative strength index (RSI)[7] are commonly used technical indicators.

Although many followers of the Random Walk[8, 9] and Efficient-market Hypothesis (EMH)[10–12] have expressed doubt about the ability to predict foreign exchange prices, traders who use Technical Analysis[4] have achieved considerable success in making profitable trades. The implication is that there are identifiable patterns in historical data that reveal market trends, and the computational power of Machine Learning techniques can be harnessed to make profitable predictions based on these patterns.

In recent years, there has been a growing preference for using neural networks to forecast financial trading markets due to their ability to detect nonlinear data relationships. Many researchers have explored a range of deep learning models, including DMLP, RNN, LSTM, CNN, RBM, DBN, AE, and DRL, to predict price movements and trends in various financial trades[13], including forex, stocks, and indices. These models have shown notable success.

Long-Short Term Memory (LSTM) is an outstanding model in the deep learning world to predict high-frequency trading data due to its capability to handle long-term corrections in sequential data while also addressing the vanishing gradient problem. LSTM enables the effective handling of long-term correlations via its memory cells and gates. Successful applications of LSTM in financial trading include its integration with Reinforcement Learning for maximizing trading profit[14] and its use in feature fusion LSTM-CNN[15] models for predicting stock prices by extracting features from stock price chart images.

Convolution Neural Network (CNN), as an important class of neural networks, has had great success in learning image representations. CNN consists of an input layer, multiple hidden layers, and an output layer. Its hidden layers consist of a series of convolutional layers that convolve the input and pass its result to the next layer; CNN keeps pooling, and convolution helps to extract useful features for each layer and to finally obtain high-quality features and provide a capable prediction result. Studies employing Convolution Neural Networks (CNN) for analysing financial trading charts[16–18] have demonstrated commendable predictive accuracy for trading prices or trends.

Other achievements in this area include successfully implementing reinforcement learning algorithms for optimizing trading decisions[14], developing advanced deep learning models that outperform traditional methods[1, 15, 19–21], and integrating natural language processing techniques to analyse news sentiment and predict market reactions[22–24]. However, it is essential to acknowledge that while machine learning models show promise in forex trading, they are not infallible and should be used with human expertise and sound risk management strategies.

As the primary resource for feature generation in most trading literature over the past few decades, TA indicators have some limitations. These include the possibility of future information leakage, challenges in reducing or selecting features due to the abundance of technical indicators, and heightened model complexity. One way to overcome these issues is by utilizing a pre-trained model that offers reliable attributes while preventing information leakage. This approach can optimize computational resources and decrease time spent on the task.

This thesis presents three studies that apply deep learning models to forex trading, including LSTM, BERT, and VGG, with Multi-head attention. The aim of this study is to create effective deep learning models that can forecast price trends and generate trade signals by analysing historical prices. Accurate predictions can support a more informed trading strategy and lead to profitable trading decisions.

1.2 Research questions

This thesis delves into the integration of deep learning methodologies for predicting forex price trends and signals, as well as investigating the underpinning techniques that drive these predictions. In order to thoroughly comprehend the impact and feasibility of deploying deep learning in the context of forex market forecasting, this research seeks to answer the following pivotal questions:

- How do different feature selection and feature generation strategies influence the accuracy and efficiency of forex price and trend predictions?
- Can we devise a viable approach to convert numerical features into categorical counterparts to facilitate embedding and harnessing the powerful BERT technology in forex trading prediction tasks?
- Is it feasible to construct a pre-trained model specifically designed for the nuanced requirements of forex trading time series data?
- How does incorporating a pre-trained model impact the performance and reliability of downstream prediction tasks in forex trading?
- In the realm of forex trading, how do deep learning architectures like LSTM, BiLSTM, GRU, Transformer, and the Convolutional Neural Network (CNN)-based VGG model compare in terms of trading signal extraction and price trend prediction?

By answering these questions, this research intends to shed light on the potential of deep learning techniques in the forex trading arena and establish innovative methodologies for improving prediction accuracy.

1.3 Proposed methods

This thesis navigates the intricate landscape of trading data analysis, primarily deploying deep learning models. Deep learning, an advanced subset of machine learning, has garnered significant attention due to its remarkable adaptability and efficacy across various sectors. From Natural Language Processing (NLP) to Image Processing, deep learning continues to foster innovative developments.

As a substantial part of this research endeavor, three impactful studies have been conducted, each contributing to the ever-evolving discourse in this field. The first published paper, titled 'Event-Driven LSTM For Forex Price Prediction,' leverages the LSTM architecture - a potent deep learning model adept at capturing temporal dependencies - to predict future trends in forex prices. This study marks a pioneering step in integrating event-driven data with LSTM for more precise financial forecasting.

The second research study, a potential future paper, is titled 'Multi-head Attention VGG For forex Trading Image Classification.' It explores the application of the Convolutional Neural Network (CNN)-based VGG model, enhanced with a multi-head attention mechanism, for forex trading image classification. By treating the forex trading data as images, this study provides a fresh perspective on forex trading prediction tasks.

The third paper, 'TradingBERT: A Pre-trained Model for Financial Trading Prediction,' is under review. This study seeks to construct a pre-trained model rooted in the BERT architecture, specifically designed for feature extraction in financial trading data. By focusing on the four primary price points: Open, High, Low, and Close, this model negates the need for features typically derived from Technical Analysis. This paper is distinctive as it makes a groundbreaking attempt to propose a pre-trained model tailored to the nuances of financial trading.

Collectively, these studies shed light on how deep learning models can be effectively utilized to predict forex trading trends and prices and subsequently generate reliable trading signals. These findings pave the way for innovative strategies that could potentially reshape the landscape of financial forecasting methodologies.

1.4 Structure

The structure of this thesis is divided into nine chapters. A short description of the chapters is listed as follows:

Chapter 1 - Introduction: This chapter gives a broad perspective on this thesis. It describes the background, explains what the topic is about, and outlines the objectives of the study.

Chapter 2 - Conceptual Framework: This chapter aims to provide an understanding of the key concepts of the financial market and the methods employed to develop a trading strategy. Furthermore, I will dive into the utilization of deep learning and the reasons behind the growing interest in this topic in the financial trading world.

Chapter 3 – Literature Review: In this literature review, a detailed study of the existing literature is presented, delivering deeply into the application of deep learning techniques in predicting forex trading patterns, with a specific focus on LSTM, CNN, Transformer, and pre-trained models. This discussion focuses on influential deep learning models such as LSTM, BiLSTM, GRU, Transformer, and transfer learning, and the use of supervised and unsupervised learning methods. The exploration of these cutting-edge modeling techniques will cover key theories, significant discoveries, and current debates, as well as identify potential research gaps in the field of market forecasting.

Chapter 4 - Methodology: This chapter thoroughly explains the empirical approach that was taken, including the methods utilized, data collection procedures, and evaluation protocols. It offers a brief summary of the unique structures of deep learning models, specifically focusing on LSTM, CNN, and Transformer models. It explains the algorithms that drive these models, highlighting their individual advantages and limitations, particularly in predicting forex trading patterns. Additionally, this chapter outlines the experimental design, data collection, and analytical methodologies employed, while acknowledging any limitations of the study.

Chapter 5 - Event-Driven LSTM For Forex Price Prediction: This chapter presents the first publication[1], which is an innovative application of LSTM for forex trading. It includes this specific study's introduction, methods, results, discussion, and conclusion.

Chapter 6 - Multi-head Attention VGG For Forex Trading Image Classification: This chapter presents a potential future paper, proposing an innovative application of image classification techniques within the forex trading domain, and again following the same structure as Chapter 5.

Chapter 7 - TradingBERT: A Pre-trained Model for Financial Trading Prediction: This chapter introduces a paper currently under review, which describes the creation and evaluation of a pre-trained model that assists in forecasting trading outcomes, structured similarly to Chapters 5 and 6.

Chapter 8 - Discussion: This chapter consolidates the findings from the previous chapters. It interprets the results, summarizes the key findings, connects the outcomes, discusses the implications, and sets the stage for the final remarks.

Chapter 9 - Conclusion and Future Work: As a final summation, in this chapter, the research question is thoroughly examined by analysing the implications and inferences derived from the model outcomes. Furthermore, a forward-looking perspective is presented, which suggests potential improvements and future directions that could enhance the value of the findings encapsulated within this thesis.

1.5 Contributions

This thesis primarily aims to design diverse deep learning architectures capable of predicting price trends and extracting forex trading signals, as demonstrated through the models presented in Chapters 5, 6, and 7.

The chief contribution of Chapter 5 lies in the LSTM architecture incorporating a unique event-driven approach that effectively removes noise interference within trading time series data. By employing BiLSTM and GRU models, this approach yielded significantly superior prediction results compared to the standard RNN model. The empirical results attest to the immense potential for the creation of an efficient robotic trading platform.

Chapter 6 introduces a groundbreaking architecture that synergizes multi-head attention with VGG16 for the classification of financial chart images. Thanks to the attention mechanism, the MHATT-VGG16 model can more effectively discern 'where' and 'what' requires paramount focus. This paper also devises a two-dimensional positional encoding mechanism using sine and cosine functions, enabling the model to discern the importance of relative locations. Compared to the baseline models (VGG16 alone, ResNet50, and Xception), the proposed MHATT-VGG16 integrated with two-dimensional positional encoding demonstrates superior performance.

Chapter 7 unveils the TradingBERT model, which applies the power and structure of the BERT method to the realm of financial trading. Acting as a foundational pre-trained model in the financial trading sector for future investigations, TradingBERT introduces a method to transform trading price data into categorical elements, which can generate embeddings for the transformer. Experimental results from the model's downstream tasks validate its efficiency in producing valuable features. While most studies in the trading domain employ technical analysis indicators as a feature source, these can occasionally use future information, leading to instability during testing and backtesting. The results from my tests for downstream tasks align with these findings, indicating that TradingBERT is a dependable feature source for trading prediction tasks.

CHAPTER 2

Conceptual Framework

This chapter provides a comprehensive background of financial markets, specifically focusing on the ever-evolving world of forex trading. I will delve into the various methods utilized for analysing the forex market, provide an in-depth analysis of the current utilization of AI technology in forex trading, and discuss their effectiveness in predicting forex trends. Additionally, I will explore the recent achievements and successes in forex trading forecasts, emphasizing the advancements that have contributed to better-informed trading decisions.

2.1 Financial markets

The financial markets form the complex framework of the global economy, providing a platform for participants to trade various financial instruments and manage risk. Financial markets are significant as they enable the exchange of capital, credit, and risk management, and play a crucial role in allocating resources and determining the prices of financial assets. I can categorize financial markets into four main types, which are equity markets, debt markets, commodities markets, and foreign exchange markets.

Equity markets, which are also known as stock markets, provide a platform for investors to trade shares of publicly traded companies. These markets are instrumental in helping businesses raise capital for various growth initiatives such as expansion, research, and development. The value of a company is determined by the law of supply and demand in stock markets, which is essential for investors to make informed investment decisions. The process of price discovery and liquidity is an integral part of this process.

Debt or bond markets are where investors trade fixed-income securities such as government, corporate, and municipal bonds. These markets play a vital role in providing financing for governments, corporations, and other entities, enabling them to borrow money to fund various projects or cover budget deficits. In return, bond issuers pay interest to the investors, known as the bondholders, making debt markets a key source of income for long-term investors.

Commodities markets are where participants trade raw materials and primary products, such as agricultural goods, metals, and energy resources. Trading can take place in either spot markets, where delivery happens immediately, or futures markets, where contracts are made for future delivery at a set price. These markets are crucial for the management of supply chain risks, hedging against price fluctuations, and speculating on future price movements for both producers and consumers of commodities.

Foreign exchange (forex) markets facilitate the exchange of one currency for another, enabling international trade, investments, and risk management. As the largest and most liquid of all financial markets, forex trading sees a daily volume exceeding \$6 trillion. The forex market is active 24 hours a day, five days a week, and facilitates connections between major financial centers across the world. This market is essential for establishing exchange rates for different currencies, which in turn plays a critical role in cross-border transactions and evaluating the economic strength of different countries.

The global economy heavily relies on financial markets, which offer investment opportunities for individuals and institutions while facilitating efficient capital allocation and international trade for economic growth. In recent times, technological advancements such as high-frequency trading, blockchain, and AI have brought about increased efficiency, accessibility, and innovation in the financial market landscape. As the world evolves, financial markets will continue to adapt to meet new challenges and opportunities, remaining a crucial component of the global economic infrastructure.

2.2 Algorithm trading in forex

Algorithmic trading in the forex market has gained significant traction over the past few years, as traders increasingly rely on computer algorithms to predict price movements and identify trading opportunities. In this context, algorithmic trading strategies are specifically designed to analyse vast amounts of historical and real-time data to forecast currency price trends and generate actionable trade signals.

forex price and trend prediction algorithms can be broadly categorized into two main approaches: technical analysis-based strategies and machine learning-based strategies.

Technical Analysis-based Strategies: These strategies are rooted in the belief that historical price patterns and trends can provide insights into future price movements. Algorithmic trading systems use various technical indicators and chart patterns, such as moving averages, oscillators, and trend lines, to identify potential entry and exit points for trades. By automating the analysis of these technical indicators, algorithms can quickly and efficiently generate trade signals, thus minimizing the risk of human error and emotional bias. Sample studies in this area can be seen in [25, 26].

Machine Learning-based Strategies: With the advancements in artificial intelligence and machine learning, forex traders are increasingly leveraging these technologies to predict price movements and trends. Machine learning models, such as linear regression, support vector machines (SVM), and neural networks, can analyse large datasets to identify underlying patterns and relationships that may not be apparent through traditional technical analysis. These models are trained on historical data and then used to generate predictions on new, unseen data. The efficacy of such approaches has been analysed in studies such as [27, 28]

One popular machine learning technique in forex trading is time series forecasting, which uses historical price data to predict future price movements. Recurrent Neural Networks (RNNs), particularly LSTM networks, have proven effective in capturing the temporal dependencies in time series data, making them a popular choice for forex price prediction.

Another emerging approach is incorporating sentiment analysis, where algorithms analyse news articles, social media posts, and other textual data to gauge market sentiment and predict price trends accordingly. Natural Language Processing (NLP) techniques are often employed in these strategies to process and interpret vast amounts of textual data.

Despite the potential advantages of algorithmic trading in forex price and trend prediction, traders must be aware of its risks and limitations. For instance, algorithmic trading models are not foolproof and may produce false signals or fail to adapt to rapidly changing market conditions. Furthermore, developing and maintaining such trading systems require technical expertise and resources.

In conclusion, forex trading is an intricate and dynamic market that requires a deep understanding of various analytical methods, such as fundamental and technical analysis. Adopting machine learning methodologies has introduced new opportunities and challenges, paving the way for more sophisticated trading strategies and improved market predictions. As the field continues to evolve, traders must stay up-to-date with the latest developments and be prepared to adapt their strategies accordingly to remain competitive in the ever-changing forex market.

CHAPTER 3

Literature Review

In the dynamic world of financial markets, predicting currency exchange rates, or forex trading, has been a subject of study for many years. This realm, characterized by its high liquidity and 24/7 schedule, has historically proven to be a complex and challenging environment for prediction models due to its non-linear and chaotic nature. As such, traditional statistical methods often need to be revised in accurately forecasting market movements.

However, the rapid advancement in Artificial Intelligence (AI), specifically deep learning, is paving a new path for this complex task. With its core foundation in neural networks and its ability to learn from vast amounts of data, deep learning has made strides in numerous fields, like image processing, speech recognition, and autonomous driving, to name a few. These successes open exciting possibilities for its application in forex trading.

This literature review aims to delve into the current knowledge surrounding deep learning use for forex trading. By exploring the variety of deep learning models employed, the strategies for their implementation, and their performance in predicting currency exchange rates, this review aims to provide a comprehensive understanding of the current state of this field and its potential future trajectory.

