A CORRELATION-EMBEDDED ATTENTION APPROACH TO MITIGATE MULTICOLLINEARITY IN FOREIGN EXCHANGE DATA USING LSTM

LEOW MUN HONG STEVEN

MASTER OF PHILOSOPHY

FACULTY OF BUSINESS AND FINANCE
UNIVERSITI TUNKU ABDUL RAHMAN
JULY 2023

A CORRELATION-EMBEDDED ATTENTION APPROACH TO MITIGATE MULTICOLLINEARITY IN FOREIGN EXCHANGE DATA USING LSTM

By

LEOW MUN HONG STEVEN

A dissertation submitted to the Department of Finance,
Faculty of Business and Finance,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Philosophy
July 2023

ABSTRACT

A CORRELATION-EMBEDDED ATTENTION APPROACH TO MITIGATE MULTICOLLINEARITY IN FOREIGN EXCHANGE DATA USING LSTM

Leow Mun Hong Steven

Technologies currently drive the collection of big data in various fields, including algorithmic trading. This leads to a notable increase in the collection and storage of variables and data points (observations). While this offers opportunities to enhance the modeling of relationships between predictors and response variables, it also presents challenges in data analysis, such as the multicollinearity problem. Multicollinearity refers to the situation where two or more independent variables exhibit an approximately linear relationship. Existing feature selection methods might undermine efforts to gather more data, since it results in the exclusion of new data. This, in turn, can lead to the loss of important and relevant information. Recent studies indicate that neural networks are more adept at handling data with multicollinearity compared to statistical estimators. Consequently, this study proposes two improvements for the Long Short-Term Memory neural network (LSTM). These improvements involve the integration of the attention mechanism and vector embeddings of correlation to address multicollinearity without eliminating features. This innovative approach enables the handling of multicollinearity without discarding variables. The study compares the performance of regression and classification in predicting the direction of the foreign exchange market, using the EUR/GBP, EUR/USD, GBP/USD, and NZD/USD data sets over a 6-year period from 1 January 2015 to 31 December 2020. Specifically, it evaluates the accuracy of predictions and their impact on trading returns under high multicollinearity settings. Furthermore, the study assesses the difference between LSTM models with and without the proposed module. The results indicate that classification enhances regression accuracy by 23.33% and trading return by 132.62% over the test set. Additionally, the proposed module offers a further improvement of 59.53% in trading returns. These findings demonstrate the superiority of classification as a problem formulation in high multicollinearity scenarios. The experimental results also reveal that neural networks can learn the relevance and redundancy of financial data to enhance classification performance.

ACKNOWLEDGEMENT

First, I would like to express my gratitude towards my academic supervisor, Dr Cheng Wai Khuen, for his continued guidance and support throughout the course of my study and always being available with his critical suggestions for the improvement of my work. I would like to thank my co-supervisor Mr Jireh Chan Yi-Le for offering me valuable advice and encouragement.

I am thankful to my academic institution, Universiti Tunku Abdul Rahman, for providing the workspace and facilities which are required to finish my dissertation. I would like to extend my thanks to the Ministry of Higher Education of Malaysia for funding my research. Also, I would like to acknowledge Mr Bea Khean Thye, my fellow research partner for providing research ideas and creating a friendly environment at work.

Finally, I am indebted to my family members and friends who have motivated me continuously throughout my life. It would not have been possible to complete this dissertation without their encouragement and loving support.

APPROVAL SHEET

This dissertation entitled "**A CORRELATION-EMBEDDED ATTENTION APPROACH TO MITIGATE MULTICOLLINEARITY IN FOREIGN EXCHANGE DATA USING LSTM"** was prepared by LEOW MUN HONG STEVEN and submitted as partial fulfillment of the requirements for the degree of Master of Philosophy at Universiti Tunku Abdul Rahman.

Approved by:

____

(Ts. Dr. CHENG WAI KHUEN)
Date:……11/7/2023……..

Assistant Professor/Supervisor

Department of Computer Science

Faculty of Information and Communication Technology

Universiti Tunku Abdul Rahman

____

(Mr Jireh Chan Yi-Le)
Date:……11/7/2023……..

Co-supervisor

Department of Finance

Faculty of Business and Finance

Universiti Tunku Abdul Rahman

FACULTY OF BUSINESS AND FINANCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 10/7/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Leow Mun Hong Steven*** (ID No: ***20 ABM 02093***) has completed this dissertation entitled "*A Approach to Mitigate Multicollinearity in F Data*" under the supervision of Dr Cheng Wai Khuen (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology, and Mr Jireh Chan Yi-Le (Co-Supervisor) from the Department of Finance, Faculty of Business and Finance.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(Leow Mun Hong Steven)

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name ______________________________

Date __________10/7/2023__________

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF TERMS

Terms	**Definition**
Algorithmic trading	The process of using predefined rules and instructions executed by computer algorithms to automatically trade financial instruments in the market.
Artificial intelligence	The field of computer science that focuses on creating intelligent machines capable of mimicking human intelligence, such as learning, reasoning, and problem-solving.
Attention mechanisms	Techniques used in artificial neural networks to assign different weights or levels of importance to different parts of input data, enabling the network to focus on relevant information.
Big data	Large and complex datasets that cannot be easily managed or processed using traditional data processing methods.
Correlation	A statistical measure that quantifies the relationship between two variables, indicating how they move or change together.
Earnings reports	Financial documents released by companies to provide information about their financial performance and profitability over a specific period.
Financial market	A marketplace where buyers and sellers trade financial assets such as stocks, bonds, commodities, currencies, and derivatives.
Forecasting	The process of predicting or estimating future outcomes or events based on historical data and statistical models.
Fundamental analysis	An approach to evaluating investments by analyzing the intrinsic value of assets, examining financial statements, industry trends, and other relevant factors.

Generalization	The ability of a machine learning model to apply learned knowledge from training data to make accurate predictions on new, unseen data.
Internet of Things (IoT)	The network of interconnected physical devices, sensors, and other objects embedded with software, allowing them to collect and exchange data.
Macroeconomic data	Data that describes the overall economic conditions of a country or region, such as GDP, inflation rate, unemployment rate, and interest rates.
Microeconomic data	Data that relates to specific economic units or individual entities, such as individual company financials, consumer spending, and market demand.
Multicollinearity	A phenomenon in statistics where two or more predictor variables in a regression model are highly correlated, making it difficult to separate their individual effects.
Ordinary least square (OLS)	A method used to estimate the parameters of a linear regression model by minimizing the sum of squared differences between the observed and predicted values.
R-Squared	A statistical measure that represents the proportion of the variance in the dependent variable explained by the independent variables in a regression model.
Ridge regression	A technique used to address multicollinearity in regression models by adding a penalty term to the ordinary least square estimation.
Root mean square error	A measure of the average difference between predicted values and actual values, calculated by taking the square root of the average of squared differences.
Stepwise regression	A method of selecting and removing predictor variables in a regression model based on their statistical significance and contribution to the model's overall performance.
Technical analysis	An approach to forecasting financial markets that relies on historical price and volume data,

	patterns, and indicators to make investment decisions.
Technical indicators	Mathematical calculations or statistical tools applied to historical price and volume data to provide insights into market trends, momentum, and potential future price movements.
Time series data	Data collected and recorded over a series of consecutive and equally spaced time intervals, such as stock prices over daily, weekly, or monthly periods.
Vector embeddings	Representations of objects or entities in a vector space, often used in natural language processing (NLP) to capture semantic relationships between words or documents.

CHAPTER 1

INTRODUCTION

1.1. Background of the Study

1.1.1. Algorithmic Trading

Forecasting in finance currently involves a wide range of variables, including macroeconomic data, microeconomic data, earnings reports, and technical indicators. Multicollinearity remains a prevalent issue in finance due to the variable dependencies that can fluctuate over time and change due to economic events. Handling financial market data differs from time series data in other fields, and there are several key reasons for this (Iba & Sasaki, 1999). The primary objective when compiling stock market data is to maximize profit rather than minimizing prediction errors. Stock market data are highly time-dependent, meaning that the output relies on the timing of the input. Additionally, they are influenced by indeterminate events, indicating that the event triggering the response is not fixed. Algorithmic trading serves as a prime example of the challenges posed by multicollinearity in finance.

With the advancement of technologies and the availability of big data, algorithmic trading has gained significant popularity. It refers to the use of programmed software that automates one or more stages of the trading process

(Treleaven, Galas, & Lalchand, 2013). Algorithms are commonly employed in pre-trade analysis, where they utilize financial data or news to generate asset price forecasts (Nuti et al., 2011). The analysis can be categorized into fundamental analysis, which involves financial data, economic data, or news data, and technical analysis, which focuses on trend analysis and chart patterns. Algorithmic trading deals with a vast amount of data and continuously incorporates new information. It may encompass hundreds of variables, and even minor changes can have a significant impact on forecast performance. Since these forecasts are typically used for trading purposes, multicollinearity has substantial implications for the profitability of the system.

To illustrate this issue, the present study focuses on the use of technical indicators in stock analysis. A problem of multicollinearity arises when these indicators measure the same type of information, such as momentum (Bollinger, 1992). In such cases, different indicators are derived from the same series of closing prices. The aim of this study is to address this problem by minimizing multicollinearity. Previous literature attempted to eliminate collinear data to mitigate the effects of multicollinearity. This was achieved through stepwise regression, which eventually yielded a model with a low root mean square error (RMSE). The computational complexity of this approach led to the development of various selection criteria for model choice. Ridge regression emerged as a breakthrough method for tackling multicollinearity. Instead of selecting variables, ridge regression employs all variables.

Ridge regression adjusts the estimator by introducing a penalty term to the ordinary least square (OLS) estimators. The objective is to reduce variance by introducing bias. Subsequent papers have expanded upon these ideas, exploring different functional forms and enhancing performance. For instance, Algamal (2018) conducted a review on Poisson regression. Furthermore, advancements in computing power have brought mathematical optimization into variable selection. The progress made in machine learning and artificial intelligence has opened up new possibilities for mitigating multicollinearity. Obite et al. (2020) utilized a feed-forward artificial neural network to model data with multicollinearity and found that it outperformed traditional OLS in terms of RMSE.

This demonstrates that machine learning approaches with intricate architectures possess the capability to generate significantly improved parameter estimates compared to statistical methods. In the realm of financial prediction, machine learning methods, including Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN), are recognized for their superior performance in contrast to regression models (Wang et al., 2021)

1.1.2. Multicollinearity

Multicollinearity emerges as a potential issue when utilizing a multiple regression model. People outside the field of statistics often lack understanding regarding the various methods available to mitigate the impact of multicollinearity (Schroeder, Lander, & Levine-Silverman, 1990). Multicollinearity occurs when there is an approximate linear relationship between two or more independent variables. The equation below represents a multiple linear regression model:

$$y = \beta_0 + x_1\beta_1 + \ldots + x_p\beta_p + e \, , \tag{1}$$

here, y denotes the dependent variable, while x_1, . . ., x_p represent the explanatory variables. β_0 represents the constant term, and β_1, . . ., β_p represents the coefficients of the explanatory variables. The error term, e, accounts for the discrepancy between the observed value and the estimated value. It follows a normal distribution with a mean of 0 and variance σ. In the presence of multicollinearity, one explanatory variable, such as x_1, may exhibit a linear dependency on another explanatory variable, like x_2. This situation leads to an unreliable model.

Lafi and Kaneene (1992) outline four primary indications of multicollinearity. Firstly, there is a notable increase in the standard error of the coefficients. Additionally, the sign of a variable coefficient may deviate from what is expected in theory. Furthermore, a high correlation exists between the predictor variable and the outcome, yet the corresponding parameter lacks statistical significance. Lastly, some correlation coefficients among predictor variables are significantly large compared to the explanatory power or R-Squared of the overall equation. These symptoms are merely indicators and do not guarantee the presence of multicollinearity. Although multicollinearity does not violate the assumptions of ordinary least squares (OLS) regression, it poses two significant challenges. Firstly, the interdependence of variables leads to unstable estimates. Secondly, the standard errors of regression coefficients become inflated, rendering the estimates unreliable and reducing precision (Alin, 2010). As a result, the model's generalization ability diminishes, and it tends to overfit the available data, performing poorly on unseen data.

Previous studies have identified four approaches for measuring multicollinearity. The first method involves calculating pairwise correlations using a correlation matrix. Mason and Perreault Jr (1991) suggest that a bivariate correlation of 0.8 or 0.9 is often used as a threshold to indicate high correlation between two regressors. However, it is important to note that correlations alone do not necessarily imply multicollinearity as they represent a different concept. The most utilized indicator of multicollinearity is the Variation Inflation Factor (VIF) or Tolerance (TOL) (Neter, Kutner, Nachtsheim, & Wasserman, 1996). VIF is defined as:

$$VIF_j = \frac{1}{(1 - R_j^2)}, \qquad (2)$$

where R_j^2 is the coefficient of determination for the regression of x_j on the remaining variables. VIF is the reciprocal of TOL. While there is no definitive threshold value for VIF indicating the presence of multicollinearity, a value of 10 or higher is often considered indicative of multicollinearity (Weisberg, 2005).

Eigenvalues, derived from the Principal Component Approach (PCA), offer another method for assessing multicollinearity. A smaller eigenvalue suggests a higher likelihood of multicollinearity. The fourth measurement approach is the Condition Index (CI), which relies on eigenvalues. CI is the square root of the ratio between the maximum eigenvalue and each individual eigenvalue. Belsley, Kuh, and Welsch (2005) propose that a CI between 10 and 30 indicates moderate multicollinearity, while a CI above 30 suggests severe multicollinearity. VIF and CI are commonly employed to assess the severity of multicollinearity in a dataset before implementing methods to address it. It is worth noting that the effectiveness of these two approaches in reducing multicollinearity is typically evaluated by comparing the root mean square error or out-sample forecast before and after applying the treatments (Tamura et al., 2017).

1.1.3. Problem Formulation

It is widely recognized in the field that financial analysts often rely on accurate price level predictions to guide their trading practices. However, recent studies have suggested that forecasting strategies based on predicting the direction of price changes may be more effective and profitable. Leung, Daouk, and Chen (2000) discovered that forecasting models focused on the direction of stock returns outperform models based on the level of stock returns in terms of predicting stock market return direction and maximizing investment profits.

Over the past three decades, twenty highly cited studies in the fields of econometrics and financial forecasting have consistently framed financial forecasting as a point estimation problem (Alexander, 2008; Barndorff-Nielsen & Shephard, 2005; Barndorff-Nielsen & Shephard, 2002; Beck, 2008; Brooks, 2019; Brownlees & Gallo, 2006; Campbell, Lo, & MacKinlay, 1997; Chen et al., 2012; Chiriac & Voev, 2011; Christoffersen & Diebold, 2000; Engle, 2001, 2004; Giacomini & Rossi, 2010; Hördahl et al., 2006; Jasiak, 2001; Koop & Korobilis, 2012; Mills & Markellos, 2008; Paolella & Taschini, 2008; Patton, 2011). In essence, forecasting problems are commonly formulated as "What will X be after N periods?" where X represents a specific metric and N denotes the number of periods. This approach is particularly useful in various decision-making scenarios, such as forecasting earnings to calculate the Price-to-Earnings Ratio (PER) for determining stock valuation or predicting exchange rates for inventory management purposes.

Discretizing the sample space involves splitting the continuous sample space into intervals. In the context of financial forecasting, when the estimation problem is framed as interval-based, the prediction sample space becomes discrete. Rather than asking "What would X be after N periods?" where X represents the stock price and N denotes the number of time-steps, interval-based estimation poses the question "What is the probability that X will fall between Y and Z after N periods?" This shift in approach introduces the possibility of treating continuous forecasting as a classification problem, opening up opportunities for innovation in financial forecasting problem formulation.