Subsequent sections will specifically delve into literature that revolves around different deep learning architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), Transformer networks, attention mechanisms, and transfer learning in relation to forex trading.

3.1 Long-Short Term Memory

In the realm of forecasting financial trading markets, the Long-Short Term Memory (LSTM) model has gained considerable attention over the years. The attraction towards LSTM is primarily driven by its distinct capabilities in handling the complexities of financial data. It is noted for its exceptional ability to address the vanishing gradient problem, a challenge that tends to hamper the performance of traditional neural networks. In addition, LSTM can understand and learn long-term dependencies in time series data, an essential feature when dealing with intricate financial data.

This proficiency stems from the unique architecture of LSTM, particularly the incorporation of memory cells and gates. These components enable LSTM to retain and manage information over longer periods, thus making it a suitable choice for data streams where temporal dependencies play a significant role. Given these advanced capabilities, LSTM has been widely explored by scholars aiming to forecast the price or the trend of various financial trading activities, such as forex, stock prices, and indices. Consequently, the LSTM model holds a prominent position in financial market prediction, demonstrating its current efficacy and potential for future advancements.

A comprehensive survey by Sezer et al. [13] outlined that a broad spectrum of deep learning models, such as DMLP, RNN, LSTM, CNN, RBM, DBN, AE, and DRL, have been employed for financial market prediction. Each of these models has achieved considerable success in their specific prediction domains.

The supremacy of LSTM was further underscored in [21], where they proposed a Wavelet-LSTM model for stock price prediction. The model outshone both RNN and LSTM models in terms of performance, establishing its efficacy in the financial prediction arena.

Adding another perspective to the ongoing discussion, [29] put forth a machine-intelligent system that incorporated support vector regression optimized by GA (SVRGA) and a multilayer perceptron optimized by GA (MLP-GA). Utilizing the Percentage Price Oscillator (PPO) from the previous 8 days as input, the system predicted the PPO for the next successive 5-day period. A training model built on the Hong Kong Hang Seng Stock Index (HSI) demonstrated that the proposed predictive trading system generated three times more profits than a conventional trading system devoid of prediction, thereby highlighting the potential of advanced predictive systems in augmenting trading profits.

Moreover, the incorporation of deep reinforcement learning in quantitative trading was demonstrated by [14]. Their LSTM-based agent system learned the temporal pattern in data and adjusted it to current market conditions and historical data. Reinforcement Learning was used to train the agent to learn trading policies. The results provided further evidence for LSTM's ability to retain the temporal relation of data and showed that LSTM could be effectively used to construct a trading agent.

On the other hand, [30] introduced a Convolutional Neural Network (CNN) model to forecast monthly and weekly price trends using technical indicators, exchange rates, commodity prices, and world indices. Despite achieving a lower accuracy rate compared to the previous 81% by an SVM model[31], the results corroborated CNN's capabilities in financial time series data forecasting. Thus, while LSTM models have exhibited notable success in market prediction tasks, alternate approaches such as CNN also hold promise and merit further investigation.

In conclusion, the aforementioned studies underline the effectiveness and potential of deep learning models, especially LSTM, in financial market prediction, offering exciting opportunities for future research in this domain.

3.2 Convolution Neural Network

Convolutional Neural Networks (CNNs) have been widely adopted for image and video processing due to their unique capabilities. Comprising an input layer, numerous hidden layers, and an output layer, CNNs utilize convolutional layers within their hidden layers to extract and pass on valuable features to the subsequent layers, culminating in high-quality predictions.

The finance sector has taken an interest in CNNs' potential for predicting market trends. Experienced traders often rely on financial charts, particularly candlestick charts, as they are visually more discernible than numerical data. Over the past years (2014-2019), a body of literature analysing CNN's efficacy for forecasting financial trends using financial chart images has emerged, with most researchers using candlestick chart images as inputs.

Kim T. and Kim H. Y. [14] proposed a fusion LSTM-CNN model for stock price prediction by leveraging time series data and chart image data characteristics. This model extracted hidden patterns from stock chart images using an SC-CNN model, then applied an ST-LSTM model to close prices and trading volumes. Testing confirmed that this LSTM-CNN model outperforms standalone SC-CNN and ST-LSTM models, especially when analysing candlestick chart images.

Likewise, [16] used candlestick charts as image inputs. They constructed a CNN autoencoder to optimize sub-chart representation, followed by a 1-D CNN for price forecasting. Compared to an Index based ensemble model, their CNN model achieved higher accuracy, precision, recall, and F1-score for binary class price movement predictions. [17] employed a similar approach, yielding variable results across years, leading them to conclude that financial chart images can serve as viable classification inputs.

Exploring short-term price trend prediction, [18] employed a CNN model on candlestick charts and observed superior performance to a ResNet-18 model benchmark. [32, 33] also achieved positive results using CNN models on candlestick charts, with [32] confirming 92.1% accuracy on the Taiwan stock exchange and 92.2% on the Indonesian stock exchange.

Rather than relying on candlestick price charts, [34] generated images of technical indicators over a 15-day period. Their 6-layer CNN model demonstrated encouraging returns on selected ETFs and Dow-30 data, with annual returns exceeding those of an LSTM model. [35] also used technical indicator line charts as inputs and confirmed that CNNs outperformed ANNs.

Exploring the use of moving average line charts, [36, 37] found success in predicting weekly price movements and price volatility respectively. However, not all studies have been successful. [38] used high and low-price line charts as inputs, and was unable to achieve satisfactory results, sparking a debate on the relative merits of images versus temporal data.

Interestingly, [39, 40] adopted an alternate approach, employing the concept of images rather than actual images as inputs. These studies focused on predicting price movements and reported better accuracy than benchmark models.

In summary, this literature review showcases the potential and versatility of CNNs in financial forecasting and sets the stage for further investigation and optimization in this promising domain.

3.3 Transformer

The Transformer model, introduced in [41], constitutes a significant stride in the field of deep learning architectures. Characteristically designed to process sequential data, it comprises two principal components: the Encoder and the Decoder. The Encoder processes the input sequence, generating a context-enriched high-dimensional vector through a self-attention layer and a feed-forward neural network. This context from the Encoder is subsequently processed by the Decoder, yielding an output sequence, which could potentially be in a different language, comprised of different symbols, or even be a reflection of the input.

The Transformer has seen widespread adoption across various machine learning domains, notably in Natural Language Processing (NLP) and Computer Vision. In NLP, Transformer-based architectures, as evidenced in [42, 43], have become predominant. [42] and [43] harness auto-encoder and auto-regressive architectures, respectively, while other researchers [44–47] propose variants of BERT, with the Transformer as their base. The application of Transformers in financial trading has also been explored [19, 20], demonstrating their versatility.

In Computer Vision, the Transformer, when augmented with attention mechanisms, has demonstrated improved prediction accuracy on popular image datasets like CIFAR-10 and ImageNet [48–51]. Further advancements, such as the Vision Transformer (ViT) [52], have replaced traditional Convolutional Neural Networks (ConvNets) with self-attention mechanisms, showcasing their effectiveness.

3.4 Attention

The attention mechanism initially introduced to handle long-distance interactions in sequence-to-sequence NLP tasks, has gained immense popularity across varied research areas including image classification, object detection, and instance segmentation. Attention mechanisms aim to foster long-term dependency learning by computing relationships among words in a sentence, enabling the model to comprehend the semantics and syntax of language data in an encoder-decoder setup.

Empirical evidence supporting the effectiveness of the attention mechanism spans across a variety of studies [41, 49, 53, 54]. For instance, [55] introduced an attention-augmented convolutional network that integrated feature maps with a self-attention layer, resulting in a notable improvement in top-1 accuracy on ImageNet classification.

Moreover, the integration of self-attention with existing architectures, such as ResNet in HaloNets [49], enhanced not only the model's performance but also optimized speed and memory usage during training. Attention mechanisms have also found applications in object detection [49, 54, 56], with [54] replacing spatial convolutions with global self-attention in the final bottleneck blocks of a ResNet for significantly improved performance.

Novel implementations of the attention mechanism have also emerged in recent studies [57, 58], enabling patch-based image classification. Remarkably, [57] developed the Vision Transformer (ViT) which entirely replaced ConvNets with local multi-head dot-product self-attention blocks. This model's success further underlines the importance of a model's ability to focus attention.

The attention mechanism has also been employed in other domains such as medical image classification [52, 59–61], demonstrating its wide applicability and effectiveness. This has encouraged us to explore its potential for financial trading image classification, inspired by the body of literature that supports its exploration in this context.

3.5 Transferred learning

Transfer learning has emerged as a popular technique in machine learning and artificial intelligence. Transfer learning is a common deep learning technique where a model is created and trained on a massive dataset, and the trained model for a task can be used for a second task. It has proven to be a cost-effective and robust method, requiring fewer labeled data and producing strong results in a relatively short time.

Recent years have witnessed the application of transfer learning in various domains, including computer vision and natural language processing. In computer vision, pre-trained models on large datasets such as ImageNet are often used as feature extractors for other tasks, examples can be seen in VGG[62], Xception[63], ResNet[64], BERT[42], GPT[43], and GloVe[65].

The success of transfer learning in natural language processing is particularly exemplified by the BERT model[42]. Pre-trained on a large corpus of text data, BERT has demonstrated superior performance on a range of downstream tasks, with only minor task-specific fine-tuning.

However, transfer learning is not without challenges. There exists a recognition of certain limitations within the BERT model when applied to domain-specific areas. This challenge has prompted researchers to train domain-specific BERT variants, including SciBERT[66] for scientific publications, FinBERT[67] for financial data, and LegalBERT[68] for legal documents, achieving notable improvements in their respective fields.

In conclusion, transfer learning offers a promising avenue for improving machine learning performance, especially in tasks with limited labeled data. Future research is necessary to tackle its challenges and explore its potential further.

3.6 Conclusion

The field of Forex trading is rapidly advancing with the exploration of deep learning. Although promising results have been achieved, there is still much room for further research. Combining different deep learning architectures to create hybrid models could be an interesting avenue to explore. Additionally, it is crucial to enhance the interpretability and transparency of these models for them to be adopted in the financial sector. Optimizing these models to perform well under changing market conditions is another area of focus for future research.

CHAPTER 4

Methodology

4.1 Research Paradigm

4.1.1 Research Design

In the pursuit of advancing financial trading prediction, I proposed three distinct deep learning models in this thesis. While each of these models stands independently and addresses different facets of the problem, together, they paint a comprehensive picture of the possible applications of deep learning in this field.

In the first study, the emphasis is on designing an Event-Driven LSTM model which leverages significant financial events for extracting trading signals. The study has been designed in a way that encapsulates the core idea of event-driven trading and merges it with the capabilities of deep learning models. The objective is to establish a preliminary understanding of how event data can be intertwined with market data to aid prediction.

The second study, building upon the insights from the first one, dives deeper into improving the extraction of trading signals. Here, the design involves a complex interplay between a pre-trained image model VGG16 and a multi-head attention mechanism (MHATTN). The incorporation of MHATTN serves the purpose of utilizing information from external data sources and integrating it into the forex market graph, thereby enriching the model's capacity to comprehend and predict intermarket dependencies.

The third study elevates the research design to an even higher level of sophistication. The focus here is on introducing and validating TradingBERT, a pre-trained model specifically engineered for trading data. The design of this study includes rigorous testing of the model's efficacy through a series of downstream tasks, providing insights into the model's strengths, limitations, and potential areas for enhancement.

Together, these models exemplify the versatility and potential of deep learning in financial trading prediction. They offer unique solutions to complex problems, demonstrating how deep learning can be leveraged for enhanced predictive accuracy in different contexts within the financial trading domain.

4.1.2 Methods

The methodology adopted in this research focuses on the empirical implementation and experimentation of deep learning models on forex trading data. This emphasis on a practical, hands-on approach ensures that the research results are grounded in real-world applicability and can be directly compared and contrasted with existing trading strategies.

The first study employs an Event-Driven LSTM model, testing its ability to extract valuable signals from a complex web of forex trading data. The LSTM component's ability to learn and remember patterns over a long sequence is combined with the event-driven paradigm, wherein significant market events serve as triggers for trading decisions.

The second study implements an MHATTN-VGG16 model, creating a symbiosis between the innovative VGG16 methodology and an Attention mechanism. The VGG16 leverages relational data, allowing the model to recognize patterns within images of pricing data.

In the third study, the research takes on a new trajectory with the introduction of the Trading-BERT model. Here, the method involves pre-training the model on a large corpus of trading data, allowing it to learn and internalize patterns before it is fine-tuned on specific downstream tasks.

4.2 Deep learning models

This thesis explores three deep learning models to forecast forex price trends, which are LSTM, CNN, and Transformer. The below sub-sections provide details of these three deep learning models.

4.2.1 Long-Short Term Memory (LSTM)

The LSTM model is a modified version of RNN which effectively addresses the vanishing gradient issue. Its architecture comprises of a cell and three gates- input, output, and forget. Figure 4.1 illustrates the detailed architecture of LSTM.

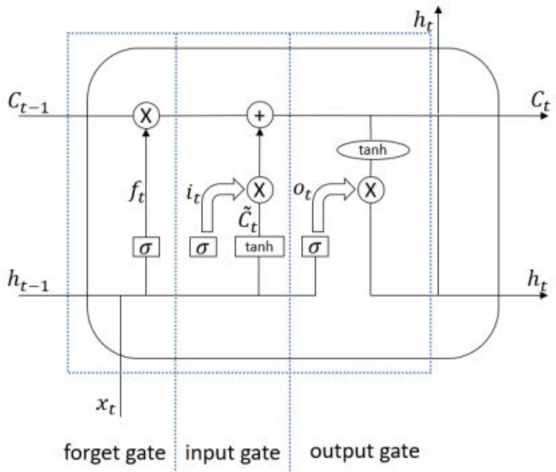

FIGURE 4.1: LSTM architecture

The forget gate examines the previous state h_{t-1} and the current input x_t, and determines which parts of the cell to discard. This decision is made through a sigmoid function f_t, which produces an output ranging from 0 to 1, where 1 means "keep" and 0 means "remove." The sigmoid function formula is represented as below:

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{4.1}$$

where W_f denotes weight function, and b_f represents bias.

The input gate, on the other hand, decides which new information will be stored in the cell. A sigmoid function i_t, with the formula given below, determines which values should be updated.

$$i_t = \sigma\left(W_i * [h_{t-1}, x_t] + b_i\right) \tag{4.2}$$

A tanh function generates a vector $\tilde{C}_t$, combined with i_t to produce an update to the cell. $\tilde{C}_t$ is denoted as below:

$$\tilde{C}_t = \tanh\left(W_C * [h_{t-1}, x_t] + b_C\right) \tag{4.3}$$

The updated cell C_t is updated by information carried over from the previous state and information generated from the current state. The formula for the C_t is:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.4}$$

Finally, the output gate determines what to output. A sigmoid function o_t filters through the cell state to select which parts to send to the output. Multiplying o_t by the cell state generates output h_t, which comes through a tanh function to constrain all values between -1 and 1. The specifics are detailed below equation:

$$o_t = \sigma\left(W_o * [h_{t-1}, x_t] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right) \tag{4.5}$$

4.2.2 Convolutional Neural Network (CNN)

CNN is the most frequently utilized deep learning model for image and video processing. The architecture usually comprises four types of layers: convolutional, ReLU (Rectified Linear Unit), pooling, and fully connected. creffig:cnn depicts CNN architecture.

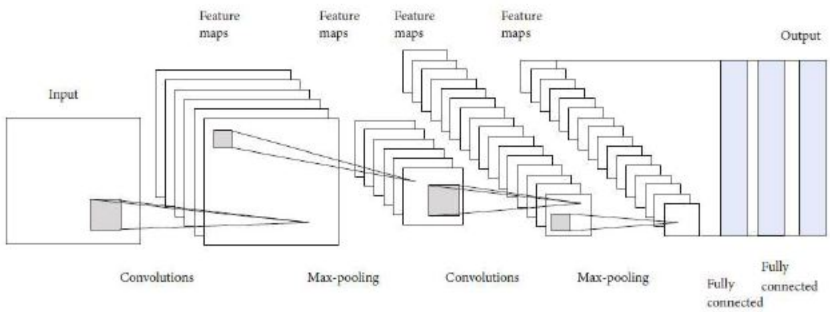

FIGURE 4.2: CNN architecture

Convolutional Layer: The primary purpose of a convolutional layer is to detect the combinations of features within a given area of the previous layer. It does this by applying small, learnable filters (also known as kernels) to the input volume. These filters extend through the full depth of the input but have small spatial dimensions. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the filter and the input at each position. This process results in a 2-dimensional activation map.

The equation for this convolution process when the filter has size F and the input size is N, with padding P and stride S is:

$$\text{Output Size } = (\mathbf{N} - \mathbf{F} + \mathbf{2P})/\mathbf{S} + 1 \tag{4.6}$$

The activation map shows where the kernel's unique features were found in the input image. The AI network trains filters to recognize specific visual elements, like edges or colors, in the first layer. As the layers become more complex, the network can identify more intricate patterns, such as honeycomb or wheel shapes.

ReLU (Rectified Linear Unit) Layer: After each convolution operation, the network applies a non-linear operation in the form of ReLU. The ReLU function returns 0 if the input is less than or equal to 0, and it returns the input itself if it is greater than 0. The purpose of this layer is to introduce non-linearity in the network. The equation of ReLU can be described below:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{4.7}$$

Pooling Layer: The pooling layer also known as a downsampling layer, is periodically inserted in-between successive convolutional layers in a CNN architecture. Its function is to progressively reduce the spatial size (width and height) of the input volume. This serves to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially. The most common approach used in pooling is max pooling.

Fully Connected Layer: The Fully Connected layer is a traditional Multi Layer Perceptron (MLP) that uses a softmax activation function in the output layer. The term "Fully Connected" implies that every neuron in the previous layer is connected to every neuron on the next layer.

4.2.3 Transformer

The Transformer is a model architecture introduced in [41] for improving the speed of training and the quality of machine translation models. Transformers have since become a cornerstone of many models in natural language processing (NLP), including GPT (Generative Pretrained Transformer) and BERT (Bidirectional Encoder Representations from Transformers).

The Transformer consists of an encoder and a decoder, each of which is a stack of identical layers. The original Transformer has six layers in the encoder and the decoder. Figure 4.3 describes the transformer architecture.

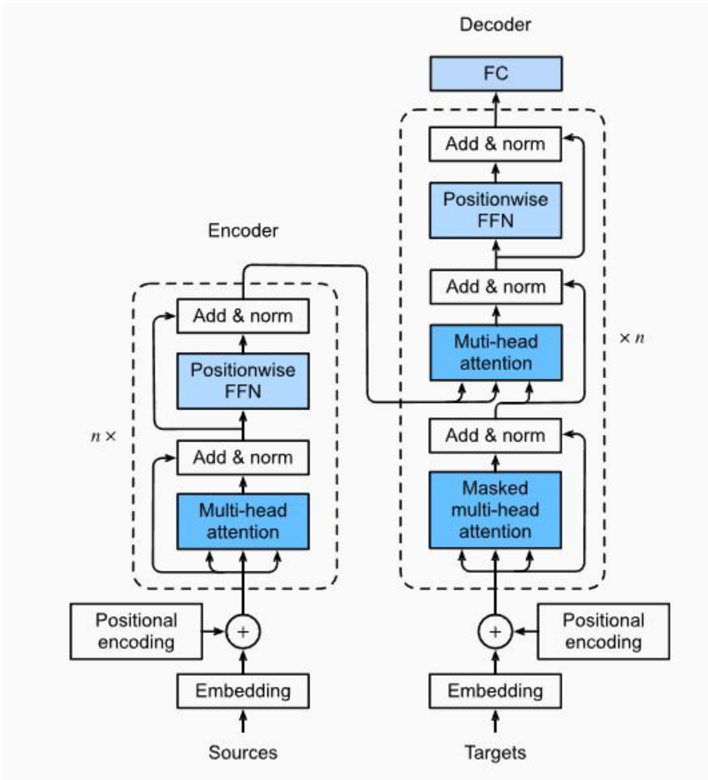

FIGURE 4.3: Transformer architecture

Encoder: Each layer in the encoder has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The input to each layer is added to its output through a residual connection and then normalized.

Decoder: The decoder also has two sub-layers similar to the encoder but has a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, residual connections and normalization are applied.

4.3 Data Collection

The primary source of data for this research is procured from Oanda API. Oanda Corporation is a globally recognized forex trading and currency information services company that provides traders access to the international currency markets. Notably, their services include access to historical currency trading data, which is an invaluable resource for financial research, such as the work presented in this thesis.

The dataset extracted for this study consists of Open, High, Low, and Close (OHLC) price data. OHLC data provides a comprehensive summary of the trading activity within a given time period, delivering key insights into market dynamics. Each data point is representative of the opening price, highest price, lowest price, and closing price of a currency within a specific trading timeframe.

This study focuses on four currency pairs: AUD/USD, EUR/GPB, USD/JPY, and CAD/CHF. These pairs have been chosen due to their high liquidity and volatility, providing a rich and robust environment for model testing and evaluation.

This research employs data spanning from January 2005 to June 2022, collected at a frequency of 15-minute intervals. The choice of 15-minute intervals provides an appropriate balance between data granularity and the quantity of data points, which is a significant factor for deep learning models that require substantial amounts of data to function effectively.

During the data collection process, missing values were identified and calculated. The percentage of missing data was found to be less than 3% for all currency pairs, a negligible amount that would have a marginal impact on the overall dataset and its subsequent analysis. Therefore, rather than applying imputation techniques or other strategies for handling missing data, these instances were ignored. This decision also mitigates the risk of introducing any bias or inaccuracy into the dataset that could potentially arise from data imputation techniques. This meticulous approach to data collection and handling ensures the integrity and reliability of the analysis and results presented in this study.

Details of the data used for each of the three distinct studies can be found in Chapters 5, 6, and 7.

4.4 Evaluation

Two distinct objectives predict the performance of the models developed in this research study. Firstly, a regression objective targeting future forex prices, and secondly, a classification objective indicating a trading signal (i.e., 'buy', 'hold', 'sell'). Depending on each study's specific objective, I forecast the future price, predict the trading signal or both.

4.4.1 Regression Evaluation Metrics

In order to assess the success of my models in accurately predicting forex prices, I rely on a suite of standard regression metrics.

Mean Square Error (MSE): The MSE measures the average squared difference between the estimated and actual values, quantifying the difference between the predicted and actual values. The MSE formula is as follows:

$$MSE = \frac{\sum_{i=1}^{n}(\text{ True } - \text{ Prediction })^2}{n} \tag{4.8}$$

Root Mean Squared Error (RMSE): The RMSE is the square root of the MSE, providing a measure of the error rate in the same units as the predicted and actual values. The RMSE formula is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\text{ True } - \text{ Prediction })^2}{n}} \tag{4.9}$$

Mean Absolute Error (MAE): The MAE computes the average absolute difference between predicted and actual values, giving an insight into the magnitude of the error without considering direction. The MAE formula is as follows:

$$MAE = \frac{\sum_{i=1}^{n} \mid \text{ True } - \text{ Prediction } \mid}{n} \tag{4.10}$$

Mean Absolute Percentage Error (MAPE): The MAPE provides a percentage-based version of the MAE, indicating the average absolute percentage difference between predicted and actual values. This is particularly useful in understanding the error rate in terms of percentage. The MAPE formula is as follows:

$$MAPE = \frac{\sum_{i=1}^{n} \frac{\mid \text{True} - \text{Prediction} \mid}{\text{True}}}{n} * 100 \tag{4.11}$$

I have selected these metrics because they offer a comprehensive understanding of the performance of my models in terms of both the magnitude and direction of the errors. Furthermore, they allow us to compare the effectiveness of different models and configurations in predicting forex prices.

4.4.2 Classification Evaluation Metrics

I employ several standard classification evaluation metrics to assess my models' performance in predicting trading signals.

Accuracy: This metric calculates the proportion of true results among the total number of cases examined.

Precision: This metric measures the proportion of true positive predictions among all positive predictions.

Recall: This metric quantifies the proportion of true positive predictions among all actual positive instances.

F1 Score: This metric combines precision and recall into a single value that balances the two.

Due to the imbalance classes, to provide a comprehensive evaluation, the macro-average (averaging the metric for each label and then taking the average) and weighted average (considering the number of true instances for each label) of precision, recall, and F1 score are employed. Their formulas are as follows:

$$\text{Weighted Average } = \Sigma(wi \star xi)/\Sigma wi \tag{4.12}$$

where Σ denotes the sum over all data points, wi is the weight corresponding to the data point xi.

$$\text{Macro Average } = (s1 + s2 + \ldots + sn)/n \tag{4.13}$$

where Σ denotes the sum over all classes, si is the score for the $i - th$ class, and n is the total number of classes.

The above metrics were chosen because they not only provide a broad view of model performance (accuracy) but also allow us to understand the model's strengths and weaknesses in predicting each type of trading signal (precision, recall, F1 score).

4.4.3 Relation to Research Goals

These evaluation metrics align with my research goals by comprehensively assessing model performance from multiple perspectives. By using a variety of metrics, a nuanced understanding of the models' effectiveness is gained in predicting forex prices and trading signals, thus guiding future model development and tuning.

4.4.4 Potential Limitations or Challenges

While these evaluation metrics are widely used and provide valuable insights, they are not without limitations. MSE, RMSE, MAE, and MAPE are all susceptible to the impact of outliers, which could skew the performance evaluation of the models. Similarly, accuracy can be a misleading metric if the classes are imbalanced. Moreover, precision, recall, and F1 score, while more nuanced, depending on the chosen threshold for classification, can influence the results. Therefore, it is essential to interpret the results of these metrics within the context of the specific data and task.

CHAPTER 5

Event-Driven LSTM For Forex Price Prediction

5.1 Introduction

Foreign Exchange (forex) Trading represents the largest financial market globally, with an extensive array of international participants, ranging from professional traders to individual investors. It is characterized by robust liquidity, leading to potentially significant profits for those able to accurately predict market trends and establish appropriate entry and exit points. Despite this potential, the forex market's inherently non-stationary, highly volatile, and complex nature makes designing such predictive models a formidable challenge.

Over the past decade, machine learning has emerged as a vital tool in studying the exchange rate market. Researchers have sought to leverage machine learning algorithms in the analysis of continuous historical price and technical analysis data, with a common focus on identifying underlying patterns that might recur in the future. Despite these efforts, many machine learning-based approaches fall short when deployed in the real-world market due to their non-stationary and high volatility.

Historical data and technical indicators are fundamental in identifying trends within trading windows. The application of technical analysis can filter out market noise and enhance trend identification, making it a popular tool in trading research. Several studies, for example, have predicted changes in the forex market direction using the closing price and moving average, with the moving average showing superior performance. However, while this approach is commonplace, it is not without limitations given the market's complexity and the substantial amount of noise in high-frequency trading data.

In response to these challenges, this paper introduces an LSTM approach, employing event-driven features and deep learning models to predict retracement points that provide optimal entry points for maximum profit. Unlike previous research that indiscriminately uses all historical data, a data selection technique that focuses on data suggesting reliable trading opportunities is proposed.

The proposed methodology incorporates LSTM models, among others, to predict the price at the retracement point using technical indicators as features. A comparative analysis of these models against a simple Recurrent Neural Network (RNN) baseline model shows promising results, with my proposed models consistently outperforming the baseline. This innovative approach could pave the way for a robust predictive system that supports accurate trading strategies while minimizing risk, potentially revolutionizing future research in financial trading.

This chapter further delves into the related work in this field, the process of feature and training data selection, the design of the model architecture, experiment details, and the results. The final section presents the conclusions based on these findings.

5.2 Methods

5.2.1 Feature and Training Data Selection

This study is based on an LSTM architecture, which is enhanced by a feature selection method. This method uses a sequence of events derived from the Elliott Wave theory[69] to identify peaks and troughs in market trends. This sequence is further augmented with a ZigZag indicator[70], a tool widely used in technical analysis[4], to determine the high and low points within a specified period.

This technical indicator can be adjusted using three parameters: Depth, Deviation, and Backstep, each defining specific properties of the candidate Peak or Trough. The ZigZag points are integrated as transition points in my method.

To complement ZigZag, a moving average crossover event is employed, where the intersection of two or more moving average lines often indicates a trend change. Lastly, a retracement point was detected and used as the prediction target, signifying any temporary reversal in price within a major price trend.

In addition to these events, 28 technical indicators are selected as features. These were derived from six types of indicators with different window sizes, using the TA-LIB package. These indicators include the Moving Average Convergence Divergence (MACD)[71], Simple Moving Average (SMA)[5], Relative Strength Index (RSI)[7], Average Directional Index (ADX)[72], Bollinger Band Indicator (BB)[73], and William R Indicator (WR)[74].

The training data was selected with a focus on reducing noise from highly volatile trading data. All data at a crossover event and retraced back 'n' timesteps are chosen for training, aiming to predict the price at the retracement point.

Table 5.1 provides the total number of raw data and the sequences of events for each currency pair.

Currency Pair	**Number of rows**	**Number of sequences (e1, e2, e3)**	**Training data**	**Test data**
GBP/USD	397,317	2,238	1,838	400
EUR/GBP	397,796	2,233	1,810	423
AUD/USD	397,329	2,260	1,849	411
CAD/CHF	397,813	2,194	1,770	424

TABLE 5.1: Raw data details

5.2.2 Model Architecture

The chosen model architecture is a Recurrent Neural Network (RNN), which excels at processing sequences of data while maintaining an internal memory. However, standard RNNs face gradient vanishing and exploding issues, leading us to opt for a Long-Short Term Memory (LSTM) network.

The LSTM architecture is equipped with a cell and three gates – an input gate, an output gate, and a forget gate. This design allows the LSTM to handle the vanishing gradient problem common in RNNs. Each gate has a specific function, enabling the LSTM to decide which parts of information should be stored, discarded, or sent to the output.

Bidirectional LSTM (BiLSTM) and Gated Recurrent Unit (GRU) are also incorporated into this study. BiLSTM, an upgraded version of LSTM, processes inputs in two ways - past to future and future to past - while GRU, another LSTM variant, comes with an update gate and a reset gate, deciding what parts of the information should be sent to the output.

The proposed LSTM model is composed of an input layer, two LSTM (or BiLSTM/GRU) layers with 64 hidden units each, and a dense layer. These three models were trained with the Adam optimizer and were designed to inform us when to enter the market by forecasting price at the retracement point.

Through this methodological design, accurate prediction of the optimal entry price into the market will be translated into a 'buy' order in a bullish market or a 'sell' order in a bearish market, thereby leading to profitable trading decisions.

Model	**Timesteps**	**MSE** (10^{-3})	**RMSE** (10^{-3})	**MAE** (10^{-3})	**MAPE** (%)
RNN	30	1.846	42.960	35.336	2.774
LSTM	30	2.435	49.351	48.357	3.749
BiLSTM	30	1.520	38.984	37.456	2.903
GRU	30	6.311	79.441	78.185	6.067
RNN	60	0.710	26.645	25.118	1.945
LSTM	60	0.853	29.200	26.617	2.075
BiLSTM	60	1.489	38.593	37.098	2.873
GRU	60	3.324	57.658	56.723	4.401

TABLE 5.2: Experiment result for GBP/USD

Model	**Timesteps**	**MSE** (10^{-3})	**RMSE** (10^{-3})	**MAE** (10^{-3})	**MAPE** (%)
RNN	30	0.032	5.628	4.456	0.503
LSTM	30	0.006	2.407	1.708	0.194
BiLSTM	30	0.168	12.962	12.701	1.440
GRU	30	0.151	12.273	11.343	1.277
RNN	60	0.075	8.684	7.270	0.817
LSTM	60	0.006	2.536	1.957	0.222
BiLSTM	60	0.076	8.705	8.434	0.957
GRU	60	0.135	11.612	10.548	1.187

TABLE 5.3: Experiment result for EUR/GBP

5.3 Results

In the experimental phase of this study, The performance of various models, namely, LSTM, GRU, and RNN, using different timesteps are analysed. The results are presented in tables 5.2 to 5.5.