The novelty of neural networks lies in their ability to model non-linear sequences and predict both continuous and discrete variables, unlike many statistical models that are limited to one or the other. This research aims to address the question of which approach—classification-based or point estimation—achieves better accuracy in forecasting the trend direction of stocks. Specifically, a Long Short-Term Memory network was developed to forecast foreign exchange using various technical indicators as input features. By comparing the trading returns of classification-based approaches to their point estimation counterparts, this study provides valuable insights and a developed model that can serve as a foundation for future research.

1.2. Problem Statement

Forecasting exact numerical values of a target variable in algorithmic trading using regression poses a challenge due to the presence of market volatility and nonlinearity. The presence of multicollinearity further complicates matters. However, classification can offer an alternative approach by categorizing data into discrete classes. This simplifies the prediction task and enhances resistance to noise. Classification models focus on identifying the general direction or category of market movements rather than precise values, and they can leverage additional data sources such as sentiment analysis. In this research, the focus is on exploring how problem formulation can improve the impact of multicollinearity.

The existing body of literature has largely overlooked the potential of neural network approaches in mitigating multicollinearity. Traditional feature selection methods prove inadequate when applied to neural networks because they fail to capture nonlinear relationships. Recent studies have shown that neural networks outperform statistical ordinary least squares (OLS) regression models when handling multicollinear data. Furthermore, researchers have emphasized the advantages of machine learning algorithms, including their ability to operate without strict assumptions about the underlying function, uncover complex patterns, and dynamically adapt to changing relationships (Rasekhschaffe & Jones, 2019; Obite et al., 2020; Wu & Feng, 2018).

Additionally, neural network approaches offer the opportunity to leverage attention mechanisms and vector embeddings to explore their effectiveness in handling multicollinearity in foreign exchange datasets. It has potential in retaining variables. In financial data, it is suboptimal to remove features due to their interrelated nature (Lucey & Muckley, 2011). When the objective is forecasting rather than hypothesis testing, utilizing a larger number of variables can yield better results even in the presence of multicollinearity (A.-S. Chen, Leung, & Daouk, 2003). This holds true in algorithmic trading, where the accuracy of forecast predictions directly impacts the algorithm's profitability. With the advancements in Internet of Things (IoT), Big Data, and digitization, a significant amount of data is becoming available, making the removal of variables a missed opportunity.

The above problems suggest that conventional methods of addressing multicollinearity have little application for algorithmic trading. This study proposed a machine learning approach that uses all available variables on a neural network. A neural network has the advantage of exhibiting significant non-linear characteristics, accounting for relationships with response variable. It should improve generalization ability compared to existing methods.

1.3. Research Questions

1. To what extent can Classification Neural Network mitigate multicollinearity when compared to Regression Neural Network?
2. To what extent can the proposed method mitigate multicollinearity when compared to Neural Network?
    a. How does the proposed attention mechanism and embeddings compare to neural network in improving prediction accuracy?
    b. How does the proposed attention mechanism and embeddings compare to neural network in improving trading returns?
3. To what extent can the proposed method applied to regression mitigate multicollinearity?

1.4. Research Objectives

1. To compare the performance of mitigating multicollinearity between Classification Neural Network and Regression Neural Network.
2. To investigate the potential improvement in performance of proposed method over Neural Network.
    a. To investigate the potential improvement in prediction accuracy of proposed attention mechanism and embeddings over neural network in the presence of multicollinearity.
    b. To investigate the potential improvement in trading returns of proposed attention mechanism and embeddings over neural network in the presence of multicollinearity.

3. To investigate the potential improvement in performance of proposed method on regression.

1.5. Contributions

This study aims to make significant contributions to both participants in financial markets and researchers in the field of data analysis. Specifically, a novel machine learning approach is developed to effectively mitigate the adverse effects of multicollinearity. The proposed method has the potential to enhance financial forecasting and algorithmic trading practices. Unlike traditional approaches, this method does not involve variable removal and instead utilizes all available variables within neural networks. The study investigates the impact of treating prediction as a classification problem versus a regression problem. Additionally, attention mechanisms are employed to assess the relevance of different variables, while vector embeddings of correlation are developed to identify redundant features. These two methods are integrated to improve the prediction model, employing an embedded approach to feature selection.

The purpose of the attention mechanism is to determine the variables that hold the most significance for predicting the target variable. It achieves this by assigning varying weights to each input variable. Initially introduced for sequence-to-sequence machine translation by Bahdanau et al. (2014), attention mechanisms have since evolved and found applications in other domains such as image data (Yuan et al., 2022). By incorporating attention, the predictive model can effectively capture the relationships between the variables and the desired output. Notably, attention mechanisms have demonstrated their utility in forecasting stock market price fluctuations using technical indicators (Lee, 2022). The proposed method offers promising advancements in financial forecasting and algorithmic trading, specifically in the context of handling multicollinearity. The model also employs correlations between variables as embeddings, providing the model with insights into feature redundancy.

The findings of this dissertation hold valuable insights for future researchers aiming to address multicollinearity in various fields beyond finance and algorithmic trading. The research makes two main contributions: firstly, it enables neural networks to effectively handle multicollinearity without the need for variable removal; and secondly, it enhances investment returns when applying the proposed mechanisms to a Long Short-Term Memory (LSTM) network. Furthermore, the study may inspire further research on utilizing neural networks to solve multicollinearity-related challenges.

1.6. Conclusion

The background of this study encompasses two key components: multicollinearity and algorithmic trading. Multicollinearity refers to a phenomenon that can occur when utilizing a multiple regression model, resulting in poor generalization ability and overfitting of the data. On the other hand, algorithmic trading pertains to the use of programmed software to automate various stages of the trading process. These algorithms often involve numerous variables, and even minor changes can significantly impact forecast performance. The presence of multicollinearity can have significant implications for the profitability of such systems. Therefore, accurate prediction of price levels is crucial in many trading practices employed by financial analysts.

This research addresses three primary issues. Firstly, recent studies have suggested that trading strategies based on forecasting the direction of price level changes may yield superior results. Secondly, advancements in mitigating multicollinearity have given limited attention to neural network approaches. Finally, this study proposes a novel approach that avoids variable removal when dealing with multicollinearity in financial data, recognizing the high interrelatedness of the variables.

CHAPTER 2

LITERATURE REVIEW

2.1. Multicollinearity in Algorithmic Trading

Algorithmic trading is a process of trading financial instruments using preprogrammed systems that gained popularity as technology advanced. It offered benefits to the market such as narrowing spreads and increasing trading activity. Yılmaz et al. (2015) found that algorithmic trading contributed to the efficiency of emerging markets like the Bursa Malaysia. In financial markets, technical analysis has been utilized for making investment decisions. It involved analyzing chart patterns, prices, and trading volume to predict future asset prices. This information was used to generate technical indicators related to trends, momentum, volume, and volatility. Traders used these indicators to receive signals for entering or exiting trades.

Forecasting time series data played a crucial role in finance and economics. Traditional approaches relied on statistical models that used past time lags to predict future values. One of the most well-known techniques was Autoregressive Integrated Moving Average (ARIMA). Neural networks, particularly Recurrent Neural Networks (RNNs), were capable of capturing dependencies between input sequences. Long Short-Term Memory (LSTM) networks, a type of RNN, could retain long-term information from data. Unlike traditional RNNs, LSTMs had memory cells and gates that allowed the model to discard irrelevant information from the previous time step while retaining important information from the current time step. LSTMs demonstrated promising performance in various sequence-based problems like translation, speech analysis, and voice recognition. This study chose the LSTM model due to its ability to model the temporal nature of time series data. Additionally, it served as a suitable benchmark for experimenting and addressing the issue of multicollinearity in algorithmic trading. Siami-Namini and Namin (2018) compared the performance of ARIMA and LSTM models, revealing an 85% improvement in prediction accuracy for stock market indexes using the LSTM.

Trading models were often comprised of numerous variables, including economic variables such as interest rates, exchange rates, monetary growth rates, and overall economic conditions. Industry-specific variables such as growth rates of industrial production and consumer prices, as well as company-specific variables like income statements and dividend yields, were also taken into account. Technical analysis, which involved analyzing chart patterns, price movements, and trading volume, along with news data related to important political events, played a significant role as well (Enke & Thawornwong, 2005). These factors interacted with each other, and even minor changes could have a significant impact on forecast performance.

Treleaven et al. (2013) provided an overview of the algorithmic trading process. Sophisticated neural networks were utilized to incorporate technical indicators and enhance the profitability of algorithmic trading. This was evident in the case of Bursa Malaysia, where it was observed that algorithmic trading outperformed the standard buy-and-hold strategy with the Kuala Lumpur Composite Index (KLCI) as a proxy for the stock market (M'ng & Aziz, 2016). Recent research by Rundo (2019) demonstrated that deep LSTM models could effectively predict short-term trends in foreign exchange rates, resulting in increased profits and reduced drawdowns. This process is illustrated in Figure 2.1.

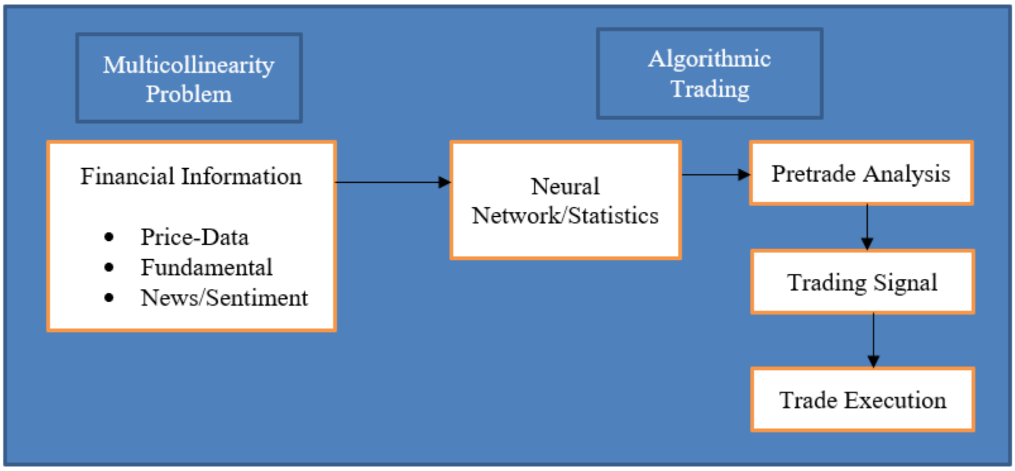


Figure 2.1 The algorithmic trading process.

The extensive volume of data encountered in stock analysis is susceptible to the influence of multicollinearity. This issue often arose when utilizing technical indicators, particularly those that measured similar information such as momentum (Bollinger, 1992). For instance, if multiple indicators were derived from the same series of closing prices, it would lead to multicollinearity concerns.

2.2. Solving the Multicollinearity Problem

In order to mitigate the impact of multicollinearity, one straightforward solution is to increase the amount of collected data, as multicollinearity is primarily a data issue rather than a problem with model specification. However, this approach is not always feasible, particularly when research relies on convenience sampling, as it may entail additional costs and compromise the data quality (Schroeder et al., 1990). Methods for addressing multicollinearity can be broadly categorized into two approaches: variable selection and modified estimators. Variable selection offers simplicity and the potential for creating a sparse model that is easy to interpret and less prone to overfitting. However, it has the drawback of being highly discretionary, assuming the existence of a single best model while multiple models with different variables can be equally valid.

Recent studies have utilized advances in computing power to tackle multicollinearity by framing the subset selection problem as an optimization task. By searching for the least redundant variables and optimizing for the most relevant ones, these approaches utilize criteria developed from previous literature to represent relevance and redundancy. This enables handling high-dimensional problems where the number of variables exceeds the number of observations. On the other hand, modifying estimators is a more complex and diverse approach. It involves adapting estimators to the specific functional form of the data, resulting in improved robustness and performance in the presence of multicollinearity. However, interpretability becomes a challenge when coefficients are close to but not exactly zero. Nevertheless, certain modified estimators are capable of performing variable selection along with their enhanced robustness.

In general, both feature selection methods and modified estimators have their advantages and disadvantages. Feature selection methods aim to reduce the number of variables to the most relevant ones, which can potentially lead to a loss of information from the available data. Moreover, modern optimization techniques rely on subjectively determined indicators of relevance and similarity, as highlighted by Tamura et al. (2017), indicating the need for exploring alternative measures of multicollinearity in future research. Thus, without directly comparing their performance on the same dataset, it is challenging to determine which method is superior. Additionally, finding a globally optimal subset without conducting an exhaustive search is difficult and computationally expensive.

Katrutsa and Strijov (2015) conducted a stress test experiment to compare the performance of various variable selection methods, including Stepwise, Ridge, Lasso, Elastic Net, LARs, and Genetic algorithms. They evaluated these methods using several quality measures on synthetic datasets. Similarly, Garg and Tai (2013) compared different statistical and machine learning methods. However, it is important to note that comparisons between methods have limitations. For instance, variations in tuning parameters can influence the performance of the methods. Hamaker (1962) emphasized the significance of domain knowledge in the field of study for selecting variables, as relying solely on statistics may not be sufficient in practical applications. Each method exhibited different degrees of performance when applied to different types of data.

Both variable selection and modified estimators can be employed in conjunction with each other. One approach is to first rapidly reduce the number of features to a level below the number of samples using variable selection methods, and then apply modified estimators. This methodology can be observed in various machine learning studies. The subsequent sub-topics provide detailed explanations of the variable selection and modified estimator methods. Additionally, the literature review also covers the introduction of machine learning approaches.

2.2.1. Variable Selection Methods

In general, researchers in the past attempted to address the impact of multicollinearity by utilizing variable selection techniques to obtain more reliable parameter estimates (Askin, 1982). These techniques typically involved heuristic algorithms and relied on indicators to combine or eliminate variables. However, caution was necessary to prevent compromising the underlying theoretical model while reducing multicollinearity. One of the earliest methods used was stepwise regression, which encompassed two primary approaches: forward selection and backward elimination (Ralston & Wilf, 1960). The forward selection method started with an empty model and gradually added variables one by one, while the backward elimination method began with a full model containing all variables and progressively removed them. In forward selection, the variable with the greatest decrease in residual sum of squares was chosen at each stage, while in backward elimination, the variable with the lowest increase in residual sum of squares was eliminated.

However, stepwise regression had certain limitations. According to Hamaker (1962), it did not always yield the best model in terms of residual sum of squares, especially in the presence of multicollinearity, due to the order in which variables were added. It remained unclear which of the two stepwise regression methods was superior. Moreover, stepwise regression assumed the existence of a single optimal equation, overlooking the possibility of multiple equations with equally favorable variables. Another concern was the computational effort required by the selection criterion (Hocking & Leslie, 1967). With k independent variables, there were 2k potential combinations, and the computational workload increased exponentially with the total number of independent variables.

To expedite computation time, Gorman and Toman (1966) devised a more comprehensive approach for fitting equations to the data. They utilized a fractional factorial design along with the statistical criteria Cp to avoid the need for evaluating all possible equations. This method proved to be more effective when dealing with data affected by multicollinearity, as it assessed the efficacy of a variable based on its inclusion or exclusion from an equation. The Cp criterion, developed by Mallows (1964), facilitated graphical comparisons between different equations. The Cp selection criterion can be expressed as:

$$Cp = \frac{RSS_p}{\hat{\sigma}^2} - (n - 2p)\,, \tag{3}$$

where, p represents the number of variables, RSS denotes the residual sum of squares for the regression being considered, and $\hat{\sigma}^2$ is an estimation of σ^2, often obtained from the residual mean square of the complete regression. A lower value of Cp indicates a better model.