The results show that LSTM with 30 timesteps demonstrated superior performance for predicting the EUR/GBP currency pair. Similarly, the GRU model with 30 timesteps excelled for the AUD/USD and CAD/CHF pairs. It is noteworthy to mention that these three models outperformed the baseline RNN model. However, an exception was noted in the GBP/USD pair case, where the RNN model with 60 timesteps turned out to be the best.

Model	Timesteps	MSE (10^{-3})	RMSE (10^{-3})	MAE (10^{-3})	MAPE (%)
RNN	30	0.530	23.030	20.406	2.875
LSTM	30	0.568	23.835	22.481	3.149
BiLSTM	30	4.821	69.433	68.646	9.566
GRU	30	0.201	14.162	13.588	1.929
RNN	60	1.061	32.578	31.106	4.377
LSTM	60	0.345	18.585	16.365	2.291
BiLSTM	60	6.603	81.263	80.897	11.320
GRU	60	0.298	17.270	16.626	2.351

TABLE 5.4: Experiment result for AUD/USD

Model	Timesteps	MSE (10^{-3})	RMSE (10^{-3})	MAE (10^{-3})	MAPE (%)
RNN	30	0.089	9.450	7.388	1.023
LSTM	30	0.092	9.570	8.537	1.167
BiLSTM	30	0.225	15.011	14.312	1.931
GRU	30	0.027	5.247	4.264	0.583
RNN	60	0.183	13.543	12.030	1.641
LSTM	60	0.154	12.396	11.088	1.514
BiLSTM	60	0.032	5.674	4.858	0.651
GRU	60	0.032	5.661	4.340	0.603

TABLE 5.5: Experiment result for CAD/CHF

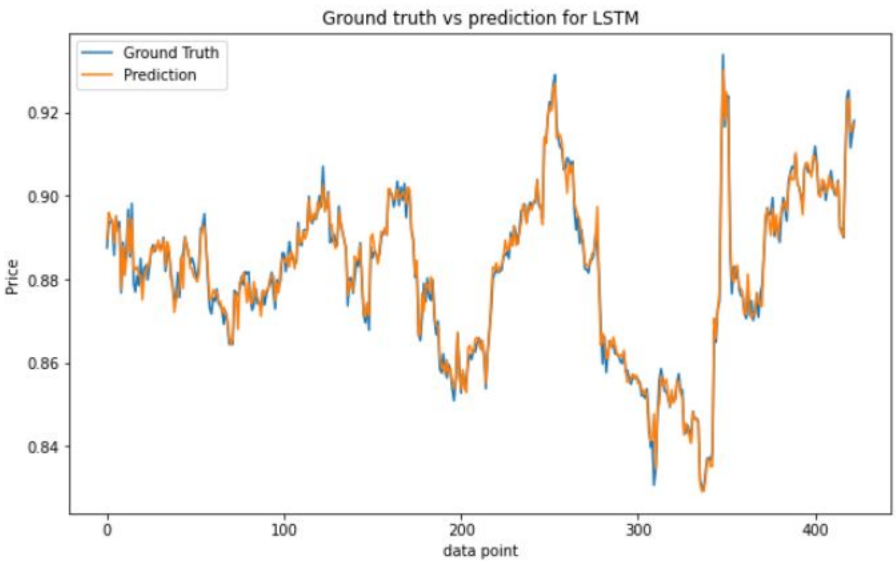

FIGURE 5.1: Real prices vs predicted prices – EUR/GBP

Figure 5.1 presents a comparative analysis of the predicted price and the actual (ground truth) price at the retracement point for the EUR/GBP pair. The predicted values were observed in close alignment with the actual values, demonstrating a Mean Absolute Percentage Error (MAPE) of only 0.194%. This low error percentage indicates the model's ability to provide reliable predictions, thus supporting real trading operations.

It should be noted that the selection of the best model was based on the pair-specific performance. This indicates the potential of using tailored models for individual currency pairs to achieve optimal trading decisions. The evaluation of other currency pairs and further tuning of the models could be areas for future work to improve the precision of predictions and thus the profitability of trading.

5.4 Discussion

This study aimed to predict future price movements in the forex market using a combination of historical data, technical analysis indicators, and advanced AI techniques. A significant part of this endeavor was the construction of an architecture consisting of a training data selection module and an LSTM model for predicting price at retracement points. The predictions generated by this system can aid traders in formulating their trading strategies with minimized risk.

The results demonstrate the potential for utilizing a combined approach in the development of effective trading strategies. However, one limitation encountered was related to the use of ZigZag indicators for predicting the "e1" element. The ZigZag indicator typically requires approximately an additional 40 bars to confirm a ZigZag transition point. This presents an area for future research, specifically in identifying ZigZag or similar price transition points based purely on historical data.

5.5 Conclusion

In conclusion, this study contributes to the existing body of research in a number of significant ways. Firstly, a method for selecting training data based on a sequence of events derived from the Elliott Wave was introduced, which successfully reduces noise from frequent time series trading data. Secondly, the LSTM and GRU models outperformed the baseline RNN model in predicting three currency pairs: EUR/GBP, AUD/USD, and CAD/CHF. Such results demonstrated a significant increase in MAPE - 159% for EUR/GBP, 49% for AUD/USD, and 75% for CAD/CHF.

Thirdly, in addition to predicting entry price points, the optimal number of timesteps for capturing underlying price patterns was tested and determined. The results indicate that different currency pairs respond differently to varying timesteps.

These findings underscore the potential for developing a Robotic Trading (RoboTrading) platform that would leverage the approach in this study to produce more accurate and profitable trading decisions. Further research and development in this direction could greatly contribute to the field of automated forex trading.

CHAPTER 6

Multi-head Attention VGG For Forex Trading Image Classification

6.1 Introduction

Experienced forex traders can identify trading opportunities by analyzing price charts, with candlesticks being a fundamental element used to depict price movements at specific time points. The structure of candlesticks, including open, high, low, and close prices, is explained in Figure 6.1. A candlestick chart, as shown in Figure 6.2, is created by a library of candlesticks at each time point, and traders can observe price patterns within the chart. For instance, Figure 6.2 highlights a morning star doji pattern[75]. To gain a comprehensive view of price movement, traders often include volume and other indicators such as RSI, MACD, and Moving Average (MA) alongside the candlestick chart, which assists in determining their trading strategy.

While CNNs have achieved significant success in learning image representations, they are not widely used in forecasting trading opportunities based on price charts. This paper introduces a multi-head attention model, MHATTN-VGG16, for forex trading image classification and prediction of forex price movement. Attention is adopted in this study to capture global and local information, with a two-dimensional positional encoding tailored to identify where and what the model should pay attention to within the image. The goal of this study is to create a model that mimics human behavior when analyzing trading patterns from images.

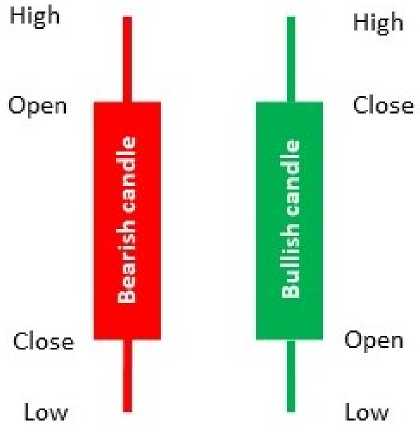

FIGURE 6.1: Candlestick

FIGURE 6.2: Candlestick chart

Key contributions from this work include:

- Development of the innovative MHATTN-VGG16 architecture which skillfully combines the VGG16 architecture, the attention mechanism, and positional encoding.
- The introduction of two-dimensional positional encoding using sine and cosine functions. This allows the model to learn from the location of the image and decide where to direct its attention.

The replication of experienced traders' behavior of scanning financial charts for trading opportunities. By enhancing VGG16 with the attention mechanism, I've extended its application into the realm of financial trading chart image analysis.

6.2 Methods

This section details the structure of the MHATTN-VGG16 model, the creation and labeling of financial chart images for predictions, and the data used in this study. The model was created using the Python Keras modules for deep learning and trained in Google Colab TPU.

6.2.1 MHATTN-VGG16 Architecture

The introduced MHATTN-VGG16 is a novel architecture that leverages the strengths of the VGG16, attention mechanism, and positional encoding and is developed to classify financial chart images. MHATTN-VGG16 first employs VGG16 for feature extraction from images, then the extracted features are inputted to the multi-head attention layer, and the computed two-dimensional positional encoding is added to the multi-head attention output to capture location-specific information. The attention output is then used as input for five fully connected layers and processed through a softmax layer for the final output. The complete MHATTN-VGG16 architecture is presented in Figure 6.3.

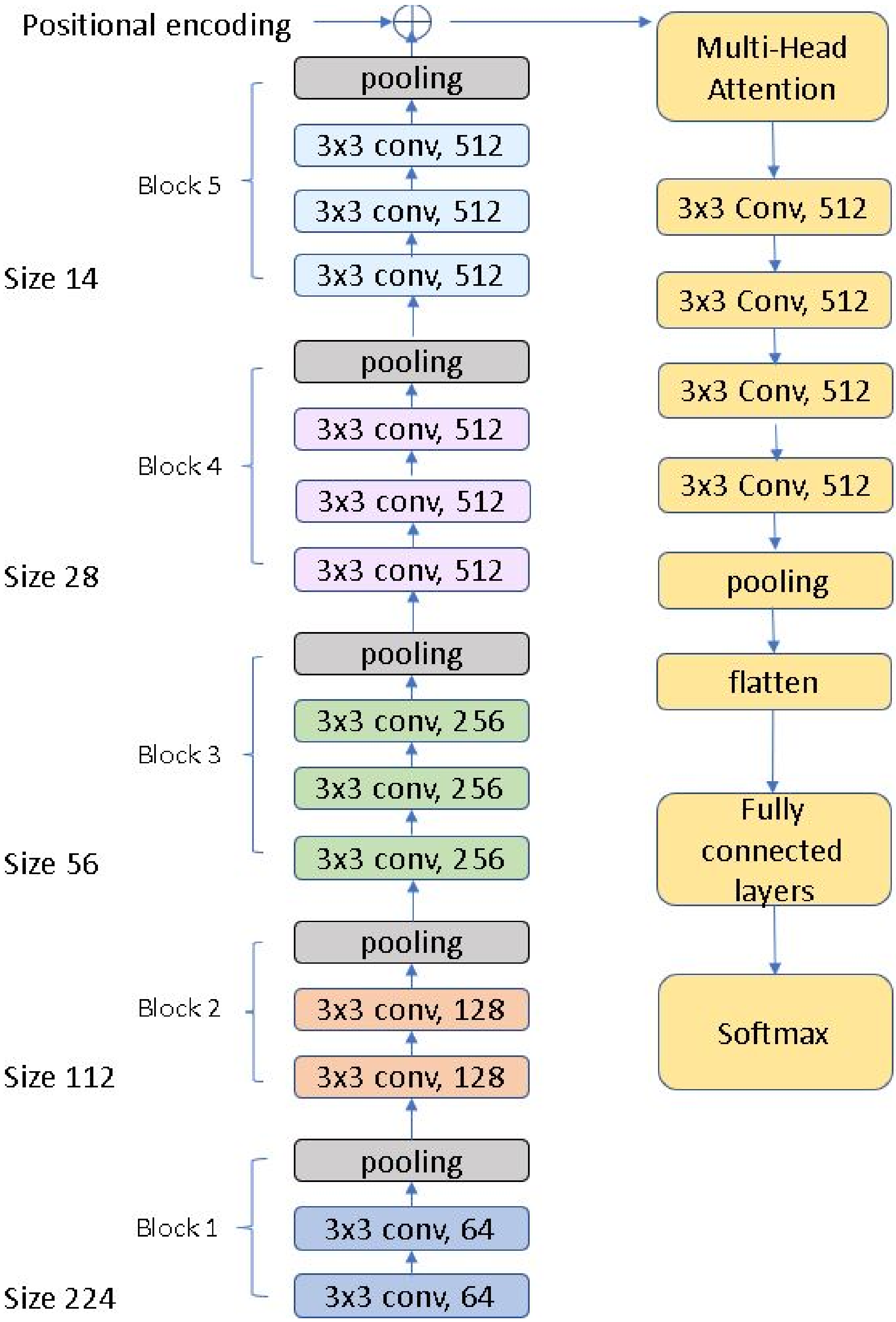

FIGURE 6.3: MHATTN-VGG16 architecture

6.2.1.1 VGG16

VGGNet, a Convolutional Neural Network (CNN) architecture, was proposed by Simonyan K. and Zisserman A. in 2014[62] and has been a critical contribution to the field of image processing. The architecture, notably the VGG16 variant, has found wide application due to its impressive depth with 13 convolutional layers and 3 fully connected layers, allowing for intricate feature extraction from images. One main application of VGG16 has been in transfer learning, where pre-trained VGGNet models have been used to extract features for image classification tasks. A typical VGG16 architecture can be described in Figure 6.4.

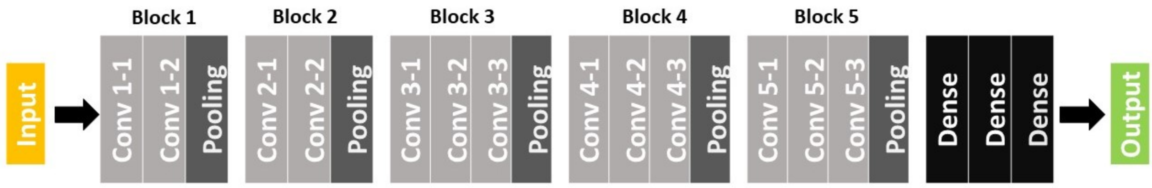

FIGURE 6.4: VGG16 architecture

In this study, only the 13 convolutional layers from blocks 1 to 5 are utilized for feature extraction. The input to these layers is a 224x224x3 image. Block 1 operates on 112x112, block 2 operates on 56x56, and block 3 operates on 28x28 through 3 convolutional layers and a pooling layer. Block 4 has a similar structure, and the final block 5 generates 7x7x512 feature vectors.

6.2.1.2 Attention

The attention mechanism, first introduced in [76], was developed to tackle the information bottleneck that arises in machine translation systems when long sentences are encoded into fixed-length vectors. This mechanism enhances the long-distance relationship-capturing ability of these systems.

A notable limitation of VGGNet is its inability to adequately highlight important features, resulting in equal importance placed on all features. This lack of prioritization could not provide a global perspective of an image, causing a deficiency in long-range dependencies and interactions. To alleviate this shortcoming, the attention mechanism is introduced in the MHATTN-VGG16 architecture to discern the areas of importance in the image input. The attention mechanism is performed by the scaled dot-product attention whose formula can be denoted as below:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6.1}$$

Where Q is the vector representing the item to calculate attention for, K and V are vectors representing the sequence to calculate attention against, and d_k is the dimension of keys. When the alignment between Q and K is good, the dot product of these vectors increases. These computed dot products are then scaled by dividing by $\sqrt{d_k}$ and applying a softmax function to obtain the weights. The weights, which sum to 1, provide an idea of which vectors in the keys are better aligned with the query. These weights are then used to calculate the final attention score. The scared dot-product attention can be described in Figure 6.5.

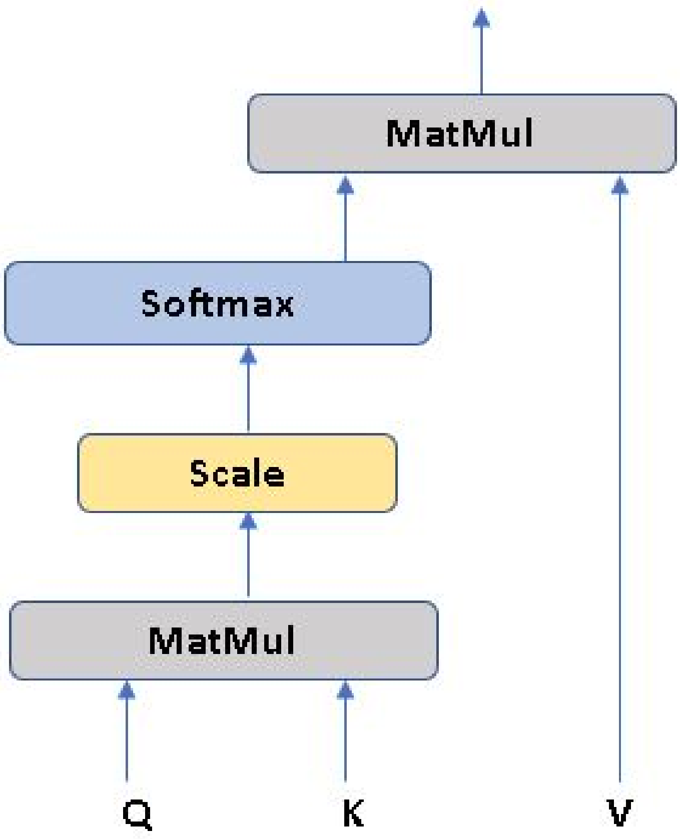

FIGURE 6.5: Scaled dot product attention

An improvement to the attention system was introduced by [41] in the form of a multi-head attention transformer. Using multiple attention heads instead of one allows the model to run the attention process several times in parallel. This gives the model the ability to attend to different parts of the input and understand multiple relationships at once. Figure 6.6 illustrates multi-head attention.

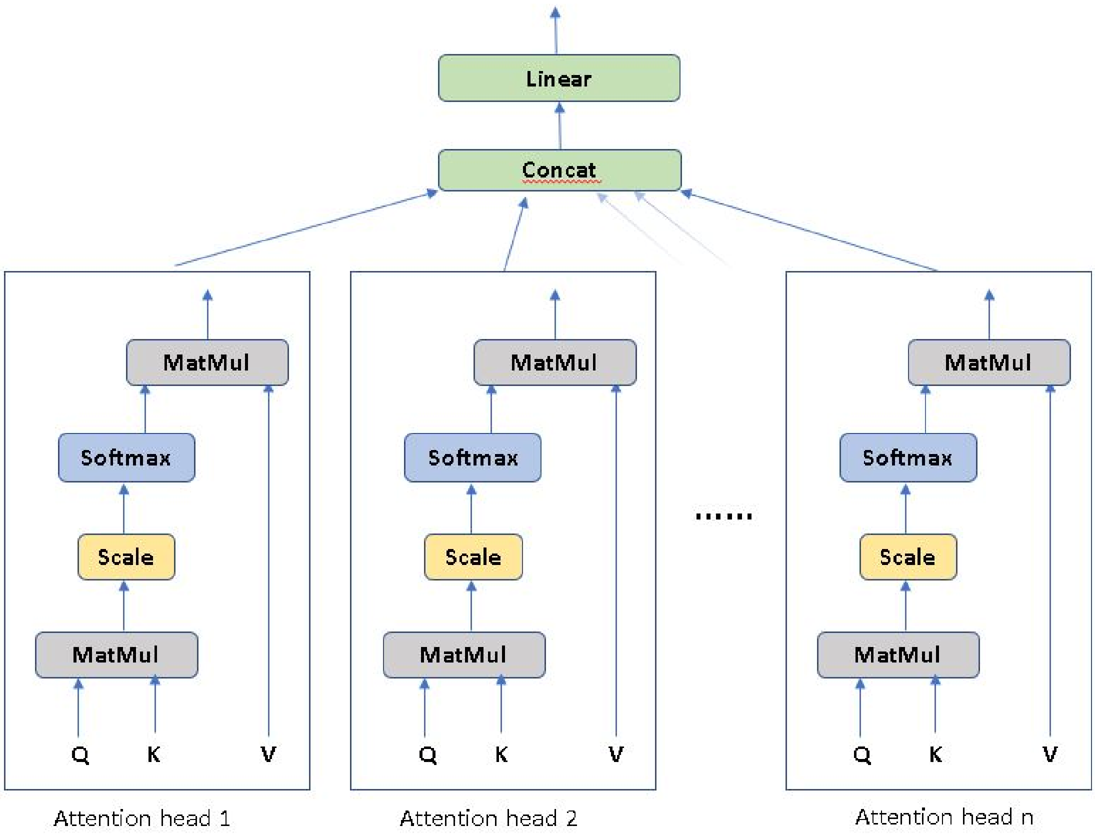

FIGURE 6.6: Multihead attention

6.2.1.3 Two-dimentional Positional Encoding

Since the input to the attention layer consists of vectors without a specific order, the concept of 'positional encoding' is adopted in MHATTN-VGG16. This ensures the model gets information about the position of an item in the sequence.

Positional encoding represents the location of an item in a sequence dimensionally. This is added to the embedding vector to provide the model with location-specific information. While previous study[41] has used a one-dimensional positional encoding from a sinusoidal function, this is inadequate for image classification due to the three dimensions of an image - height, width, and channel. In this paper, a two-dimensional representation[77] generated from a sinusoidal function is used to overcome this issue, producing a three-dimensional vector. This positional vector is added to the multi-head attention layer to capture location-specific information and enable the model to learn where to direct attention.

The formula for the one-dimensional positional encoding can be denoted as follows:

$$PE_{(\text{pos},2i)} = \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \tag{6.2}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{6.3}$$

While the formula for the two-dimensional positional encoding is as below:

$$\begin{gathered}
PE(x, y, 2i) = \sin\left(x/10000^{4i/D}\right) \\
PE(x, y, 2i+1) = \cos\left(x/10000^{4i/D}\right) \\
PE(x, y, 2j+D/2) = \sin\left(y/10000^{4j/D}\right) \\
PE(x, y, 2j+1+D/2) = \cos\left(y/10000^{4j/D}\right)
\end{gathered} \tag{6.4}$$

These equations add a unique identifier to each position, allowing the model to differentiate the order of items and understand their relative positions.

6.2.2 Labeling of Data

This study aims to predict the maximum price movement for Forex trading. Labels for the max price movement are created based on the difference between the current closing price and the highest price within the next five windows. These price differences are categorized at an interval of every five pips. Figure 6.7 presents the algorithm used to calculate these max price movement categories. Table 6.1 gives a total number of data points to train, validate, and test models. Table 6.2 displays the total number of training data points in each class for all four currency pairs under investigation. The classes less than -2 and greater than 22 are ignored due to the extremely low number of records.

Algorithm 1: label of max price movement.
Total number of observations is n, the target label l_{max} is the maximum price movement category in the next 5 windows. Each observation has four prices which are open, high, low, and close prices. We use p_o, p_h, p_l and p_c to represent these four prices.

```
For i=0, ..., n-5 do
    sequential sample[t] = {i+1, ..., i+5}
    p_max = max_{j∈[t]} p_h(j)
    l_max = floor((p_max − p_c(i))/pip)
End for loop
```

FIGURE 6.7: Algorithm to calculate max price movement label

	EUR/GBP	**AUD/USD**	**USD/JPY**	**USD/CHF**
Train 2018-01-01 to 2020-09-30	71,229	71,144	71,251	71,105
Validation 2020-10-01 to 2021-03-31	12,797	12,789	12,794	12,761
Test 2021-04-01 to 2021-07-31	8,674	8,666	8,675	8,638

TABLE 6.1: Total number of data points for train, validation, and test

6.2.3 Generation of Financial Chart Images

The financial chart image used in this study includes three sections, featuring key technical analysis indicators: the Simple Moving Average, Relative Strength Index (RSI)[7], and Moving Average Convergence Divergence (MACD)[71].

Classes	EUR/GBP	AUD/USD	USD/JPY	USD/CHF
$<$ -2	2	8	14	4
-2	10	7	5	5
-1	1,330	2,335	1,461	2,068
0	39,754	36,308	35,301	36,782
1	16,286	18,327	18,524	18,165
2	6,916	7,750	8,096	7,631
3	3,265	3,179	3,593	3,233
4	1,588	1,384	1,762	1,538
5	785	662	913	783
6	476	442	523	374
7	238	243	337	187
8	188	164	218	128
9	84	109	133	80
10	82	66	73	34
11	56	35	70	30
12	44	32	53	20
13	38	24	48	13
14	304	14	18	7
15	17	12	26	3
16	14	7	20	3
17	16	14	14	3
18	2	8	8	2
19	5	5	9	12
20	2	7	4	2
21	2	5	1	2
22	1	5	5	0
$>$ 22	5	19	15	0

TABLE 6.2: Total number of training data points by class

The topmost section of the chart consists of a candlestick graph, displaying open, high, low, and closing prices alongside two simple moving averages at window sizes of 5 and 20. The middle section exhibits the MACD, while the bottom section portrays the RSI with two threshold lines. The 70% and 30% markers are generally considered indicative of overbought and oversold conditions, respectively.

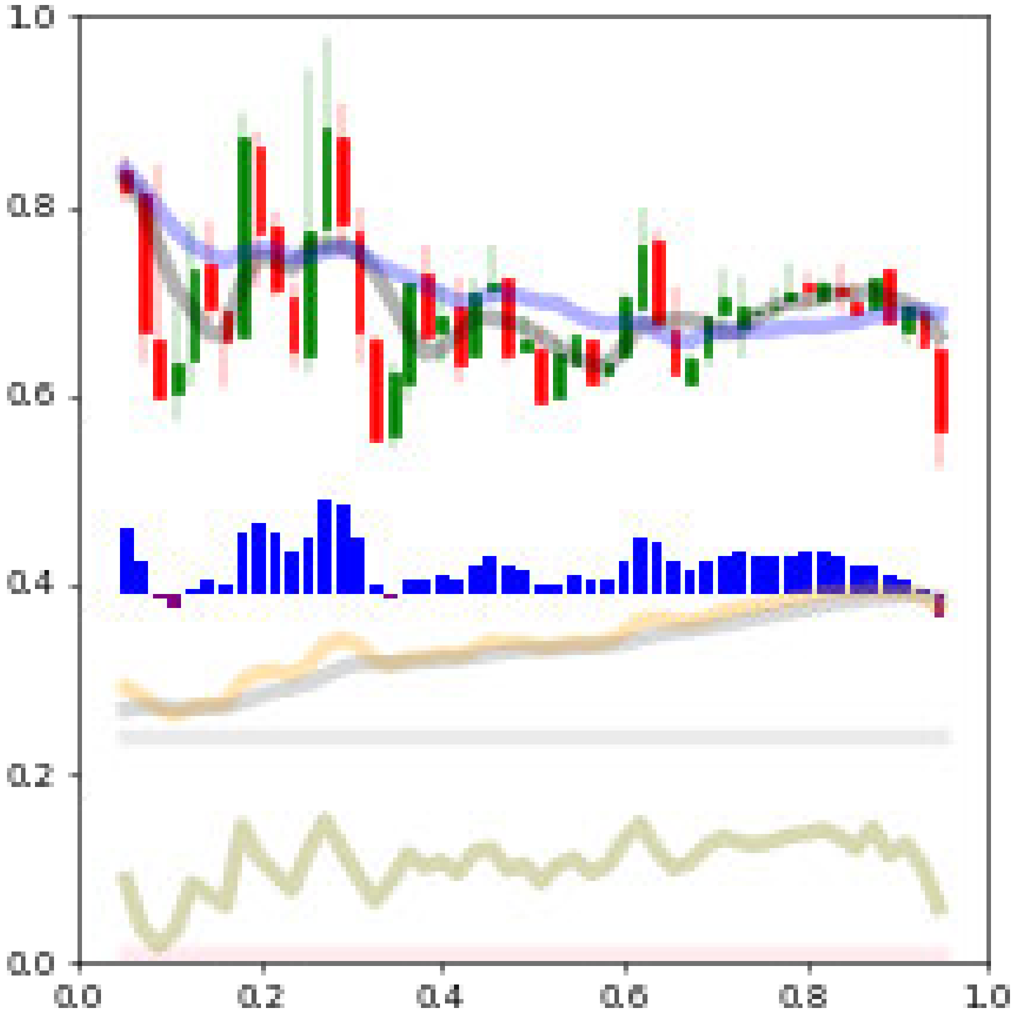

FIGURE 6.8: Example of forex trading chart image

Each section of the trading image employs RGB colors. The image, depicting time series data between t-50 and t, has a size of 3x224x224 (channel x width x height). Thicker lines are employed for each indicator to better capture chart information. Figure 6.8 shows an example of the financial chart image used in this study.

Model	Accuracy			
	Top-1	Top-1 weighted average	Top-3	Top-5
VGG16	36.67%	44.00%	75.54%	90.84%
ResNet50	32.12%	38.67%	68.71%	89.48%
Xception	30.86%	37.45%	68.75%	88.33%
MHATTN-VGG16	43.31%	48.33%	82.80%	96.81%

TABLE 6.3: Experiment result for EUR/GBP

6.3 Results

The models for this study were trained using Google's Tensor Processing Units (TPUs), with a batch size of 128. The Adam optimization algorithm is employed, the learning rate is set to 1e-5. The number of attention heads is 8. To avoid overfitting, an early stopping criterion with a patience level of 30 and a maximum training epoch of 100 is established.

Given the clear imbalance in the labels, class weights are implemented during model training. The models are assessed based on three primary evaluation metrics: top 1 accuracy, top 3 accuracy, and top 5 accuracy.

The proposed MHATTN-VGG16 model was compared to three other baseline models, namely VGG16, ResNet50, and Xception, all of which were pre-trained on 'imagenet'. The comparison was carried out on all four currency pairs and the findings are shown in Table 6.3, table 6.4, table 6.5 and table 6.6.

Based on the evaluation metrics including top 1, top 1 weighted average, top 3, and top 5 accuracy, the MHATTN-VGG16 model outperformed all baseline models across all four currency pairs. The results showed that for EUR/GBP, the top 1 accuracy increase ranged from 6.64% to 12.45%, the top 1 weighted average increased from 4.33% to 10.88%, the top 3 accuracy increased from 7.26% to 14.09%, and the top 5 accuracy increased from 5.97% to 8.48%. For AUD/USD, the top 1 accuracy increase ranged from 5.25% to 14.71%, the top 1

Model	Accuracy			
	Top-1	Top-1 weighted average	Top-3	Top-5
VGG16	26.00%	29.56%	68.48%	90.80%
ResNet50	23.02%	26.35%	65.45%	89.71%
Xception	16.54%	20.22%	54.83%	80.11%
MHATTN-VGG16	31.25%	32.04%	79.94%	96.14%

TABLE 6.4: Experiment result for AUD/USD

Model	Accuracy			
	Top-1	Top-1 weighted average	Top-3	Top-5
VGG16	31.17%	35.11%	73.38%	92.40%
ResNet50	33.56%	36.35%	75.11%	93.95%
Xception	18.20%	24.20%	62.14%	81.89%
MHATTN-VGG16	35.33%	37.68%	86.49%	97.01%

TABLE 6.5: Experiment result for USD/JPY

Model	Accuracy			
	Top-1	Top-1 weighted average	Top-3	Top-5
VGG16	35.22%	38.72%	74.30%	91.26%
ResNet50	33.15%	36.73%	70.70%	90.55%
Xception	29.09%	32.82%	73.40%	90.60%
MHATTN-VGG16	37.55%	40.32%	82.40%	95.42%

TABLE 6.6: Experiment result for USD/CHF

weighted average increased from 2.48% to 11.82%, the top 3 accuracy increased from 11.46% to 25.11%, and the top 5 accuracy increased from 5.34% to 16.03%. For USD/JPY, the top 1 accuracy increase ranged from 1.77% to 17.13%, the top 1 weighted average increased from 1.33% to 13.48%, the top 3 accuracy increased from 11.38% to 24.35%, and the top 5 accuracy increased from 3.06% to 15.12%. For USD/CHF, the top 1 accuracy increase ranged from 2.33% to 8.46%, the top 1 weighted average increased from 1.60% to 7.50%, the top 3 accuracy increased from 8.10% to 11.70%, and the top 5 accuracy increased from 4.16% to 4.87%.