Subsequently, Kashid and Kulkarni (2002) proposed a more generalized selection criterion, Sp, which exhibited superior performance compared to the Cp criterion. Unlike the Cp criterion, which relies on the least square estimator and is susceptible to outliers and deviations from normality in the error variable, the Sp criterion addresses these issues and can be employed with any estimator of β without requiring modifications. The Sp criterion is defined as:

$$S_p = \sum_{i=1}^{n} \left(\hat{Y}_{ik} - \hat{Y}_{ip}\right)^2 / \sigma^2 - (k - 2p)\,, \tag{4}$$

where k and p represent the parameters of the full and subset models, respectively.

Information criteria offer an appealing approach for model selection. Other commonly utilized criteria include the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and so on (Montgomery et al., 2021). According to Vrieze (2012), the difference between AIC and BIC lies in the fact that BIC consistently selects the model when the true model is under consideration, whereas AIC aims to minimize risk functions when the true model is not among the candidate models. The choice of criterion depends on the researcher, and it is recommended to use both AIC and BIC in conjunction.

Table 2.1 provides a summary of each stepwise feature selection and quality criterion.

Table 2.1 Stepwise feature selection and quality criterions.

Author	Objective	Method	Pros	Cons
Ralston and Wilf (1960)	Develop a method for model selection	Forward selection and backward elimination	Simple to understand and use	Final model affected by order
Mallows (1964)	A criterion for subset selection	*Cp* criterion	Graphically compare quality between models	Suffers with outlier and non-normality
Gorman and Toman (1966)	Fractional factorial design to model selection	Fractional factorial design with the statistical criteria, *Cp*	Avoid computing all possible model	Heuristic technique
Kashid and Kulkarni (2002)	A more general selection criterion than *Cp* when least square is not best	*Sp* criterion	Applicable on any estimator	Computationally difficult and not consistent result
Misra and Yadav (2020)	Improve classification accuracy in small sample size	Recursive Feature Elimination with Cross-Validation	Does not delete the records	Evaluated on small sample size

Lafi and Kaneene (1992) proposed the utilization of principal component analysis (PCA) as a means to address multicollinearity among predictor variables in a regression model. PCA is a statistical method that transforms variables into new uncorrelated variables known as principal components, effectively reducing the number of predictive variables. Regression analysis is then performed using these principal components. The principal components are independent, thereby satisfying the ordinary least squares (OLS) assumption. They are ranked based on the magnitude of their variances, with principal component 1 accounting for more variation than principal component 2, thus making PCA useful for dimensionality reduction. Principal components with eigenvalues close to zero can be eliminated, resulting in a sparse model while retaining potentially valuable information contained in the variables.

The Partial Least Squares (PLS) method was developed by Wold (1982) and represents a superior alternative to multiple linear regression and PCA due to its robustness. The model parameters do not undergo significant changes when new samples are introduced. PLS, like PCA, is a dimension reduction technique, but it captures the characteristics of both X and Y variables instead of solely focusing on X. The PLS method iteratively extracts factors from X and Y while maximizing the covariance between the extracted factors. PLS proves useful for analyzing noisy data affected by multicollinearity, as its underlying assumptions are more realistic compared to traditional multiple linear regression (Wold et al., 2001). Chong and Jun (2005) conducted a comparison between the PLS method, the lasso method, and the stepwise method, finding that PLS performed better.

Several journals have conducted comparisons between these techniques. Maitra and Yan (2008) discussed and compared PCA and PLS, as both methods serve as dimension reduction methodologies. They are employed to convert highly correlated variables into independent variables and reduce the number of variables. PCA does not consider the relationship between predictor variables and response variables, whereas PLS does. Therefore, PCA is an unsupervised dimension reduction technique, while PLS is a supervised technique. The study also revealed that the predictive power of principal components does not necessarily align with their order. For instance, principal component 1 may explain less variance in the response variable compared to principal component 2. In this regard, PLS is more efficient since it is a supervised technique that extracts components based on their significance and predictive power.

Maitra and Yan (2008) conducted a simulation study to compare three regression methods: partial least square regression (PLSR), ridge regression (RR), and principal component regression (PCR). The mean squared error (MSE) was used as a metric for comparison. The results indicated that as the number of independent variables increased, PLSR performed the best. However, when the number of observations and the level of multicollinearity were sufficiently large while the number of independent variables remained small, RR exhibited the smallest MSE. An example of a recent application of PLS can be seen in chaos phase modulation techniques for underwater acoustic communication. Li et al. (2018) integrated PLS regression into chaos phase modulation communication to mitigate the effects of multicollinearity. They described PLS as a machine learning method that simultaneously incorporates training and testing processes. The study demonstrated that this method effectively enhanced communication signals. The authors compared PLS regression with two other algorithms, namely the time reversal demodulator and the 3-layer back propagation neural network, which lacked feature analysis and relationship analysis. The results showed that PLS regression yielded the best performance.

Willis et al. (1997) initially developed a multigene genetic programming approach. Castillo and Villa (2005) utilized this method to automate predictor selection and alleviate multicollinearity issues. Bies et al. (2006) described a machine learning approach based on genetic algorithms for variable selection. Genetic algorithms are general optimization algorithms inspired by concepts such as evolution and survival of the fittest. The model was initialized by creating a population consisting of several individuals, each representing a different model. The genes of the model represented the model's features. An objective function was used to evaluate the fitness of the models. In each subsequent generation or iteration, the best-performing model was selected, and its genes underwent crossover, combining certain features from the parent models. Mutation could also occur with a determined probability, resulting in the reversal of certain features.

According to Bies et al. (2006), a hybrid model combining a derivative-based search algorithm with the machine learning concept is recommended. Genetic algorithms are effective in finding generally good solutions but may struggle with locating local minima, which is where derivative-based search algorithms excel. The derivative-based search algorithms can be applied after a certain number of iterations of the genetic algorithm. The iterations continue until no further improvement in the model's fitness is observed.

Katrutsa and Strijov (2017) proposed a quadratic programming approach to feature selection, addressing the limitation of previous methods that did not consider the dataset's configuration and were not problem-dependent. The objective of using quadratic programming was to maximize the inclusion of relevant variables while reducing redundancy. The quality criterion, denoted as Q, represented the quality of a subset of features a, and it was presented in quadratic form as $Q(a) = a^T Q a - b^T a$. Here, Q∈ R^{nxn} represented a matrix of pairwise predictor similarities, and b∈R^n represented a vector indicating the relevance of the predictor to the target variable. The authors suggested that the similarity between features x^i and x^j, as well as between x^i and y, could be measured using the Pearson correlation coefficient (Hall, 1999) or the concept of mutual information (Peng, Long, & Ding, 2005). However, these methods did not directly capture feature relevance. To address this, the authors employed a standard t-test to estimate the normalized significance of the features. This proposed method outperformed other feature selection methods such as Stepwise, Ridge, Lasso, Elastic Net, LARS, and the genetic algorithm.

Senawi, Wei, and Billings (2017) introduced the maximum relevance-minimum multicollinearity (MRmMC) method for variable selection and ranking. This approach considers both relevance and redundancy. Relevancy represents the relationship between features and the target variable, while redundancy indicates multicollinearity among features. One advantage of this method is that it doesn't require parameter tuning and is relatively straightforward to implement. Relevant features are assessed using the correlation coefficient, while redundancy is measured using the squared multiple correlation coefficient. They developed a measure J that combines both relevancy and multicollinearity:

$$J(f_j) = \ [r_{qn}^2(f_j, c) - \sum_{i=1}^{k} sc(f_j, q_i)] \ , \tag{5}$$

where r^2 is the correlation coefficient between feature f and target variable *c*, and *sc* is the squared multiple correlation coefficient between feature *f* and its orthogonal transformed variable *q*. The first feature is selected using the optimization criterion V, and subsequent features are selected based on the J criterion using a forward stepwise approach. Although not exhaustive, this method proves to be highly effective for feature selection and dimensionality reduction.

Tamura et al. (2019) suggested that mixed integer optimization (MIO) approaches for variable selection have gained attention due to advancements in algorithms and hardware. They developed a mixed integer quadratic optimization (MIQO) method to address multicollinearity in linear regression models. The MIQO method utilizes the variance inflation factor (VIF) as an indicator for detecting multicollinearity. Subset selection is performed while imposing an upper bound constraint on the VIF of each variable. This approach achieved higher R-Squared values compared to heuristic-based methods like stepwise selection. Furthermore, the solution is computationally tractable and simpler to implement than the cutting plane algorithm.

Zhao et al. (2020) introduced the profiled independence screening (PIS) method for variable screening in datasets with high dimensionality and highly correlated predictors. This method builds upon the sure independence screening

(SIS) approach proposed by Fan and Lv (2008). However, SIS faced challenges when dealing with highly correlated predictors, leading to the development of PIS. To eliminate predictor correlation, a factor profiling operator $Q(Z_I) = I_n - Z_I(Z_I^T Z_I)^{-1} Z_I^T$ was introduced. The profiled data is then used in SIS. $Z_I \in R^{nxd}$ is the latent factor matrix of X and d is the number of latent factors. Factor profiling is as follows

$$Q(Z_I)y = Q(Z_I)X\beta + Q(Z_I)\varepsilon\,, \qquad (6)$$

In the factor profiling step, $Q(Z_I)y$ represents the profiled response variable, and the columns of $Q(Z_I)X$ represent the profiled predictors. However, PIS can be misleading in a spiked population model. To address this, preconditioned profiled independence screening (PPIS) combines preconditioning with factor profiling. Two real data analyses demonstrated the good performance of PPIS.

In addition, outlier detection can be a viable approach for variable selection. Larabi-Marie-Sainte (2021) recently used projection pursuit for outlier detection-based feature selection. Projection pursuit aims to identify the most interesting linear projections, and the author optimized it specifically for outlier detection. This method was found to be effective in improving classification tasks. However, its performance tends to be poor when most features are highly correlated or when the features are binary. Table 2.2 provides a summary of the findings for various variable selection approaches.

Table 2.2 A summary of previous studies on variable selection.

Author	**Objective**	**Method**	**Pros**	**Cons**
Wold (1982)	Creates new components using the relationship of predictor and response	Partial Least Square (PLS)	Supervised component extraction	Cannot exhibit significant non-linear characteristics
Lafi and Kaneene (1992)	Using PCA to perform regression	Principal component analysis (PCA)	Reduce dimensions	Does not account for relationship with response variable
Bies et al. (2006)	Genetic algorithm-based approach to model selection	Genetic algorithm	Less subjectivity on model	Not good in finding local minima
Fan and Lv (2008)	Screening with correlation learning	Sure Independence screening	Reduce dimensionality	Assume true model is linear
Genuer et al. (2010)	Find important variables for interpretation	Random Forest	Variable importance ranking	Not diagnose for variable correlation
Andrews and McNicholas (2014)	Variable selection for clustering and classification (VSCC)	Stepwise Algorithm	Minimize within-group variance	Require effective subset selection criterion
Zeng and Xie (2014)	Group variable selection	Smoothly Clipped Absolute Deviation	Preserve variable selection accuracy	Not computationally efficient as the forward procedure
Katrutsa and Strijov (2017)	Quadratic programming approach	Quadratic programming	Investigates the relevance and redundancy of features	Cannot evaluate multicollinearity between quantitative and nominal random variable.
Senawi et al. (2017)	Feature selection and ranking	Maximum relevance-minimum	Works well with classifying problems	Non-exhaustive

		multicollinearity (MRmMC)		
Tamura et al. (2017)	Mixed integer optimization	Mixed integer semidefinite optimization (MISDO)	Uses backward elimination to reduce computation	Only applies to low number of variables
Tamura et al. (2019)	Mixed integer optimization	Mixed integer quadratic optimization (MIQO)	Uses VIF as indicator	Only applies to low number of variables
Chen et al. (2020)	Combines the result of filter, wrapper, and embedded feature selection	Ensemble feature selection	Overcome local optima problem	Higher computation cost than single solution
Zhao et al. (2020)	Variable screening based on sure independence screening (SIS)	Preconditioned profiled independence screening (PPIS)	Variable screening in high dimensional setting	Require decorrelation of the predictors
Larabi-Marie-Sainte (2021)	Feature selection based on outlier detection	Projection Pursuit	Found outliers correlated with irrelevant features	Does not work well when features are noisy
Singh and Kumar (2021)	Creates new variables	Linear combination and ratio of independent variables	Does not remove any variables	Based on trial-and- error

The objective of variable selection methods is to decrease the number of variables to a select few that hold the highest relevance. However, this reduction in variables may result in a decrease in the amount of information gained from the available data. Additionally, contemporary optimization methods rely on indicators of relevance and similarity that are determined subjectively. Tamura et al. (2017) highlighted the need for exploring alternative measures of multicollinearity in future research, illustrating the subjective nature of these indicators. Consequently, it becomes challenging to determine which method is superior without directly comparing their performance on the same dataset. It is worth noting that improved performance could also be attributed to the specific problem being examined.

2.2.2. Modified Estimator Methods

Modified Estimators were introduced as an alternative approach that employed biased and shrunken estimators to mitigate overfitting by reducing variance (Askin, 1982). Although dropping variables did not compromise the integrity of the theoretical model, the estimators became biased. The most well-known technique within this approach is ridge regression, which was developed by Horel (1962). Ridge regression incorporates a penalty term, the squared magnitude of the coefficient β, into the loss function. The cost function for ridge regression can be expressed as follows:

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\,\beta_j)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2 \,, \tag{7}$$

One challenge with ridge regression is determining the appropriate ridge parameter, ƛ. If ƛ equals zero, the estimate will be equivalent to the ordinary least square estimate. However, if ƛ is excessively large, it can result in underfitting the model. The selection of ƛ involves searching for the minimal increase in root mean square error (RMSE) while effectively decreasing the ridge variable inflation factors for each variable.

To address this issue, a ridge trace is employed. This trace represents a plot of the coefficients, $\widehat{\beta}$, against ƛ. It helps identify the smallest ƛ value at which the coefficients start to level off. Alternatively, a validation dataset can be utilized to minimize the sum of squared errors (SSE) and determine the optimal ƛ. This involves identifying ƛ such that the reduction in the variance term of the slope parameter surpasses the increment in its squared bias. Duzan and Shariff (2015) reviewed estimation methods for Λ and proposed new approaches. Additionally, Assaf, Tsionas, and Tasiopoulos (2019) introduced a Bayesian approach to tackle the problem of finding the ridge parameter, demonstrating its robustness and flexibility in handling multicollinearity through simulation results. Another method for determining the ridge parameter was proposed by Roozbeh et al. (2020), who developed a generalized cross-validation approach capable of finding the global minimum.

Furthermore, several estimators have been derived based on the ridge estimator. Singh et al. (1986) employed a jack-knife procedure to reduce the considerable bias in estimators resulting from ridge regression. Kejian (1993) introduced the Liu estimator, a new class of estimator based on the ridge estimator, which offered the advantage of a simple procedure for finding the parameter ƛ due to its linear relationship with the estimate. Liu (2003) proposed the Liu-Type estimator, observing that ridge regression's shrinkage effect was inadequate when confronted with severe multicollinearity. The Liu-Type estimator exhibited lower mean squared error (MSE) compared to ridge regression and effectively addressed severe multicollinearity. Consequently, variations of ridge and Liu-Type estimators have been developed for different regression models. Inan and Erdogan (2013) proposed a Liu-Type estimator for binary logistic regression, extending the application of the Liu-Type estimator to linear models.

According to Huang and Yang (2014), limited attention had been given to shrinkage estimators for generalized linear regression models, including Poisson regression, logistic regression, and negative binomial regression. As a result, they introduced a two-parameter shrinkage estimator specifically for negative binomial models. This estimator combined the ridge estimator and Liu estimator. Türkan and Özel (2016) made modifications to the Jackknifed ridge regression estimator, creating a Modified Jackknifed Poisson ridge regression estimator. Algamal (2018) conducted a review of biased estimators in the Poisson regression model in the presence of multicollinearity. They found that the regular maximum likelihood method for estimating regression coefficients was unreliable under multicollinearity. Comparing the performance of four estimators, including the widely used ridge estimator, they discovered that Liu-type estimators demonstrated superior performance in the Poisson regression model.

In an effort to address the limitations of regular ridge regression, Chandrasekhar et al. (2016) proposed partial ridge regression. The regular ridge regression introduced bias to all variables regardless of the degree of multicollinearity, achieved stability at the expense of mean squared error (MSE), and had an arbitrary method for selecting the ridge parameter (λ). The proposed method applied the ridge parameter only to variables with a high degree of collinearity. This approach improved the precision of parameter estimation while maintaining MSE close to that of ordinary least squares (OLS). The estimates were closer to the true OLS estimate, β, and the overall variance significantly decreased. The partial ridge regression outperformed existing methods in terms of bias, MSE, and relative efficiency.

The Lasso regression was developed by Tibshirani (1996) as a solution to the issues encountered with both stepwise regression and ridge regression. One such problem is interpretability. While stepwise regression is interpretable, the inclusion or exclusion of variables from the model occurs discreetly, without a clear understanding of the underlying reasons. On the other hand, ridge regression effectively handles multicollinearity by stabilizing the shrunken coefficients. However, it does not reduce coefficients to zero, resulting in models that are difficult to interpret. The Lasso, also known as L1 regularization, and ridge regression, known as L2 regularization, differ mainly in the fact that Lasso reduces certain parameter estimates to zero, effectively selecting variables. The cost function for Lasso is expressed as follows:

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\,\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \ , \tag{8}$$

Here, Y represents a vector of responses (nx1), X is a matrix of predictor variables (nxp), and β is a vector of unknown constants (px1). Similar to ridge regression, as the value of ƛ approaches zero, the equation becomes closer to the least square estimate. However, when the ƛ value is significantly large, the coefficients approach zero. While ridge regression shrinks the estimators without performing variable selection, Lasso achieves both objectives.

Consequently, Lasso is more desirable, as it provides a more parsimonious model and enhances the explanation of the relationship between independent and dependent variables.

In the presence of multicollinearity, ridge regression and lasso regression exhibit distinct behaviors. Ridge regression tends to distribute the effect evenly and shrink the estimators of all variables, whereas lasso regression is more unstable and tends to retain one variable while eliminating others. Lasso regression performs poorly when the number of variables (p) exceeds the number of observations (n), as it can select a maximum of n variables. In cases where n > p, lasso regression's performance is not as effective as ridge regression. To address the limitations of ridge and lasso regression, Zou and Hastie (2005) proposed the elastic net, which combines both regularization methods. The elastic net not only offers the advantages of the regularization techniques but also exhibits a grouping effect by grouping highly correlated variables together. It either retains or eliminates all variables within a group simultaneously. Typically, the tuning parameter in the elastic net is chosen using cross-validation, a technique originally introduced by Mosier (1951). Cross-validation involves reserving a subset of the sample for validation in order to assess the model's performance.

Efron et al. (2004) developed the least angle regression (LARs) algorithm as a computationally simpler alternative inspired by lasso regression and stagewise regression. LARs initially sets all coefficients to zero and then adds the predictor with the highest correlation to the response. The next variable is chosen based on its correlation with the current residuals. LARs proceeds equiangularly between predictors along the "least angle direction" until the next most correlated variable. Roozbeh et al. (2021) also made improvements to lasso regression by utilizing a mixed-integer programming approach, which eliminates structured noise and improves performance in high-dimensional environments where p > n. Furthermore, Roozbeh et al. (2022) expanded on this concept by developing several penalized mixed-integer nonlinear programming models. These models can be solved using a metaheuristic algorithm.

Nguyen and Ng (2020) introduced a modified log penalty function, which was strictly concave in contrast to the strictly convex penalty function used in Elastic net. This modification aimed to achieve a parsimonious model even in the presence of multicollinearity. Unlike methods such as Elastic net, which emphasize the grouping effect by including collinear variables together, the modified log penalty function had a different objective. Table 2.3 summarizes the findings regarding the approaches using modified estimators.

Table 2.3 A Summary of previous studies on Modified Estimators.

Author	Objective	Method	Pros	Cons
Horel (1962)	Adds bias in exchange for lower variance	Ridge regression	Reduces overfitting	Introduces significant amount of bias
Singh et al. (1986)	Address significant amount of bias in ridge regression	Jack-knife procedure	Simple method to obtain confidence intervals for the regression parameters.	Larger variance than ridge regression
Kejian (1993)	Simple procedure to	Liu estimator	Ridge estimate is a linear	Does not work in severe multicollinearity

	find ridge parameter		function of ridge parameter	
Tibshirani (1996)	Address interpretability of stepwise and ridge regression	Lasso regression	Reduces coefficient to zero	Worse performance than Ridge and does not work when p>n
Liu (2003)	Existing method does not work in severe multicollinearity	Liu-type estimator	Allows large shrinkage parameter	Two parameter estimation
Efron et al. (2004)	Computational simplicity	Least angle regression (LARs)	Computationally simpler Lasso	Very sensitive to the presence of outliers
Zou and Hastie (2005)	Combines Ridge and Lasso regression	Elastic net	Achieves grouping effect	No parsimony
Chandrasekhar et al. (2016)	Applies Ridge parameters only on variable with high collinearity	Partial ridge regression	More precise parameter estimates	Subjective measure of high collinearity
Assaf et al. (2019)	A conditionally conjugate prior for the biasing constant	Bayesian approach to finding ridge parameter	Produce a marginal posterior of parameter given the data	Only focus on getting a single parameter
Nguyen and Ng (2020)	Strictly concave penalty function	Modified log penalty	Parsimony variable selection under multicollinearity	No grouping effect
Kibria and Lukman (2020)	Alternative to the ordinary least squares estimator	Kibria–Lukman estimator	Outperforms Ridge and Liu-type regression	Results depends on certain conditions
Arashi et al. (2021)	High-dimensional alternative to Ridge and Liu	Two-parameter estimator	Has asymptotic properties	Lower efficiency in sparse model
Qaraad et al. (2021)	Tune parameter alpha of Elastic Net	Optimized Elastic Net	Effective with imbalanced and multiclass data	Accuracy metric not discussed

Modified estimators aimed to enhance the efficiency of parameter estimation when dealing with multicollinearity. However, achieving this improvement involved a trade-off between bias and variance. Researchers had the flexibility to choose among different methods based on their specific objectives, such as emphasizing the grouping effect or prioritizing parsimony in the model. Nevertheless, determining the most suitable method for a particular problem often required extensive knowledge. Factors such as the dimensionality (high or low), the degree of multicollinearity, and the functional form of predictions or classification problems also influenced the effectiveness of different methods. It was necessary to modify certain methods originally designed for linear regression to adapt them to other types of predictions or classification tasks.

2.2.3. Neural Network Approaches

This section aimed to provide an overview of the multicollinearity problem in machine learning and introduce notable algorithms that implicitly addressed it. It has been demonstrated that neural networks outperform traditional statistical models. Obite et al. (2020) employed a feed-forward artificial neural network to model data with multicollinearity and found that it achieved significantly better performance in terms of RMSE compared to the traditional ordinary least squares (OLS) method. This highlights the potential of machine learning methods with more complex architectures to generate more accurate parameter estimates than statistical approaches. Chandrasekhar et al. (2016) provided explanations as to why machine learning algorithms may be more effective, as they do not require assumptions about the underlying function, can uncover intricate patterns, and dynamically learn changing relationships.

Furthermore, variable selection methods have been employed within neural networks. Garg and Tai (2012) proposed a hybrid approach that combined factor analysis and artificial neural networks to address multicollinearity. Since neural networks cannot perform variable selection directly, factor analysis (FA) was employed to extract components, which were then used as input for the neural network. This method, called FA-ANN (factor analysis - artificial neural network), was compared to regression analysis and genetic programming, and it demonstrated the highest accuracy. The advantage of FA-ANN and genetic programming was their lack of reliance on statistical assumptions, making them more reliable and trustworthy. Moreover, they were capable of generalizing over new sample data, unlike regression analysis. However, a drawback of these approaches was their "black-box" nature, making them difficult to interpret.

In recent research, this approach has been applied in the field of quality control. Kim et al. (2020) proposed a residual control chart for data with multicollinearity, where they suggested using a neural network instead of a generalized linear model (GLM) due to the asymmetric distribution of the data. They concluded that neural network models and functional PCA (FPCA) were suitable for handling high-dimensional and correlated data. Additionally, regularization and penalty mechanisms have been employed to address multicollinearity in machine learning models. Examples of such algorithms include the Regularized OS-ELM algorithm (Huynh & Won, 2011), OS-ELM Time-varying (OS-ELM-TV) (Ye et al., 2013), Timeliness Online Sequential ELM algorithm (Gu et al., 2014), Least Squares Incremental ELM algorithm (Guo et al., 2014), and Regularized Recursive least-squares (Mahadi et al., 2022).

However, these mechanisms increase the computational complexity. For this reason, Nobrega and Oliveira (2019) proposed a method called Kalman Learning Machine (KLM). It is an Extreme Learning Machine (ELM) that uses a Kalman filter to update the output weights of a Single Layer Feedforward Network (SLFN). Kalman filter is an equation that can efficiently estimate the state of a process that minimizes mean squared error. The state does not get updated in the learning stage like the concept of ELM. The resulting model has shown to outperform basic machine learning models in prediction error (RMSE) and computing time. However, it requires manual optimization by humans. A constructive approach to building the model is suggested. Although deep learning (DL) has emerged as an efficient method to automatically learn the data representation without the feature engineering, its discussion in terms of multicollinearity is very limited.

Based on this motivation, this study discussed the properties of neural networks such as convolutional neural network (CNN), recurrent neural network (RNN), attention mechanism and graph neural network before illustrating the example in mitigating the multicollinearity issue. This can be seen on our work in (Chan et al., 2022). CNN is a neural network which was first introduced by LeCun et al. (1998) in the field of computer vision. It developed the concept of local receptive fields and shared weights to reduce the number of network parameter. It is very interesting in its way of addressing relationship between features. Traditional deep neural network suffers from booming parameter issue. CNN adopted multiple convolutional and pooling (subsampling) layers to detect the most representative features before connecting to a fully connected network for prediction.

Specifically, in the past, the convolutional layer was utilized to apply multiple feature extractors (filters) to detect local features and generate corresponding feature maps to represent each local feature. The combination of multiple feature maps could represent the entire series. The pooling layer was used as a dimensionality reduction method to extract the most representative features and reduce noise. The resulting feature maps were likely to be independent of each other, potentially mitigating the issue of multicollinearity. For instance, Hoseinzade and Haratizadeh (2019) proposed the CNNpred framework to model the correlations among different variables in predicting stock market movement. Their paper introduced two variants of CNNpred, namely 2D-CNNpred and 3D-CNNpred, to extract combined features from a diverse set of input data, including major US stock market indices, currency exchange rates, future contracts, commodities prices, treasury bill rates, and more. Their results demonstrated a significant improvement in predictive accuracy compared to state-of-the-art baselines.

Another noteworthy study by Kim and Kim (2019) proposed integrating features learned from different representations of the same data to predict stock market movement. They incorporated chart images (e.g., Candle bar, Line bar, and F-line bar) derived from stock prices as additional inputs for predicting the movement of the SPDR S&P 500 ETF. The proposed model combined Long Short-Term Memory (LSTM) and CNN models to leverage their respective strengths in extracting temporal and image features. The results showed that integrating temporal and image features from the same data efficiently reduced prediction errors. In addition to feature maps, another significant development in Recurrent Neural Network (RNN) is the attention mechanism.

RNN, initially introduced by Elman (1990), was designed to process sequential information. According to Young et al. (2018), the term "recurrent" describes the general architecture idea where a similar function is applied to each element of the sequence, and the computed output of the previous element is aggregated and retained in the internal memory of the RNN until the end of the sequence. This enables RNN to compress information and generate a fixed-size vector to represent a sequence. The recurrent operation of RNN is advantageous in handling series data as it effectively captures the inherent information in sequential data. Unlike CNN, RNN is more flexible in modeling sequences of variable length, allowing it to capture unbounded contextual information. However, Bahdanau et al. (2014) raised concerns about the ability of recurrent-based models to handle long-range dependencies in data due to the memory compression issue, where the neural network struggles to compress all the necessary information from a long sequence input into a fixed-length vector.

In other words, in the past, it was challenging to represent the entire input sequence without any information loss using a fixed-length vector. Despite the use of gated activation functions, the issue of forgetting in RNN-based models became more significant as the length of the input sequence increased. To address this, the attention mechanism was proposed to handle long-range dependencies by allowing the model to focus on the relevant parts of the input sequence when predicting specific parts of the output sequence. The attention mechanism has been adapted to various fields of study, including finance. For instance, Zhang et al. (2021) introduced a CNN based on deep factorization machine and the attention mechanism (FA-CNN) to enhance feature learning. In addition to capturing temporal influences, the attention mechanism enabled the modeling of intraday interactions among input features. This research aimed to apply the attention mechanism as the weighting process simulates the feature selection process in traditional statistical methods.

Recently, another promising approach has emerged, involving the application of Graph Convolutional Networks (GCN) or graph embeddings in series data. Graph neural networks convert series data into graph-structured data, allowing the model to capture the interconnectivity between nodes. This interconnectivity or correlation modeling proves useful in reducing the effects of multicollinearity. For instance, Kim et al. (2019) proposed the hierarchical graph attention network (HATS) to process relational data for stock market prediction. Their study defined the stock market graph as a spatial-temporal graph, where each individual stock (company) was considered a node. The node features represented the current state of each company in response to its price movement, and this state varied over time. Using HATS, important information from various relational data was selectively aggregated to represent the company as a node. The model was then trained to learn the interrelation between nodes before being fed into a task-specific layer for prediction. Table 2.4 provides a comprehensive summary of the reviewed machine learning approaches.

Table 2.4 A summary of neural network approaches on solving multicollinearities.