6.4 Discussion

This study explores the use of attention mechanisms to classify financial trading images. The MHATTN-VGG16 model, which incorporates a multi-head attention mechanism and a two-dimensional positional encoding, outperformed the baseline models. The experimental results suggest that these mechanisms can effectively enhance the VGG16 architecture and perform better than other pre-trained image models such as ResNet50 and Xception. This practical approach serves as a baseline for future research in financial image classification, inspiring further advancements and innovations.

While the results are promising, there are potential directions for further improvement. For example, fine-tuning hyperparameters could lead to better performance. Additionally, creating a more diverse training dataset by generating more images or synthetic images for underrepresented classes could also be considered.

6.5 Conclusion

This study investigates the implementation of attention mechanisms, specifically multi-head attention, within the VGG16 Convolutional Neural Network (CNN) architecture to classify financial trading images. The novel MHATTN-VGG16 model extended the capabilities of the standard VGG16, thereby improving image classification performance in the context of financial trading.

A significant finding of this study is that integrating attention mechanisms with CNNs can substantially improve performance. It has demonstrated that attention mechanisms, previously proven effective in domains such as machine translation, can also provide advantages in financial trading image classification. This is a valuable contribution to the field, as the application of attention mechanisms in this context has been relatively unexplored.

The work has paved the way for future research directions. For instance, the model's performance could be further enhanced through hyperparameter tuning, generating more diverse training datasets, or creating synthetic images for underrepresented classes.

While this study has shown promising results, it is essential to note that it is the starting point for a more detailed exploration of attention mechanisms in financial trading image classification. Hopefully, this approach will inspire additional research efforts in this direction, leading to further advancements in the field.

CHAPTER 7

TradingBERT: A Pre-trained Model for Financial Trading Prediction

7.1 Introduction

In the world of Forex trading, technical analysis indicators have been a major source of features for predictive modeling [4]. These indicators are mathematical patterns derived from historical data, such as the simple moving average (SMA) [5], exponential moving average (EMA) [6], and relative strength index (RSI) [7]. However, there are some drawbacks to relying solely on these indicators. First, they can potentially use future information, such as the ZigZag indicator [70], to determine a profitable entry or exit point, but this requires approximately 40 future data points to confirm a transition point [1]. Second, the large number of technical indicators that can be generated by changing the window size makes it difficult to reduce noise in the trading data and select useful features for modeling. Third, using a high number of indicators increases the complexity of the model, leading to increased computational cost and longer training time.

With the recent success of pre-trained models, a new research question arises: Can a pre-trained model be practically applied to forex trading? The development of such a model could provide a reliable source of features without the risk of leaking future information, thereby saving computational resources and time.

This study introduces TradingBERT, a novel application of the BERT model in the realm of financial trading. The pre-trained model, specifically tailored for financial markets, pioneers the use of transfer learning to enhance the predictive capabilities of future price trends and currency movements. As the first of its kind, TradingBERT aims to leverage the rich contextual information present in sequential forex trading data and offer improved predictions. In essence, it seeks to answer a pertinent question: Can the success of BERT, a model initially developed for language data, be effectively transferred to a numerical sequential data landscape, such as forex trading?

This chapter will delve into the methodology, results, discussion, and conclusion of the research on TradingBERT.

7.2 Methods

This study seeks to exploit the powerful capabilities of transformer architectures, self-supervised learning, and masked language modeling to deliver enhanced forex trading predictions.

7.2.1 Masked Language Modeling

At the core of this approach is Masked Language Modeling (MLM). Leveraged effectively in both Natural Language Processing (NLP) and computer vision, this method involves masking parts of a sentence (or image) and having the model predict the masked words (or image patches) based on the surrounding context. This technique promotes the understanding of bidirectional dependencies and the statistical properties of sequences, a feature integral to this study.

7.2.2 Self-Supervised Learning

Self-Supervised Learning (SSL) complements the MLM strategy by transforming unsupervised tasks into supervised ones through automatic label generation. The SSL approach allows the model to predict part of an input based on another part, eradicating the necessity for human-annotated data. This technique has found wide-ranging success in NLP and computer vision tasks by learning useful representations of unlabeled data, which subsequently boosts supervised downstream tasks with limited labeled data through fine-tuning.

7.2.3 Candlestick Embedding and Self-learning Labels

Standardizing price data for forex trading relies on transforming open, high, low, and close (OHLC) prices into six specific elements, collectively forming a candlestick. This allows the construction of four separate vocabularies, each serving a unique purpose in terms of length, direction, starting position, and the entire body of the candlestick.

Two self-learning labeling approaches are used to enhance learning: masked tokens and the IsNext label. Masked tokens hide 15% of the candlesticks along with all six elements, following the masking procedure of the original BERT model. The IsNext label, on the other hand, predicts the next segment in a dataset composed of paired sequences of data.

7.2.4 Transformer

The transformer architecture provides the foundation for this study. An attention-based model facilitates the conversion of one type of sequential data into another. Consisting of an Encoder and a Decoder, the transformer architecture processes input sequences and generates high-dimensional vectors (the Encoder's output) that contain the input's context information. The output is then fed to the Decoder to generate the output sequence. Transformers have demonstrated significant success in NLP and computer vision applications, making it a fitting choice for the forex trading model.

In this study, the encoder of the transformer model, consisting of multiple layers of residual attention blocks, is used to process the input sequence. To predict the actual values of the masked parts in the input sequence, the encoder output is feed into seven dense layers, each predicting a different element within the missing candlestick and the "IsNext" label in the paired sequence.

To evaluate the effectiveness of the pre-trained model, TradingBERT is fine-tuned to aid three downstream tasks: one classification task and two regression tasks. The downstream tasks are labeled in three ways to identify trading opportunities: the trading action label, which represents the optimal trading action in the next windows; the scaled maximum price in the subsequent windows; and the scaled minimum price within the subsequent windows. This comprehensive labeling allows the devise effective trading strategies and robustly evaluate the proposed methodology.

7.2.5 Upstream Data

In this study, TradingBERT was trained using Forex trading data from 2005-01-01 to 2019-12-31, which was obtained from the Oanda API with 15-minute OHLC prices. With only 1.12% of missing data, no techniques were used to handle it and the missing values were ignored.

The training data consisted of paired sequences A and B, with 50% of the data representing a true next segment sequence B for sequence A, while the other 50% was randomly generated as a fake next segment sequence B for sequence A. To capture different contextual information, the total length of sequences A and B were generated as 20, 30, 40, 60, and 80, with both sequences A and B having equal lengths. Approximately 3.7 million input sequences were generated for each currency pair of EUR/GBP and AUD/USD, and two TradingBERT models were trained for each currency pair. The number of input records for the TradingBERT EUR/GBP is shown in Table 7.1, and the input sequence information for the TradingBERT AUD/USD can be found in Table 7.2.

7.2.6 Downstream Data Labeling

The downstream tasks are labeled by the below three methods to identify trading opportunities:

TABLE 7.1: TradingBERT input data - EUR/GBP

Sequence length	Number of records	IsNext segment label	Masking percentage
20	369,585	True	15%
20	369,585	False	15%
30	369,580	True	15%
30	369,580	False	15%
40	369,575	True	15%
40	369,575	False	15%
60	369,565	True	15%
60	369,565	False	15%
80	369,555	True	15%
80	369,555	False	15%

TABLE 7.2: TradingBERT input data - AUD/USD

Sequence length	Number of records	IsNext segment label	Masking percentage
20	369,133	True	15%
20	369,133	False	15%
30	369,133	True	15%
30	369,133	False	15%
40	369,133	True	15%
40	369,133	False	15%
60	369,103	True	15%
60	369,103	False	15%
80	369,093	True	15%
80	369,093	False	15%

(1) Trading Action Label (l^n_{action}): This represents the optimal trading action in the next n windows, with possible values being "buy", "hold", or "sell". This label is calculated by using the pre-defined Take Profit (PT) and Stop Loss (SL) values in forex trading, with the following equation:

$$l^n_{\text{action}} = \begin{cases} 0, & \text{if} \quad P^n_{\max} - P_c \geq PT \quad \text{and} \quad P_c - P^n_{\min} < SL \\ 0, & \text{if} \quad P^n_{\max} - P_c \geq PT \quad \text{and} \quad P_c - P^n_{\min} \geq SL \quad \text{and} \quad IP^n_{\max} < IP^n_{\min} \\ 1, & \text{if} \quad P^n_{\max} - P_c < SL \quad \text{and} \quad P_c - P^n_{\min} \geq PT \\ 1, & \text{if } P^n_{\max} - P_c \geq SL \quad \text{and} \quad P_c - P^n_{\min} \geq PT \quad \text{and} \quad IP^n_{\min} < IP^n_{\max} \\ 2, & \text{otherwise} \end{cases} \tag{7.1}$$

Here, l^n_{action}=0 denotes "buy", l^n_{action}=1 means "sell", and l^n_{action}=2 implies "hold".

(2) Scaled Maximum Price (Ps^n_{max}): This is the predicted maximum price in the next n windows, scaled according to the minimum and maximum prices within the input sequence. This prediction helps develop an effective trading strategy. The equation for Ps^n_{max} is:

$$Ps^n_{\max} = \frac{P^n_{\max} - Pt^m_{\min}}{Pt^m_{\max} - Pt^m_{\min}} \tag{7.2}$$

In this equation, P^n_{max} denotes the highest price within the future n windows, whereas Pt^m_{max} and Pt^m_{min} signify the minimum and maximum prices respectively, observed in the previous m windows.

(3) Scaled Minimum Price (Ps^n_{min}): The third prediction goal, Ps^n_{min}, reflects the estimated minimum price within the next n windows, scaled according to the minimum and maximum prices in the input sequence. The equation to calculate Ps^n_{min} is:

$$Ps^n_{\min} = \frac{P^n_{\min} - Pt^m_{\min}}{Pt^m_{\max} - Pt^m_{\min}} \tag{7.3}$$

Where P^n_{min} represents the minimum price in the next n windows, and Pt^m_{min} and Pt^m_{max} represent the minimum and maximum prices in the preceding m windows, respectively.

Data from 01-01-2020 to 30-06-2022 is leveraged to perform a series of downstream tasks. The data was segmented into training, validation, and test sets. The analyses focused on forecasting three targets within every ten-time steps: l^{10}_{action}, P^{10}_{max}, and P^{10}_{min}. Details of the downstream data can be seen in 7.3, and 7.4 presents class information for l^{10}_{action}.To enhance the robustness of the evaluation, TradingBERT is employed, a model explicitly trained for currency pair analysis, on EUR/GBP and AUD/USD, respectively.

TABLE 7.3: Raw dataset for downstream tasks

	Number of rows	Number of missing values	Percentage of missing values
AUD/USD	62,069	490	0.79%
EUR/GBP	62,114	435	0.70%

TABLE 7.4: Class information for l^{10}_{action}

Label of classes	AUD/USD			EUR/GBP		
	Train	Validation	Test	Train	Validation	Test
0 (buy)	16,873	2,328	2,305	12,237	1,406	1,629
1 (sell)	16,925	2,188	2,422	12,294	1,360	1,614
2 (hold)	16,072	1,639	1,307	25,368	3,390	2,806

7.3 Results

Evaluation of model performance for l^{10}_{action} was conducted using accuracy, precision, recall, and F1 score, both macro and weighted average. Evaluation for P^{10}_{max} and P^{10}_{min} employed metrics like MSE, RMSE, MAE, and MAPE.

7.3.1 Prediction of l^{10}_{action}

Results are presented in table 7.6 and table 7.7. In summary, the six models augmented by TradingBERT all outperform all of the baseline models for both EUR/GBP and AUD/USD currency pairs. All six TradingBERT-enhanced models for EUR/GBP exhibit similar performance in terms of accuracy, macro average F1 score, and weighted average F1 score. This suggests that TradingBERT can effectively extract features from candlestick data, and the addition of price features and technical analysis indicators does not greatly impact model performance. This pattern holds true for AUD/USD as well, where using only candlestick elements provides similar results to incorporating price data and technical analysis indicators. The baseline model relying solely on candlestick elements performed the poorest, indicating that TradingBERT is crucial in making sense of this type of data and generating effective trading signals. The TradingBERT feature extraction model for EUR/GBP achieved a 53.43% accuracy, a 6.17% increase over the best-performing baseline model with technical indicators only. Similarly, the TradingBERT feature extraction model for AUD/USD had a 4.85% advantage over the best-performing baseline model, which combined price data, technical analysis indicators, and candlestick elements.

7.3.2 Prediction of P^{10}_{max} and P^{10}_{min}

The results to predict P^{10}_{max} and P^{10}_{min} are displayed table 7.8 and table 7.9. Among the six models that incorporate TradingBERT, the model that uses candlestick element data performed the best for both currency pairs, and the addition of price features and technical analysis indicators did not result in a significant improvement in performance. The best-performing models with TradingBERT outperformed the baseline models that used only candlestick elements, leading to a decrease of 0.13x10^{-3} in RMSE for EUR/GBP and 0.36x10^{-3} in RMSE for AUD/USD. This further supports the effectiveness of TradingBERT in extracting features from candlesticks.

However, the best-performing models with TradingBERT still underperformed compared to the baseline models that included price data and technical analysis indicators. For EUR/GBP, the best-performing baseline model combined price data, technical analysis indicators, and candlestick elements, achieving an RMSE of 1.03x10^{-3}, which is a decrease of 0.10x10^{-3} compared to the TradingBERT fine-tuning model using only candlestick elements. For AUD/USD, the best-performing baseline model used only price data and achieved an RMSE of $1.31\text{x}10^{-3}$, which is a decrease of $0.12\text{x}10^{-3}$ compared to the TradingBERT feature extraction model with only candlestick elements.

TABLE 7.5: Details of the model and feature sets

Features	Baseline	TradingBERT	
		Fine-tuning	Feature extraction
Candlestick elements only	1. add 7 candlestick elements embeddings with positional embedding; 2. an LSTM layer with 128 units for the added embeddings; 3. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 4. three dense layers to predict the three targets.	1. update the parameters of the TradingBERT in the last layer; 2. output of the TradingBERT goes through a normalization layer, a regularization layer, and a dropout layer with a rate of 0.1; 3. three dense layers to predict the three targets.	1. pass the candlestick elements to the TradingBERT to generate features, without updating the parameters; 2. output of the TradingBERT goes through a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 3. three dense layers to predict the three targets.
Price data + TA + Candlestick	1. an LSTM layer with 128 units for the TA indicators; 2. an LSTM layer with 128 units for the price data; 3. add 7 candlestick elements embeddings with positional embedding; 4. an LSTM layer with 128 units for the added embeddings; 5. concatenate the above 3 LSTM outputs; 6. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 7. three dense layers to predict the three targets.	1. an LSTM layer with 128 units for the TA indicators; 2. an LSTM layer with 128 units for the price data; 3. update the parameters of the TradingBERT in the last layer; 4. concatenate the above 2 LSTM and the TradingBERT outputs; 5. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 6. three dense layers to predict the three targets.	1. an LSTM layer with 128 units for the TA indicators; 2. an LSTM layer with 128 units for the price data; 3. extract features of the candlestick from the TradingBERT, without updating the parameters; 4. concatenate the above 2 LSTM and the TradingBERT outputs; 5. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 6. three dense layers to predict the three targets.
Price data + Candlestick	1. an LSTM layer with 128 units for the price data; 2. add 7 candlestick elements embeddings with positional embedding; 3. an LSTM layer with 128 units for the added embeddings; 4. concatenate the above 3 LSTM outputs; 5. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 6. three dense layers to predict the three targets.	1. an LSTM layer with 128 units for the price data; 2. update the parameters of the TradingBERT in the last layer; 3. concatenate the above LSTM and the TradingBERT outputs; 4. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 5. three dense layers to predict the three targets.	1. an LSTM layer with 128 units for the price data; 2. extract features of the candlestick from the TradingBERT, without updating the parameters 3. concatenate the above LSTM and the TradingBERT outputs; 4. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 5. three dense layers to predict the three targets.
Price data only	1. an LSTM layer with 128 units for the price data; 2. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 3. three dense layers to predict the three targets.		
TA indicators only	1. an LSTM layer with 128 units for the TA indicators; 2. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 3. three dense layers to predict the three targets.		
Price data + TA	1. an LSTM layer with 128 units for the TA indicators; 2. an LSTM layer with 128 units for the price data; 3. concatenate the above 2 LSTM outputs; 4. a normalization layer, a regularization layer, a dropout layer with a rate of 0.1; 5. three dense layers to predict the three targets.		