Author	Objective	Method
Huynh and Won (2011)	Multi-objective optimization function to minimize error	Regularized OS-ELM algorithm
Garg and Tai (2012)	Hybrid method of PCA and ANN	Factor analysis-artificial neural network (FA-ANN)
Ye et al. (2013)	Input weight that changes with time	OS-ELM Time-varying (OS-ELM-TV)
Gu et al. (2014)	Penalty factor in the weight adjustment matrix	Timeliness Online Sequential ELM algorithm
Guo et al. (2014)	Smoothing parameter to adjust output weight	Least Squares Incremental ELM algorithm
Hoseinzade and Haratizadeh (2019)	Model the correlation among different features from a diverse set of inputs	CNN-pred
Kim and Kim (2019)	Using features from different representation of same data to predicting the stock movement	LSTM-CNN
Nobrega and Oliveira (2019)	Kalman filter to adjust output weight	Kalman Learning Machine (KLM)
Hua (2020)	Decision tree to select features	XGBoost
Obite et al. (2020)	Compare ANN and OLSR in presence of multicollinearity	Artificial neural network
Zhang et al. (2021)	Applied attention to capture the intraday interaction between input features	CNN-deep factorization machine and attention mechanism (FA-CNN)
Mahadi et al. (2022)	Regularization parameter that varies with time	Regularized Recursive least-squares

The findings from the literature indicate that feature selection leads to the removal of variables and a reduction in information gain. The optimization of multicollinearity measures is subjective, and there is no guarantee of finding the global minimum. Moreover, each method exhibits inconsistent performance depending on the data and may not be applicable to every problem. On the other hand, hybrid or ensemble methods show promising performance and have the potential to improve financial forecasting by combining the strengths of filter, wrapper, and embedded methods. This research suggests that the concepts of relevancy and redundancy from feature selection can be adopted. The attention layer and the predictive model can be constructed in a way that learns both relevance and redundancies. The literature review also demonstrates that deep learning algorithms outperform simple OLS estimators in fitting data with multicollinearity. These algorithms do not require prior knowledge of the data relationship or distribution. This served as motivation to use LSTM as the predictive model in our research.

2.3. Proposed Attention and correlation embedding

This research draws inspiration from the concept of relevance and redundancy. Senawi et al. (2017) and Katrutsa and Strijov (2017) utilized this concept in their attempts to perform feature selection. Relevancy refers to the dependency between variables and the target feature, while redundancy refers to the dependence between variables. The objective of the algorithm was to select features that maximize relevancy and minimize redundancy. In this study, a different measure of relevance was proposed based on the attention mechanism introduced by Bahdanau et al. (2014). According to Zhang et al. (2018), the attention mechanism simulates visual attention, where humans adjust their focal point over time to perceive a "high resolution" when focusing on a particular region of an image but perceive a "low resolution" for the surrounding image. Similarly, the attention mechanism enables the model to learn how to assign different weights to features based on their relevance to the target feature, potentially capturing asymmetric influences and mitigating the multicollinearity problem.

However, the attention mechanism alone is not sufficient for the research objective. It reduces the amount of information available to the prediction model and overlooks feature interactions. Feature interactions are crucial because even if two variables have an 85% correlation, removing one variable would eliminate the potential 15% marginal predictive value. Therefore, this study used correlation as a measure of redundancy, as done in the literature, with the motivation that correlation can capture feature interactions. Additionally, deep learning was employed to transform the correlations into representations that contain essential information. These two components are referred to as the Multicollinearity Reduction Model (MRM), which is expected to enhance the prediction reliability of LSTM.

2.4. Regression or Classification

In algorithmic trading, a forecast is required to generate a trade signal, which can take the form of a price level or a trend direction. There have been numerous studies exploring both methods, each with its own benefits. However, the models, input data, and justifications proposed in these studies vary. Point estimation involves approximating the value of certain parameters, such as the future price of an asset. While financial forecasting is often formulated as a point estimation problem, there are instances where an alternative approach, known as interval-based estimation, is more suitable. Interval-based estimation focuses on estimating the range of values for the parameters, such as the range of future prices of an asset. This subtle change in framing a financial forecasting problem as an interval-based estimation problem can transform an unreliable forecast into a useful one and, in some cases, overcome the inherent limitations of point estimation, such as multicollinearity.

Several papers have approached financial forecasting as a regression problem. For instance, Jasic and Wood (2004) attempted a univariate approach to predicting stock market indices. They utilized a neural network to predict the short-term returns of four major indices: S&P 500, DAX, TOPIX, and the FTSE. Positive predicted returns generated a buy signal, while negative predicted returns indicated a sell signal. Their strategy yielded significantly higher profits compared to a buy-and-hold strategy. Similarly, Yong et al. (2017) employed regression in their stock trading system for the Singapore stock market. They utilized a deep neural network and the historical price of the FTSE Straits Time Index (STI) as independent variables. Their trading rules were based on the predicted closing price for the next two days, resulting in a profitable rate of 70.83% for their trades.

Wang and Mishra (2018) also designed a stock trading system using regression. They forecasted a value and established buy and sell rules based on the predicted value. Their experiment focused on the Taiwan Capitalization Weighted Stock Index (TAIEX). Chen et al. (2021) employed a LSTM network to forecast the stock price of a company listed on the NASDAQ stock exchange (Intel Corporation). They used three different types of input variables: daily open, high, low, and close prices; technical indicators of the stock; and various broad market indices. Their trading strategy relied on predicting the next day's price, buying a share if it was higher than the current closing price and selling a share if it was lower. The results demonstrated that the return of the LSTM model outperformed both Locally Weighted Regression and a buy-and-hold strategy.

Some researchers chose a classification approach. Mingyue et al. (2016) argued that predicting the daily return of a stock market index is challenging, and it is more practical for traders to predict the direction of movement for making buy or sell decisions. They utilized an ANN model to predict the next day's direction of the Nikkei 225 index, using various technical indicators computed from the index data. Their model accurately predicted the daily direction of the index in 86.39% of the trials. Zhong and Enke (2017) employed a classification model to investigate whether high predictability leads to high trading returns. They used an ANN to forecast the daily direction of the S&P 500 index, incorporating 60 financial and economic indicators as features and performing dimensionality reduction. Their approach achieved significantly higher risk-adjusted profits compared to the benchmark buy-and-hold strategy and the Treasury bill strategy.

Chen and Hao (2020) employed a support vector machine to develop a stock trading signals framework. They randomly selected 30 stocks from the Shanghai and Shenzhen stock exchanges and used the daily opening price, lowest price, highest price, closing price, and trading volume as input variables. The framework categorized the signals into four classes: strong buy, ordinary buy, ordinary sell, and strong sell. Vo and Yost-Bremm (2020) focused on cryptocurrencies, specifically Bitcoin, and developed a trading strategy. They utilized Bitcoin trading data from six different exchanges and created five technical indicators using price and volume data. Their approach involved using a Random Forest machine learning algorithm with a trading horizon of 15 minutes, generating categorical buy and sell signals.

Several studies have compared the performance of point estimation with classification-based prediction and have suggested that classification-based prediction outperforms point estimation in the aforementioned financial forecasting problems. Leung et al. (2000) conducted empirical experiments indicating that classification models outperformed point estimation models in terms of predicting the direction of stock market movements and maximizing investment trading returns. The classification models employed included discriminant analysis, logit, probit, and probabilistic neural network, while the regression models included adaptive exponential smoothing, vector autoregression with Kalman filter, multivariate transfer function, and multilayered feedforward neural network. The study focused on forecasting three major global broad market indices: S&P 500, FTSE 100, and Nikkei 225.

Olson and Mossman (2003) also found that classification-based models outperformed point estimation models in predicting whether 2352 firms in Canada would have high or low returns based on 61 accounting ratios. Enke and Thawornwong (2005) discovered that classification performed better than regression in terms of profitability while maintaining the same level of risk exposure. Their study focused on the S&P 500 index and utilized fundamental economic data such as interest rates, treasury rates, industrial production, and inflation rates. Trading based on neural network classification forecasts yielded higher profits compared to the buy-and-hold strategy, neural network regression forecasts, and linear regression. The authors suggested that classification-guided trading could enhance profitability by generating trade signals only when significant price changes occurred.

According to Leung et al. (2000) and Olson and Mossman (2003), another important factor contributing to the superior performance of classification models over point-based models in terms of translating financial forecasts into profitability is the ability to enhance profitability through thresholding the outputs of the classification neural network. Thresholding allows researchers to filter out weak forecasts, which is crucial for profitability since a trader's investment capital is at risk only when a trade is made. Traders can avoid risk and losses by disregarding poor forecasts. Currently, researchers are adopting this paradigm shift in financial forecasting and experimenting with new approaches to classification-based financial forecasting, including formulating innovative forecasting problems, working with high-dimensional data and output, and optimizing classification-based models.

2.5. Conclusion

Each of these factors in financial forecasting is interconnected, and even minor changes can have a significant impact on forecast performance. The elimination of multicollinearity can be approached through two methods: variable selection and modified estimates. Variable selection methods aim to reduce the number of variables to a select few that are deemed most relevant, although this reduction may result in a decrease in information gain due to the reduced dataset. On the other hand, modified estimators focus on enhancing parameter estimation efficiency in the presence of multicollinearity, but this improvement comes with a trade-off between bias and variance. Furthermore, the literature review demonstrates that deep learning algorithms outperform the simple OLS estimator when fitting data with multicollinearity.

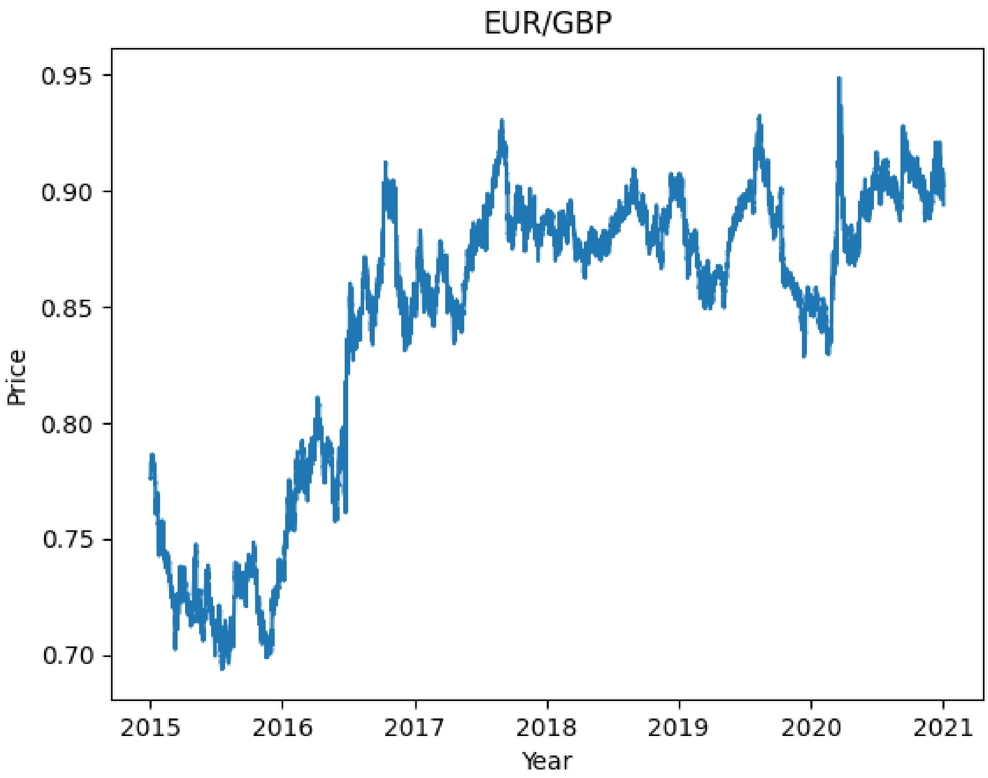
In this study, the concept of relevance and redundancy serves as inspiration. The attention mechanism is employed to represent relevance, while correlation is used to capture redundancy. Additionally, there has been a shift from point estimation to classification-based models in financial forecasting. The present research investigates the impact of this shift on mitigating multicollinearity.

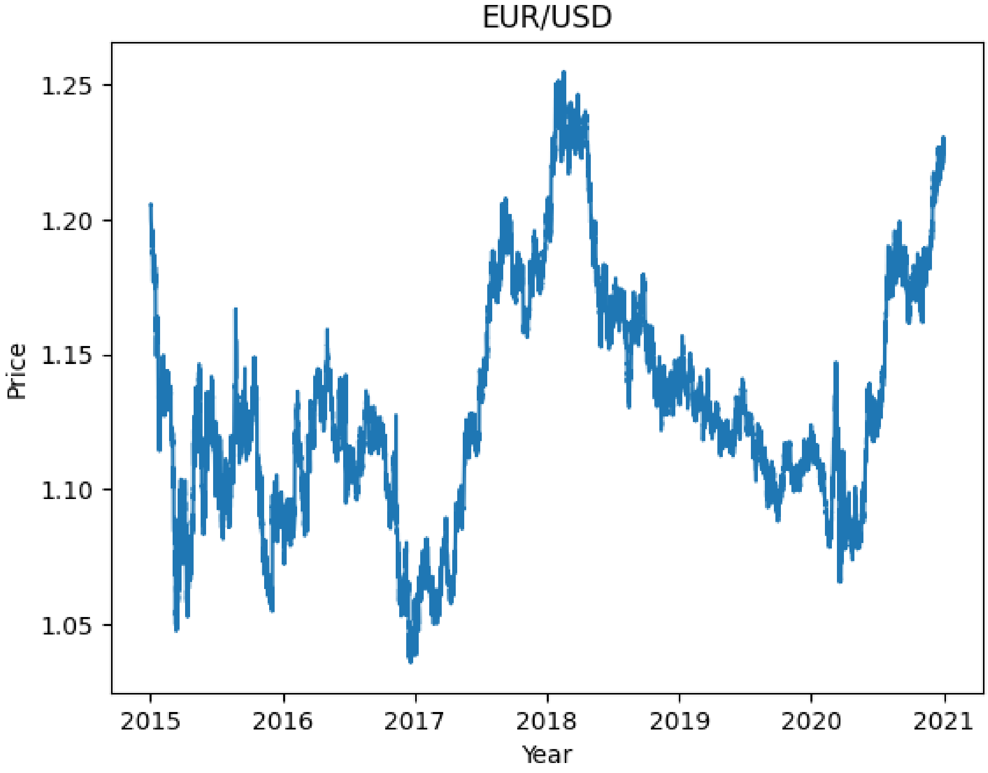
CHAPTER 3

METHODOLOGY

3.1. Data Collection

The data used in this research was the daily foreign exchange rate. Four different pairs of exchange rates were selected for this study: EUR/GBP, EUR/USD, GBP/USD, and NZD/USD. The first three pairs were chosen because they had the highest trading volume and trading data (Cook, 2021). A higher trading volume was considered more desirable for algorithmic trading. The last pair of exchange rates was included to introduce diversity in the data. The data was sourced from TradingView. The timeframe for the data collection was from 1 January 2015 to 31 December 2020, covering a period of 6 years. A model testing period of 20% of the data was used. The exchange rate was collected at an hourly interval. Figure 3.1 displayed the plot of the hourly closing price for each foreign exchange pair.


EUR/GBP
Price
0.95
0.90
0.85
0.80
0.75
0.70
2015
2016
2017
2018
2019
2020
2021
Year


EUR/USD
Price
1.25
1.20
1.15
1.10
1.05
2015
2016
2017
2018
2019
2020
2021
Year

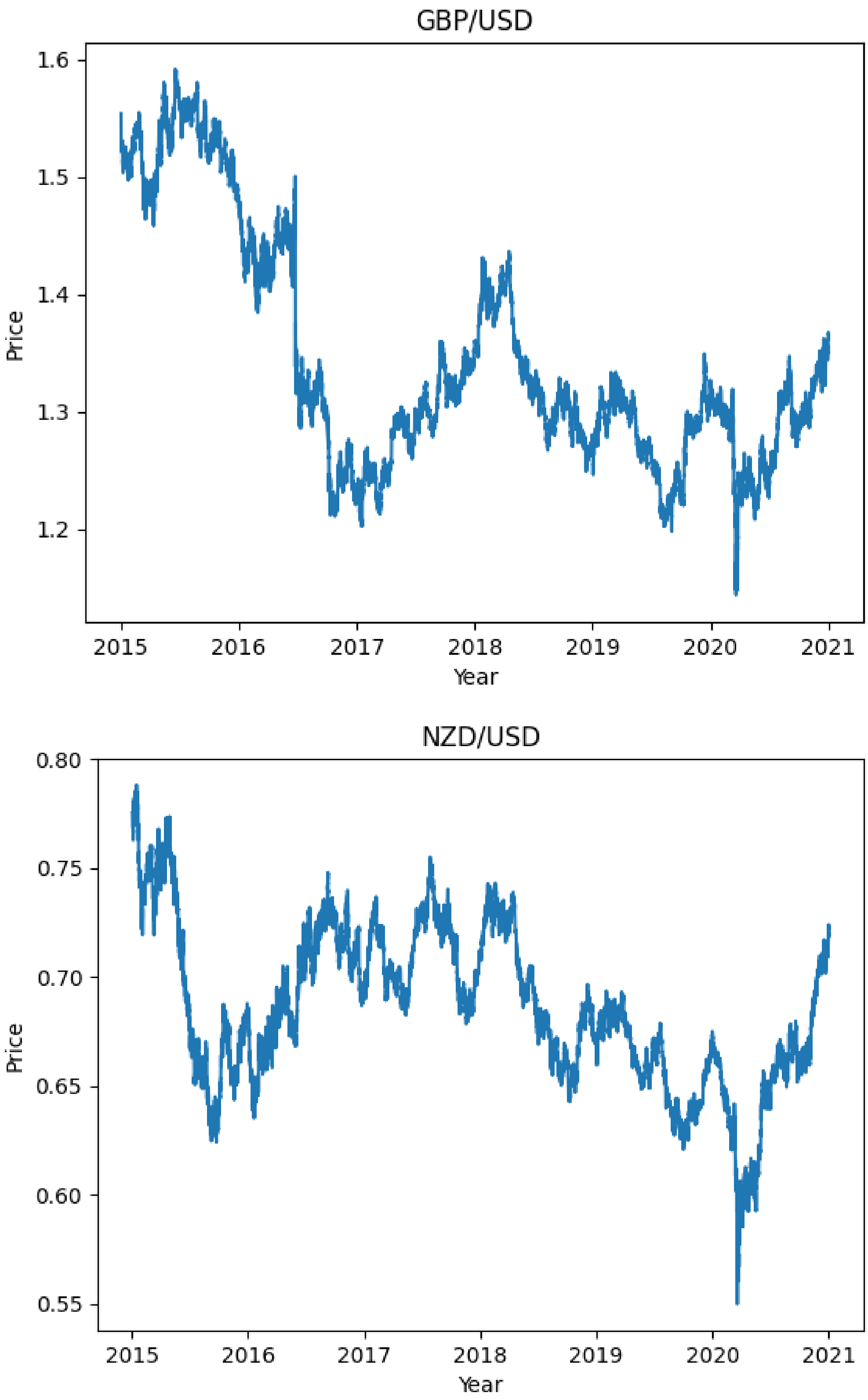


Figure 3.1 Price chart for each foreign exchange rate from 1 January 2015 to 31 December 2020

In analyzing the plot of EUR/GBP exchange rate over the past years, we observed two price ranges. Initially it ranged from 0.7 to 0.8 in the year 2015. After a bullish trend in 2017, the rate exhibited a consolidation phase. Notable support and resistance levels can be identified, with the price consistently finding support around the 0.85 level and encountering resistance near 0.93.

The plot of EUR/USD exchange rate over the past six years reveals a neutral trend. During the first two years, the price exhibited a relatively stable sideways movement, indicating a neutral trend with minor fluctuations within a narrow range. However, in the year 2017, there was a bullish trend, reaching its highest rate of 1.25. Following the peak, the rate experienced a drop in the subsequent years until 2020. Since then, it has slowly begun an upward trajectory once again to levels near its peak.

The plot of GBP/USD exchange rate over the past year depicts a clear downtrend followed by a period of consolidation. In the initial 2 years, the price exhibited a consistent downward movement, with lower highs and lower lows. However, starting around 2017, the price entered a phase of consolidation. This consolidation phase can be observed by the price oscillating between a defined 1.2 and 1.4. The exchange rate remained range-bound, with the upper and lower boundaries acting as barriers to further price movement. The highest rate during this period is in early 2018.

The plot of NZD/USD exchange rate over the past years reveals a predominantly range-bound pattern, with one notable exception in the year 2020. For the majority of the period, the price exhibited a relatively stable sideways movement within a defined range. This range-bound behavior indicates a lack of significant upward or downward trends. However, in the year 2020, the rate experienced a sharp decline to 0.55, deviating significantly from its typical range-bound pattern.

Table 3.1 Descriptive Statistics for each dataset.

Currency Pair	Mean	Median	Standard Deviation	Minimum	Maximum
EUR/GBP	0.8456	0.8715	0.0634	0.6937	0.9488
EUR/USD	1.1314	1.1246	0.0435	1.0356	1.2549
GBP/USD	1.3445	1.3102	0.0999	1.1434	1.5918
NZD/USD	0.6849	0.6825	0.0369	0.5498	0.7881

The descriptive statistics of each dataset were presented in Table 3.1. The currency pairs provided exhibited different characteristics for trading. EUR/GBP displayed moderate volatility with a distribution that was slightly left-skewed. EUR/USD presented a more stable trading environment, characterized by lower volatility and a narrower range of 1.0356 to 1.2549. GBP/USD demonstrated higher volatility and slightly right-skewed distribution, with a wider range of 1.1434 to 1.5918, making it suitable for traders who were comfortable with increased risk. NZD/USD offered moderate volatility similar to EUR/USD. It is important to note that these statistics did not represent the testing set used in the result presentation of this research. The test set consisted only of the last 20% of the dataset.

3.2. Predictor Analysis

The predictor variables consisted of highly correlated technical analysis indicators. Previous studies (Sezer et al., 2017) had demonstrated that neural networks utilizing these technical indicators exhibited predictive capabilities comparable to a buy-and-hold strategy. The underlying principle of technical analysis was that prices incorporated all pertinent information and that price movements could be predicted based on historical asset prices and volume trends (Nuti et al., 2011). The proposed LSTM model employed the following nine technical indicators as features.

The RSI (Relative Strength Index) is a momentum indicator used to measure the velocity and magnitude of price movements. It ranges from 0 to 100. When the RSI is above 70, it suggests that the price is overbought, meaning it may be due for a downward correction. Conversely, when the RSI is below 30, it suggests that the price is oversold, indicating a potential upward correction. The RSI is calculated by comparing the average gains and losses over a specific period. The indicator is computed as follows

$$RSI = 100 - \frac{100}{1 + \frac{Average\ Gain}{Average\ Loss}}, \tag{9}$$

The average gain represents the average price increase during that period, while the average loss represents the average price decrease. The relative strength is then calculated by dividing the average gain by the average loss.

The MACD (Moving Average Convergence Divergence) indicator consists of two lines: the MACD line and the signal line. The MACD line is calculated by subtracting the value of a longer-term exponential moving average (EMA) from a shorter-term EMA. The signal line, on the other hand, is an EMA of the MACD line itself. Interpreting the MACD involves observing the crossovers between the MACD line and the signal line.

$$MACD\ Line: (12\ Days\ EMA - 26\ days\ EMA) \tag{10}$$

$$Signal\ Line: 9\ days\ EMA\ of\ MACD\ line \tag{11}$$

When the MACD line crosses above the signal line, it is considered a bullish signal, indicating a potential uptrend. Conversely, when the MACD line crosses below the signal line, it is seen as a bearish signal, suggesting a potential downtrend.

The Parabolic Stop and Reverse (SAR) is a popular technical analysis indicator used by traders to determine potential stop-loss levels and signal reversals in the price trend of an asset. It aims to provide trailing stop-loss levels that adjust dynamically as the price moves in a trend. Generally, the SAR below price is bullish and the SAR above is bearish. The formula is as below:

$$SAR_{n+1} = SAR_n + \alpha(EP - SAR_n)\,, \tag{12}$$

where EP, the extreme point, is the highest or lowest value achieved by an uptrend or downtrend in a period n. α represents the acceleration factor and determines the rate at which the stop-loss level moves closer to the price. It is initiated with 0.02. Each time a new EP is recorded, the factor increases by 0.02, and thus the SAR will converge to the price faster. This 0.02 value is not based on any fixed mathematical or fundamental principle, but rather it has been chosen empirically as a starting point that tends to work well for various market conditions. (Treleaven et al., 2013)

The SMA is an unweighted moving average of the closing price of the previous n days. By taking the sum of the closing prices of the previous n days and dividing it by n, the SMA offers a single data point that represents the average price over the selected time frame. The number n can be selected depending on the period of trend desired, for example, short term, medium-term, and long-term trends. It is calculated with the formula:

$$SMA_n = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n}, \tag{13}$$

where x_n is the price at period n and n is the total number of periods. The SMA line can be used as the support or resistance level of the stock or used in conjunction with the SMA line of different periods to determine if it is on an uptrend or downtrend. For example, the stock price above the SMA indicated an uptrend and, therefore, a buy signal.

The CMA (Cumulative Moving Average), also known as the running average, is a calculation method that considers all the data points accumulated up to a specific point in time. It provides a continuous average by incorporating the historical data into the calculation. The advantage of using the CMA is its ability to smoothen short-term fluctuations and reveal long-term trends in a dataset. Unlike other moving average methods that use a fixed window of data, the CMA is updated incrementally with each new data point, ensuring that the average reflects the entire dataset. It is calculated with the formula:

$$CMA_n = \frac{x_n + x_{n-1} * (CMA_{n-1})}{4}, \tag{14}$$

where x_n is the price at period n. Traders and analysts often employ the CMA to gain insights into the overall direction of the data and make informed decisions based on the trend revealed by the continuous average. By considering the entire dataset, the CMA can provide a more comprehensive understanding of the data's behavior and help identify significant changes or turning points.

The EMA (Exponential Moving Average) is a variation of the moving average calculation that incorporates a weighting scheme, giving more significance to recent data points while gradually reducing the influence of older data. Unlike the simple moving average, which assigns equal weights to all data points, the EMA places the highest weight on the most recent observation. This weighting is determined by the parameter α, which controls the rate at which the weights decrease exponentially. It is calculated with the formula below:

$$S_t = \alpha \times Y_t + (1 + \alpha) \times S_{t-1}, \tag{15}$$

where α is the parameter for the degree of decrease in weight and Y_t is the observation at time t. By adjusting the value of α, traders and analysts can tailor the EMA to different timeframes or sensitivity requirements. (Nuti et al., 2011).

The stochastic oscillator is a momentum indicator that measures the current price's position within a recent price range. It compares the closing price to the highest and lowest prices observed over a specific period. The oscillator is calculated using two main components: %K and %D.

$$\%K = 100 \frac{closing\ price - L}{H - L}, \tag{16}$$

$$\%D = 3\, period\ moving\ average\ of\ \%K\ , \tag{17}$$

%K represents the current price's percentage in relation to the range between the lowest (L) and highest (H) prices. %D, on the other hand, is a three-period moving average of %K. Traders often look for trading signals when the %K and %D lines cross. If %K crosses above %D, it generates a bullish signal, indicating a potential buying opportunity, while a crossover below %D generates a bearish signal for a potential selling opportunity. The intuition is that prices tend to approach the extremes of the range before turning. The stochastic oscillator is commonly used to identify overbought and oversold market conditions, with readings above 80 suggesting overbought and readings below 20 indicating oversold conditions. (Treleaven et al., 2013)

The Williams %R, also referred to as the Williams Percent Range, is a widely used oscillator in technical analysis that provides insights into the proximity of the asset price to recent highs or lows. It is expressed as a value ranging from -100 to 0. When the %R value reaches -100, it signifies that the closing price is at or near the lowest point observed over the past n days, indicating potential oversold conditions. This suggests that the price may have reached a level where selling pressure has pushed it to an extreme, and a price reversal or upward correction might be anticipated. The formula is shown below:

$$\%R = \frac{(high)\ - close}{-min\ (low)} * -100\ , \tag{18}$$

By monitoring the %R indicator, traders and analysts can gain valuable information about the current strength or weakness of a stock's price and identify potential trading opportunities.

The Bollinger Bands is a popular technical analysis tool used to identify potential price levels of support and resistance based on volatility. It consists of an upper band, a lower band, and a middle band, which is typically a simple moving average. The upper band is calculated by adding a specified number of standard deviations to the middle band, while the lower band is calculated by subtracting the same number of standard deviations. The Bollinger Bands help traders assess whether the price is approaching extreme levels, indicating a potential reversal in trend. The formula is as follows:

$$Bollinger\ High = MA_n + 2 * \sigma_n \ , \tag{19}$$

$$Bollinger\ Low = MA_n + 2 * \sigma_n \ , \tag{20}$$

where MA is the simple moving average, σ is the standard deviation, and n is the number of days in the smoothing period. When the price nears the upper band, it suggests that the market may be overbought, potentially signaling a trend reversal or a pullback. On the other hand, when the price approaches the lower band, it indicates that the market may be oversold, suggesting a possible buying opportunity or an upward reversal.

3.3. Data Generation

The samples were constructed using a rolling window mechanism. Figure 3.2 illustrated an example of this mechanism with a time series consisting of 10-time steps. With a window size of 3-time steps, the first row of data represented the period from t = 1 to t = 3, the second row represented t = 2 to t = 4, and the third row represented t = 3 to t = 5. By applying this mechanism, a total of eight rows of data were computed. For this research, a lag of 10 trading days (window size) was utilized.

The target label denoted a classification of either profit or loss. This label was determined based on the subsequent five trading days using the following criteria: If the price reached a predetermined take profit price level within five trading days, the label was assigned as 1 (significant profit). Conversely, if the price reached the stop loss level, the label was assigned as 0 (significant loss). If, after five trading days, the price fell between the two price levels, it was labeled as 3 (profit) if the trade was still profitable, and 2 (loss) if it resulted in a loss. The labeling methodology was depicted in Figure 3.3.

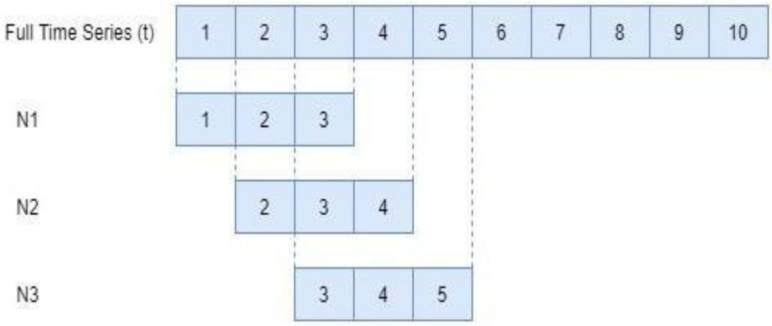


Figure 3.2 Illustration of the rolling window mechanism.

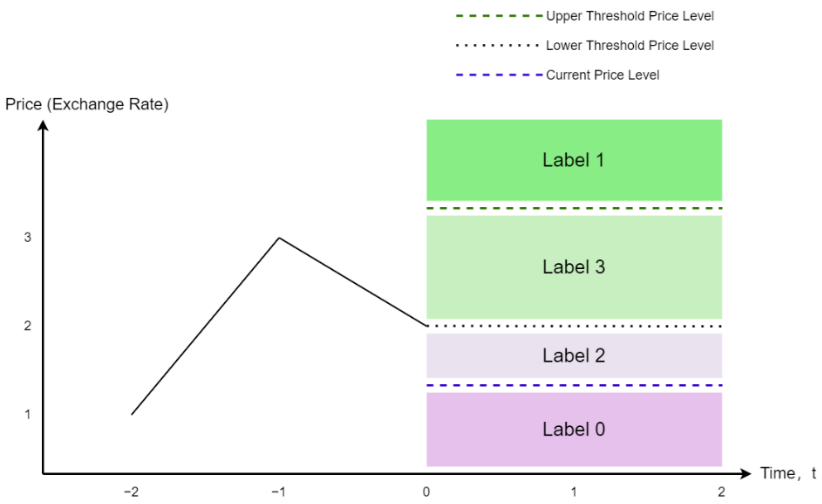


Figure 3.3 Illustration of dataset labeling methodology.