TABLE 7.6: Result for l_{action}^{10} - EUR/GBP

Features	Label class	Label support	Baseline				TradingBERT							
							Fine-tuning				Feature extraction			
			Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
	0 (buy)	1,629		0.00%	0.00%	0.00%		40.21%	23.57%	29.72%		40.15%	26.15%	31.67%
	1 (sell)	1,614		0.00%	0.00%	0.00%		40.39%	23.30%	29.55%		40.72%	24.47%	30.57%
Candlestick elements only	2 (hold)	2,806	46.39%	46.39%	100.00%	63.38%	53.17%	59.00%	87.53%	70.48%	53.43%	60.00%	85.92%	70.66%
	macro avg			15.46%	33.33%	21.13%		46.53%	44.80%	43.25%		46.96%	45.52%	**44.30%**
	weighted avg			21.52%	46.39%	29.40%		48.97%	53.17%	48.58%		49.51%	53.43%	**49.47%**
	0 (buy)	1,629		38.32%	2.52%	4.72%		44.19%	16.33%	23.85%		40.74%	27.13%	32.57%
	1 (sell)	1,614		40.00%	0.37%	0.74%		41.60%	30.05%	34.89%		41.81%	18.03%	25.19%
Price data + Candlestick	2 (hold)	2,806	46.70%	46.87%	99.00%	63.62%	**53.73%**	58.37%	89.06%	70.52%	53.27%	58.32%	88.70%	70.37%
	macro avg			41.73%	33.96%	23.03%		48.05%	45.15%	43.09%		46.96%	44.62%	42.71%
	weighted avg			42.73%	46.70%	30.98%		50.08%	53.73%	48.45%		49.18%	53.27%	48.14%
	0 (buy)	1,629		35.78%	6.88%	11.53%		40.69%	21.73%	28.33%		42.08%	17.13%	24.35%
	1 (sell)	1,614		32.58%	4.46%	7.85%		41.84%	23.05%	29.72%		41.92%	29.55%	34.67%
Price data + TA + Candlestick	2 (hold)	2,806	47.13%	48.36%	95.05%	64.10%	53.23%	58.14%	88.88%	70.29%	53.53%	58.43%	88.45%	70.37%
	macro avg			38.91%	35.46%	27.83%		46.89%	44.55%	42.78%		47.47%	45.04%	43.13%
	weighted avg			40.76%	47.13%	34.94%		49.09%	53.23%	48.17%		49.62%	53.53%	48.45%
	0 (buy)	1,629		0.00%	0.00%	0.00%								
	1 (sell)	1,614		100.00%	60.00%	12.00%								
Price data only	2 (hold)	2,806	46.40%	46.41%	100.00%	63.40%								
	macro avg			48.80%	33.35%	21.17%								
	weighted avg			48.21%	46.40%	29.44%								
	0 (buy)	1,629		32.20%	8.10%	12.95%								
	1 (sell)	1,614		34.09%	6.51%	10.93%								
TA only	2 (hold)	2,806	47.26%	49.18%	93.44%	64.45%								
	macro avg			38.49%	36.02%	29.44%								
	weighted avg			40.58%	47.26%	36.30%								
	0 (buy)	1,629		36.20%	6.20%	10.59%								
	1 (sell)	1,614		32.28%	6.32%	10.57%								
Price data + TA	2 (hold)	2,806	47.25%	48.68%	94.62%	64.29%								
	macro avg			39.05%	35.71%	28.48%								
	weighted avg			40.94%	47.25%	35.49%								

TABLE 7.7: Result for l_{action}^{10} - AUD/USD

Features	Label class	Label support	Baseline				TradingBERT							
							Fine-tuning				Feature extraction			
			Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Candlestick elements only	0 (buy)	2,305	21.66%	0.00%	0.00%	0.00%	42.08%	40.97%	58.05%	48.03%	41.71%	40.69%	65.51%	50.20%
	1 (sell)	2,422		0.00%	0.00%	0.00%		43.48%	34.85%	38.69%		45.00%	20.27%	27.95%
	2 (hold)	1,307		21.66%	100.00%	35.61%		43.17%	27.31%	33.46%		41.88%	39.48%	40.65%
	macro avg			7.22%	33.33%	11.87%		42.54%	40.07%	40.06%		42.53%	41.75%	39.60%
	weighted avg			4.69%	21.66%	7.71%		42.45%	42.08%	**41.13%**		42.68%	41.71%	39.20%
Price data + Candlestick	0 (buy)	2,305	35.27%	37.51%	35.18%	36.31%	**42.09%**	38.41%	32.71%	35.33%	41.88%	41.91%	56.96%	48.29%
	1 (sell)	2,422		41.69%	34.81%	37.94%		43.66%	59.54%	50.38%		47.24%	24.77%	32.50%
	2 (hold)	1,307		25.62%	36.27%	30.03%		47.79%	26.32%	33.16%		37.65%	46.98%	41.80%
	macro avg			34.94%	35.42%	34.76%		42.29%	39.52%	39.62%		42.27%	42.90%	**40.86%**
	weighted avg			36.61%	35.27%	35.60%		41.90%	42.09%	40.90%		43.13%	41.88%	40.55%
Price data + TA + Candlestick	0 (buy)	2,305	36.86%	38.37%	44.60%	41.25%	41.22%	41.22%	68.94%	51.59%	41.08%	40.96%	62.21%	49.40%
	1 (sell)	2,422		40.70%	33.69%	36.86%		45.33%	12.22%	19.25%		44.97%	19.74%	27.43%
	2 (hold)	1,307		28.15%	29.07%	28.60%		39.45%	46.06%	42.50%		38.57%	43.38%	40.84%
	macro avg			35.74%	35.79%	35.57%		42.00%	42.41%	37.78%		41.50%	41.78%	39.22%
	weighted avg			37.09%	36.86%	36.75%		42.49%	41.22%	36.64%		42.05%	41.08%	38.73%
Price data only	0 (buy)	2,305	35.81%	38.44%	52.36%	44.33%								
	1 (sell)	2,422		40.89%	21.14%	27.87%								
	2 (hold)	1,307		26.92%	33.82%	29.98%								
	macro avg			35.42%	35.77%	34.06%								
	weighted avg			36.93%	35.81%	34.62%								
TA only	0 (buy)	2,305	36.71%	39.02%	44.03%	41.38%								
	1 (sell)	2,422		41.17%	36.09%	38.46%								
	2 (hold)	1,307		24.89%	24.94%	24.91%								
	macro avg			35.03%	35.02%	34.92%								
	weighted avg			36.82%	36.71%	36.64%								
Price data + TA	0 (buy)	2,305	36.46%	39.26%	42.34%	40.74%								
	1 (sell)	2,422		39.64%	36.87%	38.20%								
	2 (hold)	1,307		25.56%	25.33%	25.44%								
	macro avg			34.82%	34.85%	34.80%								
	weighted avg			36.44%	36.46%	36.41%								

TABLE 7.8: Result for P^{10}_{max} and P^{10}_{min} - EUR/GBP

Features	**Baseline for max price prediction**				**TradingBERT for max price prediction**							
					Fine-tuning				**Feature extraction**			
	MSE(10^-6)	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**
Candlestick elements only	1.58	1.26	8.16	9.60	1.27	1.13	7.00	8.23	1.31	1.15	7.00	8.23
Price data + Candlestick	1.08	1.04	6.47	7.61	1.35	1.16	7.17	8.44	1.36	1.17	7.07	8.31
Price data + TA + Candlestick	1.08	1.03	6.50	7.65	1.33	1.15	7.09	8.34	1.30	1.14	6.95	8.17
Price data only	1.08	1.04	6.48	7.62								
TA only	1.10	1.05	6.70	7.88								
Price data + TA	1.09	1.04	6.58	7.74								

Features	**Baseline for min price prediction**				**TradingBERT for min price prediction**							
					Fine-tuning				**Feature extraction**			
	MSE(10^-6)	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**
Candlestick elements only	1.88	1.37	8.39	9.90	1.34	1.16	6.67	7.87	1.35	1.16	6.68	7.88
Price data + Candlestick	1.00	1.00	6.48	7.64	1.36	1.17	7.01	8.28	1.34	1.16	6.71	7.92
Price data + TA + Candlestick	1.02	1.01	6.53	7.71	1.42	1.19	6.91	8.15	1.36	1.17	6.72	7.93
Price data only	1.07	1.03	6.68	7.89								
TA only	1.19	1.09	8.15	6.91								
Price data + TA	1.06	1.03	6.66	7.86								

TABLE 7.9: Result for P_{max}^{10} and P_{min}^{10} - AUD/USD

Features	**Baseline for max price prediction**				**TradingBERT for max price prediction**							
					Fine-tuning				**Feature extraction**			
	MSE(10^-6)	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**
Candlestick elements only	3.19	1.79	12.62	17.67	2.43	1.56	10.44	14.59	2.04	1.43	9.77	13.68
Price data + Candlestick	1.73	1.32	9.07	12.69	2.67	1.64	10.92	15.27	2.24	1.50	10.58	14.82
Price data + TA + Candlestick	1.71	1.31	9.00	12.59	2.14	1.46	9.77	13.66	2.15	1.47	10.17	14.24
Price data only	1.72	1.31	9.08	12.71								
TA only	1.81	1.35	9.23	12.90								
Price data + TA	1.73	1.32	9.02	12.62								
Features	**Baseline for min price prediction**				**TradingBERT for min price prediction**							
					Fine-tuning				**Feature extraction**			
	MSE(10^-6)	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**	**MSE(10^-6)**	**RMSE(10^-3)**	**MAE(10^-4)**	**MAPE(%)**
Candlestick elements only	3.19	1.79	12.64	17.77	3.13	1.77	12.11	17.06	2.41	1.55	1.08	15.24
Price data + Candlestick	1.96	1.40	10.15	14.28	3.75	1.94	13.37	18.85	2.89	1.70	11.84	16.68
Price data + TA + Candlestick	1.81	1.35	9.77	1.38	2.56	1.60	11.02	15.50	2.74	1.66	11.51	16.22
Price data only	1.96	1.40	10.10	14.21								
TA only	1.92	1.38	10.17	14.31								
Price data + TA	1.85	1.36	9.88	13.90								

7.4 Discussion

Through this research, using a pre-trained model can be beneficial in predicting financial market trends. The results have been promising is discovered, with TradingBERT effectively extracting and utilizing relevant information from candlestick data. It has performed as well as, and sometimes even better than, traditional data processing methods such as price data analysis and technical indicators. While there is still room for improvement, this research has identified areas where modifications could enhance the model's performance.

Interestingly, exploring different masking techniques holds promise for enhancing TradingBERT's efficacy. Current masking strategies are founded on empirical evidence, with a 15% masking rate. However, other studies have demonstrated the effectiveness of higher masking rates and the application of alternative masking techniques. For instance, different BERT variants have introduced dynamic and entity-centric masking strategies. Further experimentation on the optimal masking rate and strategy for financial trading data could result in significant performance gains for TradingBERT.

In addition, the scope of sequential data used for training TradingBERT was restricted due to computational constraints. Expanding the sequential length of training data and the maximum length could lead to superior outcomes. It should also be noted that TradingBERT performed better with classification tasks compared to regression models. It might be that the categorical transformation of numerical data favors classification tasks, thus providing a potential avenue for improving TradingBERT's application on regression tasks.

7.5 Conclusion

In summary, TradingBERT has shown remarkable promise in extracting and interpreting meaningful signals from trading data. Despite computational requirements and training time limitations, the benefits of implementing a pre-trained model in the financial trading domain are evident. TradingBERT's performance on classification tasks, in particular, demonstrates its utility in providing robust and informative features for trading time series data.

The transformation techniques implemented in this research, including the normalization of price data and the generation of candlestick elements, render TradingBERT a viable and practical solution for financial data analysis and prediction. Moreover, my study reveals a new perspective for transmuting regression problems into classification ones, expanding the possible applications of pre-trained models in financial trading.

For further exploration, research, and application of TradingBERT, the model and code are publicly available in my GitHub repository at https://github.com/liqi6811/TradingBERT. Additional research and experimentation are invited in this field to further harness the potential of pre-trained models in financial trading and beyond.

CHAPTER 8

Discussion

Across the three studies presented in this thesis, the potential of deep learning models in forex trading signal extraction has been thoroughly explored.

Chapter 5 introduces an Event-Driven LSTM model that uses significant events in the financial market as input. This event-driven approach offers a fresh perspective for modeling forex markets, especially given the volatility and unpredictability of these markets. However, this approach also reveals limitations, such as not fully considering the intricate network of dependencies that exist among different financial markets. This limitation can lead to an incomplete understanding of market dynamics.

Chapter 6 builds upon the first paper by presenting the MHATTN-VGG16 model, a novel approach that combines the VGG16 model with an attention mechanism. By modeling financial prices as an image, MHATTN-VGG16 effectively captures the dependencies and interrelationships among different markets, leading to improved prediction performance. However, the model's effectiveness is linked with appropriate hyperparameter tuning and sufficient quality training data, the lack of which can affect its performance. Additionally, integrating synthetic images for underrepresented classes, while beneficial, can introduce complexity. Also, the novel use of attention and CNN mechanisms for financial time series images might not be apt for all scenarios. Finally, the model's complexity could increase training time and computational requirements. Despite these challenges, the study provides a promising base for future exploration in this field.

Chapter 7 goes a step further, introducing TradingBERT, a pre-trained model based on BERT that is specifically designed for trading data. It demonstrates that a pre-trained model can effectively extract valuable information from trading data and significantly improve the performance of downstream tasks. While showcasing promising results, the third study's TradingBERT model requires substantial computational resources and extended training time. This could potentially limit its adoption in resource-constrained environments. Moreover, it tends to perform better on classification tasks than regression tasks, suggesting further improvements to enhance its versatility.

A comparative analysis of the models reveals the progression of the research and the increasingly sophisticated techniques used to extract trading signals from forex data. Each model represents a unique approach, with its strengths and weaknesses, and each of these limitations, however, paves the way for future enhancements and provides direction for the next steps in this line of research.

CHAPTER 9

Conclusion and Future Work

The series of studies presented in this thesis demonstrate the evolving capabilities of deep learning models in extracting trading signals from forex data. Starting from the Event-Driven LSTM model, moving on to the MHATTN-VGG16, and culminating with TradingBERT, each step of the research journey highlights innovative approaches and solutions to the challenges of forex trading.

In the first study, future iterations of the model could strive to incorporate additional event types and refine the event detection mechanism. The model could also be further improved to better handle noise in the data and unexpected market events. These would provide a more comprehensive understanding of market dynamics and potentially improve the model's predictive capabilities.

For the second study, future work could concentrate on refining the MHATTN-VGG16 model's hyperparameter tuning process, possibly by implementing automated optimization strategies to balance performance and computational efficiency. Additionally, future work could explore the use of synthetic minority classes to address the issue of class imbalance. The model can achieve better accuracy and performance across all classes by generating synthetic samples for underrepresented classes, leading to more robust predictions. Another aspect of future research might involve broadening the applicability of attention and CNN mechanisms within various financial time series scenarios. Further enhancements could also explore simplifying the model's complexity without sacrificing its performance, thus making it more accessible and easier to interpret.

For the third study, future work could focus on improving the computational efficiency of the TradingBERT model and devising methods to reduce training time without compromising the model's performance. Further enhancements could also focus on improving its performance on regression tasks, perhaps by integrating regression-oriented pre-training strategies.

Additionally, exploring different masking strategies and techniques used in other BERT variants could provide an exciting avenue for future research. A potential area of improvement is the exploration of different data transformation strategies, which could enhance the versatility and applicability of TradingBERT to various types of trading data.

Overall, while offering significant insights and advancements, each of the three studies opens up numerous avenues for future research and development, contributing to advancing deep learning based approaches in forex trading signal extraction. In addition, as advancements in deep learning continue to emerge, other promising models or strategies could be integrated into the current models or serve as bases for new models. These possibilities could help to push the boundaries of what can be achieved in forex trading signal extraction.

In conclusion, this thesis represents a substantial contribution to the field of forex trading, providing valuable insights and tools that have the potential to enhance trading strategies and outcomes significantly. The possibility of deep learning models in this field is vast, and the journey to fully harness this potential is ongoing.

References

[1] L Qi, M Khushi and J Poon. 'Event-driven LSTM for forex price prediction'. In: *Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE): 16-18 December 2020; Virtual*. 2020, pp. 1–6.

[2] B. Graham and D. Dodd. 'Security Analysis: Sixth Edition, Foreword by Warren Buffett'. In: McGraw Hill Professional, 2009.

[3] J.J. Murphy. 'Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications'. In: New York Institute of Finance, 1999.

[4] J J Murphy. 'Philosophy of Technical Analysis'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, pp. 1–21.

[5] J J Murphy. 'Moving Averages'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, p. 199.

[6] J J Murphy. 'Moving Averages'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, p. 199.

[7] J J Murphy. 'The Relative Strength Index (RSI)'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, p. 239.

[8] E. F. Fama. 'The Behavior of Stock-Market Prices'. In: *The Journal of Business* 38(1) (1965).

[9] B. G. Malkiel. ‘A Random Walk Down Wall Street: The Time-Tested Strategy for Successful Investing’. In: *W. W. Norton & Company* (1973).

[10] E. F. Fama. ‘Efficient Capital Markets: A Review of Theory and Empirical Work’. In: *The Journal of Finance* 25(2) (1970).

[11] P. A. Samuelson. ‘Proof That Properly Anticipated Prices Fluctuate Randomly’. In: *Industrial Management Review* 6 (1965).

[12] B. G. Malkiel. ‘The Efficient Market Hypothesis and Its Critics’. In: *Journal of Economic Perspectives* 17(1) (2003).

[13] O Sezer, M Gudelek and A Ozbayoglu. ‘Financial time series forecasting with deep learning: A systematic literature review: 2005-2019’. In: *Applied Soft Computing Journal* 90 (2020).

[14] J. Wu et al. ‘Quantitative trading on stock market based on deep reinforcement learning’. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2019: 14-19 July, 2019; Budapest, Hungary*. 2019, pp. 1–8.

[15] T. Kim and H.Y. Kim. ‘Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data’. In: *PloS One* 14(2) (2019).

[16] SJ Guo, FC Hsu and CC Hung. ‘Deep candlestick predictor: a framework towards forecasting the price movement from candlestick charts’. In: *9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2018: 26-28 December, 2018; Taipei, Taiwan*. 2018.

[17] J.F. Chen et al. ‘Financial time-series data analysis using deep convolutional neural networks’. In: *7th International Conference on Cloud Computing and Big Data, 2016: 16-18 November, 2016; Macau, China*. 2016.

[18] K. Jearanaitanakij and B. Passaya. ‘Predicting short trend of stocks by using convolutional neural network and candlestick patterns’. In: *4th International Conference on Information Technology (InCIT), 2019: 24-25 October, 2019; Bangkok, Thailand*. 2019.

[19] L Yang et al. ‘Html: Hierarchical transformer-based multi-task learning for volatility prediction’. In: *Proceedings of the Web Conference 2020: 20-24 April 2020; Taipei, China*. 2020, pp. 441–451.

[20] J Liu et al. ‘Transformer-based capsule network for stock movement prediction’. In: *Proceedings of the First Workshop on Financial Technology and Natural Language Processing: 10 August 2019; Macao, China*. 2019, pp. 66–73.

[21] W. Bao, J. Yue and Y. Rao. ‘A deep learning framework for financial time series using stacked autoencoders and long-short term memory’. In: *PloS One* 12(7) (2017).

[22] J. Bollen, H. Mao and X. Zeng. ‘Twitter mood predicts the stock market’. In: *Journal of Computational Science* 2.1 (2011), pp. 1–8.

[23] T. Loughran and B. McDonald. ‘When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks’. In: *The Journal of Finance* 66.1 (2011), pp. 35–65.

[24] M. Hagenau, M. Liebmann and D. Neumann. ‘Automated news reading: Stock price prediction based on financial news using context-capturing features’. In: *Decision Support Systems* 55.3 (2013), pp. 685–697.

[25] A. W. Lo, H. Mamaysky and J. Wang. ‘Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation’. In: *The Journal of Finance* 55.4 (2000), pp. 1705–1770.

[26] W. Brock, J. Lakonishok and B. LeBaron. ‘Simple technical trading rules and the stochastic properties of stock returns’. In: *The Journal of Finance* 47.5 (1992), pp. 1731–1764.

[27] M. Ballings et al. ‘Evaluating multiple classifiers for stock price direction prediction’. In: *Expert Systems with Applications* 42.20 (2015), pp. 7046–7056.

[28] S. Baek and D. Cho. ‘Machine learning-based anomaly detection via integration of manufacturing, inspection, and after-sales service data for the automotive industry’. In: *Journal of Manufacturing Systems* 54 (2020), pp. 138–150.

[29] M. Zhu and L. Wang. 'Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms'. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. 2010, pp. 1–5.

[30] A. Dingli and K. S. Fournier. 'Financial time series forecasting - a deep learning approach'. In: *International Journal of Machine Learning and Computing* 7.5 (2017).

[31] A. Dingli and K. S. Fournier. 'Financial time series forecasting - a machine learning approach'. In: *International Journal of Machine Learning and Computing* 14.1/2/3 (2017).

[32] R. M. I. Kusuma et al. 'Using deep learning neural networks and candlestick chart representation to predict stock market'. In: *arXiv preprint arXiv:1903.12258v1* (2019).

[33] G. Hu et al. 'Deep stock representation learning: from candlestick charts to investment decisions'. In: *arXiv preprint arXiv:1709.03803v3* (2018).

[34] O. B. Sezer and A. M. Ozbayoglu. 'Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach'. In: *Applied Soft Computing* 70 (2018), pp. 525–538.

[35] H. S. Sim et al. 'Is deep learning for image recognition applicable to stock market prediction'. In: *Complexity* 2019 (Feb. 2019), p. 4324878.

[36] Y. C. Tsai et al. 'Predict forex trend via convolutional neural networks'. In: *Journal of Intelligent Systems* 29.1 (Sept. 2018), pp. 941–958.

[37] Y. Y. Chen et al. 'Developing arbitrage strategy in high-frequency paris trading with filter bank CNN algorithm'. In: *2018 IEEE International Conference on Agents (ICA)*. 2018.

[38] A. Siripurapu. 'Convolutional networks for stock trading'. In: (2014). Standford University course project report.

[39] Z. Yang et al. 'Prediction of stock trading signal based on multiindicator channel convolutional neural networks'. In: *2019 IEEE 8th Data Driven Control and Learning Systems Conference*. May 2019.

[40] M. U. Gudelek et al. 'A deep learning based stock trading model with 2-D CNN trend detection'. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. Nov. 2017.

[41] A Vaswani et al. 'Attention is all you need'. In: *Advances in neural information processing systems, 30, 2017: 4-9 December 2017; Long Beach, CA, USA*. 2017.

[42] J Devlin et al. 'BERT: Pre-training of deep bidirectional transformers for language understanding'. In: *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): 3-5 June 2019; Minneapolis, MN, USA*. 2019.

[43] A Radford et al. *Improving Language Understanding by Generative Pre-Training*. 2018. URL: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (visited on 15/08/2022).

[44] Z Lan et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2019. URL: https://arxiv.org/abs/1909.11942 (visited on 15/08/2022).

[45] Y Liu et al. *Roberta: A robustly optimized bert pretraining approach*. 2019. URL: https://arxiv.org/abs/1907.11692 (visited on 15/08/2022).

[46] Y Sun et al. *Ernie: Enhanced representation through knowledge integration*. 2019. URL: https://arxiv.org/abs/1904.09223 (visited on 15/08/2022).

[47] V Sanh et al. *DistilBERT, a distilled version of BERT*. 2019. URL: https://arxiv.org/abs/1910.01108 (visited on 15/08/2022).

[48] I Bello et al. 'Attention augmented convolutional networks'. In: *Proceedings of the IEEE/CVF international conference on computer vision 2019: 27 October - 2 November 2019; Seoul, Korea*. 2019, pp. 3286–3295.

[49] A Vaswani et al. 'Scaling local self-attention for parameter efficient visual backbones'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021: 19-25 June 2021; Virtual*. 2021, pp. 12894–12904.

[50] J Lee et al. ‘Set transformer: A framework for attention-based permutation-invariant neural networks’. In: *International conference on machine learning 2019: 10-15 June 2019; Long Beach, CA, USA*. 2019, pp. 3744–3753.

[51] N Carion et al. ‘End-toend object detection with transformers’. In: *European conference on computer vision 2020: 23-28 August 2020; Virtual*. 2020, pp. 213–229.

[52] Q Guan et al. *Diagnose like a radiologist: attention guided convolutional neural network for thorax disease classification*. 2018. URL: https://arxiv.org/abs/1801.09927 (visited on 15/08/2022).

[53] Bolei Xu et al. ‘Look, investigate, and classify: A deep hybrid attention method for breast cancer classification’. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 914–918.

[54] Aravind Srinivas et al. ‘Bottleneck Transformers for Visual Recognition’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[55] Irwan Bello et al. ‘Attention Augmented Convolutional Networks’. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2020.

[56] Nicolas Carion et al. ‘End-to-End Object Detection with Transformers’. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020.

[57] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv preprint arXiv:2010.11929. 2020.

[58] Meng-Hao Guo et al. *Beyond Self-attention: External Attention using Two Linear Layers for Visual Tasks*. arXiv preprint arXiv:2105.02358. 2021.

[59] B Xu et al. ‘Look, investigate, and classify: A deep hybrid attention method for breast cancer classification’. In: *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019): 8-11 April 2019; Venice, Italy*. 2019, pp. 914–918.

[60] Hongdou Yao et al. ‘Parallel Structure Deep Neural Network Using CNN and RNN with an Attention Mechanism for Breast Cancer Histology Image Classification’. In: *Cancers* 11.12 (2019).

[61] Mazhar Shaikh, Varghese Alex Kollerathu and Ganapathy Krishnamurthi. 'Recurrent Attention Mechanism Networks for Enhanced Classification of Biomedical Images'. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 1260–1264.

[62] K Simonyan and A Zisserman. *Very deep convolutional networks for large-scale image recognition*. 2014. URL: https://arxiv.org/abs/1409.1556 (visited on 15/08/2022).

[63] F Chollet. 'Xception: Deep learning with depthwise separable convolutions'. In: *Proceedings of the IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR): 21-26 July 2017; Honolulu, HI, USA*. 2017.

[64] K He et al. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR): 26 June - 1 July 2016; Las Vegas*. 2016, pp. 770–778.

[65] J Pennington, R Socher and C D Manning. 'Glove: Global vectors for word representation'. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP): 25-29 October 2014; Doha, Qatar*. 2014, pp. 1532–1543.

[66] B Iz, K Lo and A Cohan. 'SciBERT: A Pretrained Language Model for Scientific Text'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019: 3-7 November 2019; Hong Kong, China*. 2019, pp. 3615–3620.

[67] Z Liu Z et al. 'Finbert: A pre-trained financial language representation model for financial text mining'. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI) 2021: 19-26 August, 2021; Montreal-themed Virtual Reality, Canada*. 2021, pp. 4513–4519.

[68] I Chalkidis et al. 'LEGAL-BERT: The Muppets straight out of Law School'. In: *EMNLP (Findings) 2020: 16–18 November, 2020; Virtual Conference*. 2020.

[69] R.N. Elliott. 'R.N. Elliot's Masterworks'. In: (1994).

[70] J Chen. *Zig Zag Indicator*. 2022. URL: https://www.investopedia.com/terms/z/zig%5C_zag%5C_indicator.asp (visited on 15/08/2022).

[71] J J Murphy. 'Moving Average Convergence/Divergence (MACD)'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, pp. 252–255.

[72] J. Welles Wilder. 'New concepts in technical trading systems'. In: *Trend Research* 47 (1980), pp. 40–49.

[73] John Bollinger. 'Bollinger on Bollinger Bands'. In: (2001).

[74] J J Murphy. 'Larry Williams %R'. In: *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. Ed. by E S Coleman. 2nd. New York: New York Institute of Finance, 1999, p. 249.

[75] Ed Ponsi. 'Technical Analysis and Chart Interpretations: A Comprehensive Guide to Understanding Established Trading Tactics for Ultimate Profit'. In: Somerset: John Wiley & Sons, Incorporated, 2016, pp. 182–183.

[76] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. 'Neural machine translation by jointly learning to align and translate'. In: *International Conference on Learning Representations (ICLR)*. 2016.

[77] Zelun Wan and Jyh-Charn Liu. *Translating Math Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-level Training*. arXiv preprint arXiv:1908.11415. 2019.

Abbreviations

ADX	Average Directional Index
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BB	Bollinger Bands
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long-Short Term Memory
CIFAR	Canadian Institute for Advanced Research
CNN	Convolutional Neural Network
DBN	Deep Belief Networks
DMLP	Deep Multilayer Perceptron
DRL	Deep Reinforcement Learning
EMA	Exponential Moving Average
EMH	Efficient-market Hypothesis
ETF	Exchange-Traded Fund
FinBERT	Financial BERT
GCN	Graph Convolutional Network
GDP	Gross Domestic Product
GloVe	Global Vectors for Word Representation
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit

HSI	Hang Seng Stock Index
LegalBERT	Legal BERT
LSTM	Long-Short Term Memory
MACD	Moving Average Convergence Divergence
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MHATTN-VGG16	Multi-head Attention VGG16
ML	Machine Learning
MLM	Masked Language Model
MLP	Multi Layer Perceptron
MLP-GA	Multilayer Perceptron optimized by GA
MSE	Mean Square Error
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OHLC	Open, High, Low, Close
PPO	Percentage Price Oscillator
ResNet	Residual Network
RBM	Restricted Boltzmann Machines
RGB	Red, Green, Blue
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
SciBERT	Science BERT
SC-CNN	Stock Chart Convolutional Neural Network
SSL	Self-Supervised Learning
SMA	Simple Moving Average
ST-LSTM	Stock Time Long-Short Term Memory
SVM	Support Vector Machines

SVRGA	Support Vector Regression optimized by GA
TA-LIB	Technical Analysis Library
TAT	Triangle Area-based Trend
TPU	Tensor Processing Units
TradingBERT	Trading BERT
VGG	Visual Geometry Group
ViT	Vision Transformer
WR	William R Indicator
Xception	Extreme Inception