3.4. Model Framework

The historical prices of foreign exchanges utilized in this research were also sequential. The LSTM network had the ability to learn potential temporal information from this sequential data. The LSTM network had demonstrated outstanding performance in numerous real-world applications involving sequential data. Hence, an LSTM model served as an appropriate baseline model for this study. The LSTM, being a type of RNN, was designed to address the issue of exploding and vanishing gradients that hindered the performance of RNNs in long-sequence data (Althelaya et al., 2018). Unlike an RNN, the LSTM model incorporated memory gates that captured and retained information from long time lags while selectively discarding stored information.

Figure 3.4 depicted the cell of an LSTM network. The top horizontal line represented the cell state. The four neural net layers served as gates that controlled the addition or removal of information. At time lag t, the first sigmoid (σ) layer determined what information to forget. It took into account the previous hidden state h_{t-1} and the current input x_t to generate an output value between 0 and 1. The hidden state h_{t-1} represented the encoded input from the previous time step, while x_t represented the input at time step t, encompassing all features. A sigmoid output value of zero indicated complete forgetting, while a value of one indicated complete retention. The subsequent sigmoid layer, known as the input gate, determined which value to update. A subsequent tanh layer generated new values to be added to the cell state. These two values were combined through pointwise multiplication to update the cell state. Finally, a sigmoid layer determined which part of the cell state to output.

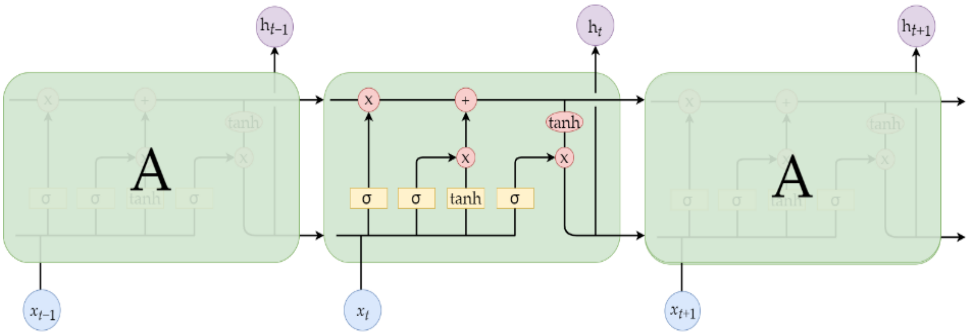


Figure 3.4 LSTM module diagram.

The proposed model employed the attention mechanism, which had been originally developed for neural machine translation by Bahdanau et al. (2014). This mechanism enabled the model to search for segments of a source sentence that were most relevant to the target word. Additionally, attention weighting aligned with the author's intuition. In this study, the attention mechanism was utilized to determine the relevance of each feature to the target variable. For instance, one Moving Average indicator might hold more relevance to the target compared to another indicator due to the different time frames used. The attention module consisted of a linear layer, dropout layer, sigmoid activation, another linear layer, and a SoftMax layer. The resulting weight from the module represented a learned measure of importance for each feature. Finally, the SoftMax function was applied to ensure that the weightings summed up to one.

Moving on, the subsequent part of the model involved correlation embedding. This section elaborated on how the correlation embeddings were derived. In this study, correlation embeddings were employed as a proxy for redundant information between features. A correlation quantified the strength of the relationship between the relative movements of two variables. The calculation of correlation was performed using the formula below:

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}}, \tag{21}$$

The correlation coefficient R_{ij} between x_i and x_j, as well as the covariance matrix C_{ij} of x_i and x_j, the variance C_{ii} of x_i, and the variance C_{ij} of x_j were calculated. These computations yielded a correlation matrix, where the values ranged from -1.0 to 1.0. A correlation of -1.0 indicated a perfect negative correlation, while a correlation of 1.0 indicated a perfect positive correlation. For instance, the Bollinger High and Bollinger Low exhibited high collinearity due to their association as the upper and lower price bands. Subsequently, the correlation matrix was passed through a neural network layer with an output size matching the input data. The resulting output was referred to as the correlation embedding.

The overall structure of the proposed model, known as the Multicollinearity Reduction Model (MRM), was depicted in Figure 3.5. This model received an input in the form of a batch, sequence length, and input size denoted as X_t. The input data was then fed into the input attention module, which generated weights for each feature. These weights were multiplied by the input data to obtain the weighted input data. Next, the weighted input data was multiplied by the correlation matrix before being passed into the LSTM layer. This correlation embedding, represented as *cr* in the diagram, provided the LSTM layer with information regarding the relevance of each feature and the redundancy between features. The underlying idea was that the model could make predictions using the learned attention and correlation information of the features, eliminating the need to discard features for achieving satisfactory results.

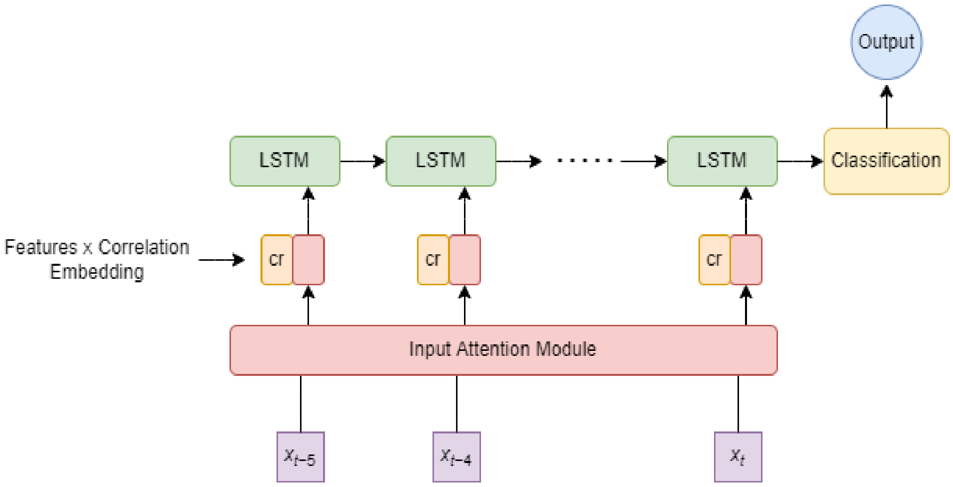


Figure 3.5 Proposed Multicollinearity Reduction Model Framework.

The final output of the LSTM was fed into the classification network, which consisted of two additional linear layers. The resulting output provided predictions for the classes. Both the LSTM layer and the classification layer served as the baseline model for this study. Previous work by Chan et al. (2022)

demonstrated the effectiveness of the proposed methodology in classification. In this research, we aimed to investigate its effectiveness in regression tasks.

3.5. Performance Measure

In the context of financial forecasting, this section presented the appropriate performance measure for the research. A conventional measure of performance involves assessing forecast error and accuracy. Forecast error represents the disparity between the actual and predicted outcomes. However, when comparing regression and classification in financial forecasting, the different forecast errors used (such as root mean squared error and cross entropy loss) do not allow for a proper comparison.

RMSE is a widely used metric for assessing the accuracy of predictive models. It measures the average difference between the predicted values and the actual values in a dataset. By taking the square root of the mean of the squared differences, RMSE provides a single value that represents the overall performance of a model. Researchers often employ RMSE to evaluate regression models and quantify the extent of error between predicted and observed values.

On the other hand, Cross Entropy Loss serves as a crucial measure for evaluating classification models. It quantifies the dissimilarity between the predicted probability distribution and the true probability distribution of classes in a classification task. By calculating the negative logarithm of the predicted probabilities of the correct classes, Cross Entropy Loss enables the assessment of how well a model distinguishes between different classes.

Accuracy, commonly used as a performance metric, is easy to calculate and understand. To calculate the accuracy, the model's predictions are compared to the true class labels of the dataset. For each instance, if the predicted class matches the true class, it is considered a correct prediction. Conversely, if the predicted class differs from the true class, it is deemed an incorrect prediction. Nevertheless, its reliability diminishes when dealing with imbalanced datasets. A predictive model with lower accuracy may possess better prediction power than a model with higher accuracy (Valverde-Albacete & Peláez-Moreno, 2014), which is known as the accuracy paradox. In financial forecasting, datasets often exhibit skewed class distributions, with the minority class (significant profit) being of particular interest. Therefore, accuracy becomes problematic in these settings.

Moreover, higher accuracy does not necessarily correspond to higher profits. This inconsistency is referred to as profit bias by Liu and Wang (2019). It arises from trades not having the same magnitude of return. The profit factor, which considers both accuracy and the ratio between average win and average loss (Harris, 2008), becomes a crucial factor. Accuracy alone does not suffice to generate positive returns. Thus, in this study, the focus is on the profits generated from trading to provide a clearer reflection of predictive performance. This is often measured in pips, which stands for "percentage in point" and represents the smallest unit of price movement in the forex market.

3.6. Conclusion

The data used in this research consisted of daily foreign exchange rates. The VIF diagnosis test indicated a high level of multicollinearity within the datasets. To predict the price changes for the next five trading days, a lag of 10 trading days (window size) was employed. The LSTM model was selected as the baseline model for the experiment. Attention mechanism was utilized in this study to determine the relevance of each feature to the target variable. Additionally, correlation was used as a measure of redundancy between features, providing valuable information about the strength of the relationship between the relative movements of two variables. Given the unique nature of financial forecasting, the evaluation metric utilized in this study is the profit generated from trading activities.

Figure 3.6 depicted the overall methodology employed in the research. The first step involved gathering raw data, which included retrieving historical foreign exchange prices. The subsequent step entailed preprocessing the data for analysis, encompassing tasks such as generating samples and target labels, feature engineering, and normalization. Lastly, the data was fed into the predictive model for forecasting and comparison with the baseline results.

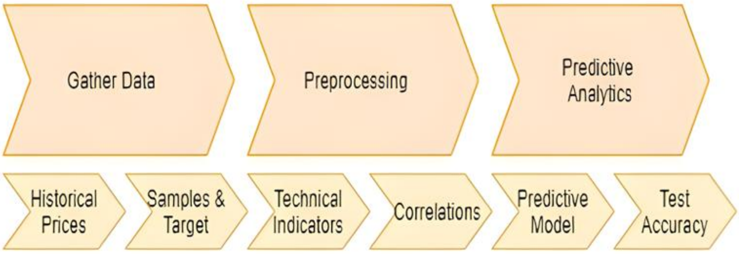


Figure 3.6 Overview of methodology for foreign exchange prediction based on technical indicators.

CHAPTER 4

PRESENTATION OF RESULT

4.1 Multicollinearity Analysis

In this study, a multicollinearity diagnosis was conducted on each variable to determine the level of dependencies between them. The Variation Inflation Factor (VIF) was utilized for this purpose. The VIF of the k-th variable was computed using the formula:

$$VIF_k = \frac{1}{1 - R_k^2} \tag{22}$$

To calculate the VIF of the k-th variable, the k-th variable was taken as the explained variable with all the remaining variables as predictors, and the regression coefficient was estimated. The coefficient of determination (R^2) of the regression was then obtained and used in the VIF formula. There is no specific threshold value for determining the presence of multicollinearity, but a value of 10 is often considered indicative of multicollinearity (Lavery et al., 2019). Therefore, a VIF value exceeding 10 suggests a significant presence of multicollinearity. The results for each variable in each dataset are presented in Table 4.1. The diagnosis revealed that the datasets exhibited a high degree of multicollinearity.

Table 4.1 Variation Inflation Factor Analysis for each dataset.

Variables	EUR/GBP	EUR/USD	GBP/USD	NZD/USD
Open	20,637	7,216	15,198	6,157
High	17,431	6,396	16,101	6,775
Low	20,848	7,283	11,619	5,430
Close	24,950	8,248	17,376	7,677
RSI	7	7	5	7
MACD	21	21	21	20
SAR	877	275	484	262
***SMA* 5**	13,692	4,056	9,220	4,588
***SMA* 10**	3,599	1,089	2,391	1,319
***SMA* 20**	243,345	51,086	233,443	60,390
CMA	4	1	3	2
EMA	67,907	20,013	44,480	22,461
%*K*	32	28	30	36
%*D*	13	14	13	16
%*R*	19	15	18	21
Bollinger High	61,274	13,145	62,736	16,354
Bollinger Low	62,111	13,772	57,264	14,816

In addition, Table 4.2, Table 4.3, Table 4.4 and Table 4.5 shows the correlation coefficient of each foreign exchange pairs. Correlation ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation. If the absolute value of the correlation coefficient is greater than 0.7, it indicates a strong relationship between the variables. It's important to note that correlation does not imply causation. Even if two variables are strongly correlated, it does not necessarily mean that one variable causes the other to change.

Table 4.2 Correlation coefficient for the EUR/GBP foreign exchange.

	Open	High	Low	Close	RSI	MACD	SAR	5 days SMA	10 days SMA	20 days SMA	CMA	EMA	%K	%D	%R	Bollinger High	Bollinger Low
Open	1.00																
High	1.00	1.00															
Low	1.00	1.00	1.00														
Close	1.00	1.00	1.00	1.00													
RSI	0.05	0.05	0.05	0.05	1.00												
MACD	0.04	0.04	0.04	0.04	0.80	1.00											
SAR	1.00	1.00	1.00	1.00	0.02	0.02	1.00										
5 days SMA	1.00	1.00	1.00	1.00	-0.01	-0.03	1.00	1.00									
10 days SMA	0.99	0.99	0.99	0.99	-0.02	-0.04	0.99	1.00	1.00								
20 days SMA	0.98	0.98	0.98	0.98	-0.02	-0.05	0.98	0.99	1.00	1.00							
CMA	0.83	0.83	0.83	0.83	-0.01	-0.03	0.83	0.83	0.84	0.85	1.00						
EMA	1.00	1.00	1.00	1.00	0.00	-0.02	1.00	1.00	1.00	0.99	0.83	1.00					
%K	0.13	0.13	0.13	0.13	0.50	0.51	0.11	0.06	0.02	-0.03	-0.04	0.07	1.00				
%D	0.14	0.14	0.13	0.14	0.09	0.14	0.13	0.10	0.05	-0.02	-0.04	0.10	0.85	1.00			
%R	0.12	0.12	0.12	0.12	0.57	0.58	0.10	0.05	0.00	-0.03	-0.03	0.05	0.95	0.73	1.00		
Bollinger High	0.98	0.98	0.98	0.98	-0.02	-0.05	0.98	0.99	0.99	0.99	0.83	0.99	-0.03	-0.01	-0.03	1.00	
Bollinger Low	0.97	0.97	0.97	0.97	-0.02	-0.05	0.97	0.98	0.98	0.99	0.86	0.98	-0.03	-0.02	-0.03	0.97	1.00

Table 4.3 Correlation coefficient for the EUR/USD foreign exchange.

	Open	High	Low	Close	RSI	MACD	SAR	5 days SMA	10 days SMA	20 days SMA	CMA	EMA	%K	%D	%R	Bollinger High	Bollinger Low
Open	1.00																
High	1.00	1.00															
Low	1.00	1.00	1.00														
Close	1.00	1.00	1.00	1.00													
RSI	0.08	0.08	0.08	0.08	1.00												
MACD	0.09	0.09	0.09	0.09	0.81	1.00											
SAR	1.00	1.00	1.00	1.00	0.02	0.05	1.00										
5 days SMA	0.99	0.99	0.99	0.99	-0.03	-0.05	0.99	1.00									
10 days SMA	0.97	0.97	0.97	0.97	-0.03	-0.05	0.97	0.99	1.00								
20 days SMA	0.95	0.95	0.95	0.95	-0.03	-0.04	0.95	0.97	0.99	1.00							
CMA	0.22	0.22	0.22	0.22	-0.06	-0.08	0.22	0.25	0.26	0.29	1.00						
EMA	0.99	0.99	0.99	0.99	-0.02	-0.04	0.99	1.00	0.99	0.97	0.25	1.00					
%K	0.24	0.24	0.24	0.24	0.49	0.51	0.22	0.12	0.04	-0.03	-0.13	0.13	1.00				
%D	0.25	0.25	0.25	0.25	0.06	0.11	0.25	0.19	0.09	0.00	-0.13	0.19	0.85	1.00			
%R	0.21	0.21	0.21	0.21	0.55	0.58	0.19	0.08	0.01	-0.03	-0.11	0.09	0.94	0.71	1.00		
Bollinger High	0.93	0.93	0.93	0.93	-0.02	-0.05	0.93	0.95	0.96	0.98	0.26	0.95	-0.02	0.01	-0.02	1.00	
Bollinger Low	0.93	0.93	0.93	0.93	-0.03	-0.04	0.93	0.95	0.96	0.98	0.31	0.95	-0.03	-0.01	-0.03	0.91	1.00

Table 4.4 Correlation coefficient for the GBP/USD foreign exchange.

	Open	High	Low	Close	RSI	MACD	SAR	5 days SMA	10 days SMA	20 days SMA	CMA	EMA	%K	%D	%R	Bollinger High	Bollinger Low
Open	1.00																
High	1.00	1.00															
Low	1.00	1.00	1.00														
Close	1.00	1.00	1.00	1.00													
RSI	0.02	0.02	0.02	0.02	1.00												
MACD	0.03	0.03	0.03	0.03	0.78	1.00											
SAR	1.00	1.00	1.00	1.00	-0.01	0.01	1.00										
5 days SMA	0.99	0.99	0.99	0.99	-0.04	-0.05	0.99	1.00									
10 days SMA	0.99	0.99	0.99	0.99	-0.05	-0.06	0.99	1.00	1.00								
20 days SMA	0.98	0.98	0.98	0.98	-0.04	-0.05	0.98	0.99	0.99	1.00							
CMA	0.77	0.77	0.77	0.77	-0.04	-0.04	0.77	0.78	0.78	0.79	1.00						
EMA	1.00	1.00	1.00	1.00	-0.04	-0.05	1.00	1.00	1.00	0.99	0.78	1.00					
%K	0.07	0.07	0.07	0.07	0.50	0.49	0.05	-0.01	-0.06	-0.10	-0.08	0.00	1.00				
%D	0.07	0.07	0.07	0.07	0.09	0.11	0.07	0.03	-0.03	-0.09	-0.09	0.03	0.85	1.00			
%R	0.06	0.06	0.06	0.06	0.57	0.56	0.04	-0.02	-0.07	-0.09	-0.07	-0.02	0.94	0.72	1.00		
Bollinger High	0.96	0.96	0.96	0.96	-0.04	-0.06	0.96	0.97	0.98	0.99	0.80	0.97	-0.10	-0.10	-0.09	1.00	
Bollinger Low	0.97	0.97	0.97	0.97	-0.04	-0.05	0.97	0.98	0.99	0.99	0.77	0.98	-0.09	-0.08	-0.08	0.95	1.00

Table 4.5 Correlation coefficient for the NZD/USD foreign exchange.

	Open	High	Low	Close	RSI	MACD	SAR	5 days SMA	10 days SMA	20 days SMA	CMA	EMA	%K	%D	%R	Bollinger High	Bollinger Low
Open	1.00																
High	1.00	1.00															
Low	1.00	1.00	1.00														
Close	1.00	1.00	1.00	1.00													
RSI	0.04	0.05	0.05	0.05	1.00												
MACD	0.06	0.06	0.06	0.06	0.82	1.00											
SAR	1.00	1.00	1.00	1.00	-0.01	0.02	1.00										
5 days SMA	0.99	0.99	0.99	0.99	-0.06	-0.07	0.99	1.00									
10 days SMA	0.98	0.98	0.97	0.97	-0.07	-0.08	0.98	0.99	1.00								
20 days SMA	0.95	0.95	0.95	0.95	-0.06	-0.07	0.95	0.97	0.99	1.00							
CMA	0.47	0.48	0.47	0.47	-0.08	-0.09	0.48	0.50	0.50	0.52	1.00						
EMA	0.99	0.99	0.99	0.99	-0.05	-0.06	0.99	1.00	0.99	0.97	0.50	1.00					
%K	0.12	0.12	0.12	0.12	0.50	0.53	0.10	0.00	-0.07	-0.14	-0.17	0.01	1.00				
%D	0.13	0.13	0.13	0.13	0.11	0.14	0.13	0.06	-0.03	-0.13	-0.17	0.06	0.86	1.00			
%R	0.10	0.10	0.10	0.10	0.56	0.58	0.08	-0.02	-0.10	-0.13	-0.15	-0.01	0.95	0.75	1.00		
Bollinger High	0.93	0.93	0.93	0.93	-0.06	-0.06	0.93	0.95	0.97	0.98	0.55	0.95	-0.14	-0.13	-0.13	1.00	
Bollinger Low	0.94	0.94	0.94	0.94	-0.06	-0.07	0.94	0.96	0.98	0.98	0.46	0.96	-0.13	-0.12	-0.13	0.93	1.00

4.2 Comparison between Neural Network and Statistical Methods

In this section, the LSTM model was compared to two non-neural network methods in order to establish the baseline model. The selected methods were stepwise regression and ridge regression, which are commonly used statistical approaches. The LSTM model, on the other hand, is a neural network specifically designed for sequential data, capable of capturing long-term information. Stepwise variable selection was employed as a procedure to sequentially add new variables to the model while allowing for the removal of variables at each stage. In this research, a p-value of 0.01 was used as the selection criteria for both adding and removing variables. A linear regression model was then constructed using the subset obtained from stepwise selection to make predictions on the foreign exchange data. Ridge regression, on the other hand, is a linear regression model that incorporates a penalty term in the loss function. The magnitude of the penalty term is controlled by the ridge parameter. For this experiment, a ridge parameter of 0.2 was chosen. The results of all three models are presented in Table 4.1.

Table 4.6 Comparison of RMSE between stepwise regression, ridge regression and LSTM model.

RMSE	**EUR/GBP**	**EUR/USD**	**GBP/USD**	**NZD/USD**
Stepwise	1.3346	0.9398	0.5940	1.5967
Ridge	0.6408	0.5991	0.6113	0.6101
LSTM	0.0126	0.0148	0.0148	0.0120

The table displays the root mean square error (RMSE) loss for each prediction, which represents the square root of the variance of the residuals. The residuals are the differences between the predicted values and the true values. A lower RMSE value indicates a better fit of the model to the data. Upon analyzing the results, it can be observed that stepwise regression exhibited the worst performance among the three models. Ridge regression achieved a lower RMSE than stepwise regression in all datasets except for GBP/USD, where stepwise regression had a slight advantage. LSTM outperformed the other models and yielded the best results across all four datasets. These findings align with the literature review, indicating that deep learning methods, such as LSTM, are more effective in addressing the issue of multicollinearity compared to statistical methods (Garg & Tai, 2013; Obite et al., 2020).

It is important to note that statistical approaches, such as linear regression, are based on certain assumptions, including a linear relationship between predictor and response variables, independence of residuals, homoscedasticity, and normally distributed residuals (Schmidt & Finan, 2018). These assumptions can limit the ability of statistical methods to fit each dataset adequately. In contrast, neural networks do not have such assumptions and can achieve better fitting and forecasting performance. With these insights, this research introduces the multicollinearity reduction extension to the LSTM model.

4.3. Model Configuration

The architecture of the baseline algorithmic trading network was based on an LSTM model, as depicted in Figure 4.1. The LSTM model is a specialized variant of recurrent neural networks (RNNs) designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. It introduces memory cells and three gating mechanisms: input, forget, and output gates. These gates allow the model to selectively retain or discard information at each time step, facilitating the capture of important context over longer distances. LSTMs excel in tasks such as natural language processing, speech recognition, and time series analysis. Our experiment harnessed the power of LSTMs to leverage temporal patterns in the data, aiming to improve model accuracy and predictive performance.

All the algorithms in this study were implemented using the Python programming language. In order to optimize the performance of our LSTM-based model, careful parameter tuning, and thoughtful model configuration were essential. The LSTM architecture relies on several hyperparameters, including the number of LSTM layers, the number of hidden units within each layer, the dropout rate, and the learning rate. We conducted an extensive parameter search to find the optimal values for these parameters, employing techniques such as grid search. Additionally, we experimented with various activation functions and optimization algorithms to determine the most suitable configuration for our model. Through comparison of different settings, we were able to identify the parameter values and model configuration that yielded the best results in terms of accuracy and generalization capability.

The final model consisted of three fully connected layers. The first layer was the LSTM recurrent layer, with an input size of 17 for the 17 input variables and 100 neurons in the hidden state. The second layer was a linear layer with 50 neurons, and the final layer contained four neurons representing the model's output categories: significant loss, loss, profit, and significant profit. This resulted in a total trainable parameter count of 52,701. The backpropagation algorithm used in this study was the Adam optimizer (Kingma & Ba, 2014).

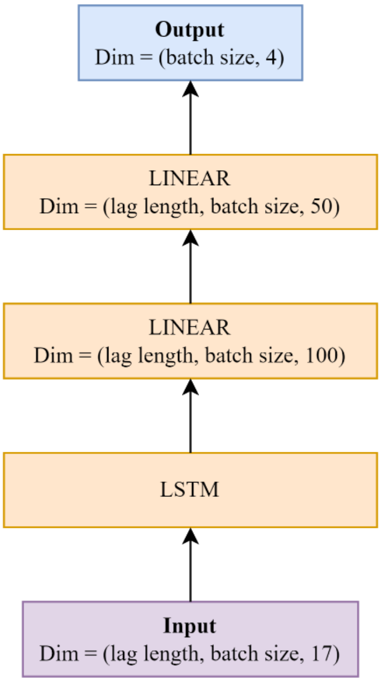

Output
Dim = (batch size, 4)
LINEAR
Dim = (lag length, batch size, 50)
LINEAR
Dim = (lag length, batch size, 100)
LSTM
Input
Dim = (lag length, batch size, 17)

Figure 4.1 Model configuration of the baseline LSTM model.

The weights and biases of the LSTM layer were initialized using a uniform distribution ranging from $-\sqrt{k}$ to $\sqrt{k}$, where k = 1/hidden size. The initial hidden state and cell state of the LSTM layer were set to zero. Similarly, the weights and biases for the subsequent linear layers were initialized with k = the number of input features.

For each model, the learning rates were independently configured to minimize the error. The same learning rate was applied to all layers. The learning rate was determined based on a simple heuristic in this study. It was initially set to 0.1 and then divided by 10 until no further improvements in the loss function were observed. The cross-entropy loss was used for classification tasks, while the root-mean-square error was used for regression tasks.

4.4. Comparison between Classification and Regression

This section aimed to demonstrate the trading performance of both the LSTM classification and regression forecasting models. The models were trained using the first 80% of the data, while the remaining 20% served as the test set. Each model underwent training for 100 epochs. The training and evaluation process was repeated 10 times for each of the 4 currency pairs. In essence, this involved training 10 models using the training set data for each currency pair, followed by measuring the profitability of each model using the test set data. By repeating this process, it ensured a more comprehensive range of performance outcomes, as the resulting neural network exhibited probabilistic behavior due to factors such as initialization methods and parameter optimization algorithms. Subsequently, this study examined how these factors affected the trading performance. The performance metrics considered for this purpose were accuracy and return.

Accuracy was defined as the sum of correct predictions for each label divided by the total number of predictions. Return referred to the profit or loss generated by the trading model on the test set. The dataset consisted of approximately 7000 hours of data. The model executed a trade if the prediction corresponded to label 1 (significant profit). Table 4.2 presented the mean and standard deviation of both accuracy and return for the regression and classification models.

Table 4.7 Comparison of trading result between classification model and regression model.

Evaluation Metric	Regression	Classification
Mean of Accuracy	22.47%	45.80%
Std of Accuracy	2.61%	2.67%
Mean of Returns (pips)	751	1747
Std of Returns (pips)	451	468

The regression model exhibited a mean accuracy of only 22.47% across the four datasets. In contrast, the classification model showed a significant improvement in this metric, achieving an accuracy of 45.80%, marking a 23.33% increase. Generally, classification has the potential to enhance the profitability of algorithmic trading models. The regression model yielded a trading return of 751 pips, whereas the classification model generated a return of 1747 pips. This translated to an additional 132.62% profit during the testing period. The standard deviation of accuracy was comparable for both models, with classification showing a slightly higher value, but it was the trading return that demonstrated a substantially higher mean for the classification model. The examined metrics indicate that classification is a superior problem formulation for datasets characterized by high multicollinearity. Figure 4.1 presents the box plot of accuracy for each foreign exchange pair, clearly illustrating the distinction between the results obtained through classification and regression.

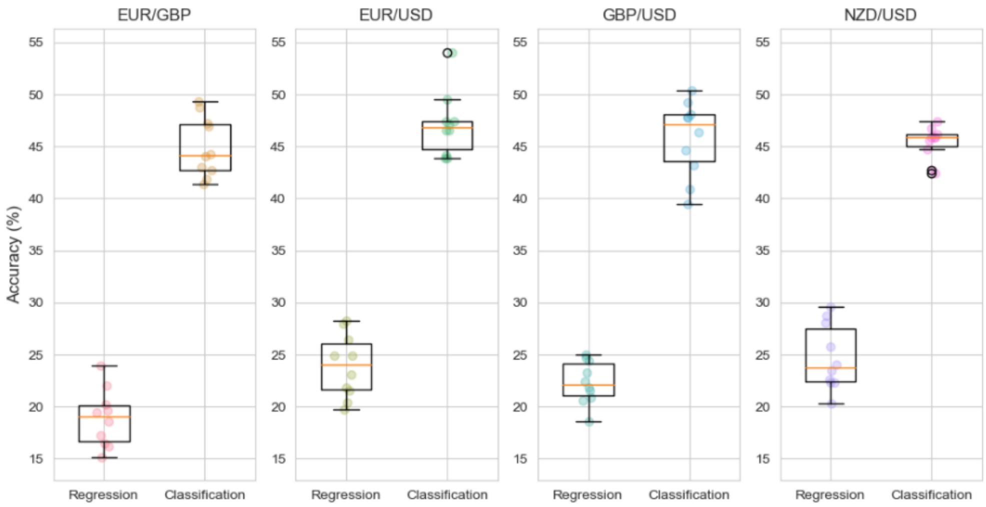


Figure 4.2 Comparison of accuracy between classification model and regression model for each foreign exchange pairs.

4.5. Performance of Proposed Method

The effectiveness of the proposed Multicollinearity Reduction Model (MRM) was evaluated by comparing it to the LSTM model. The MRM architecture incorporated the baseline LSTM, the proposed correlation embeddings, and the attention mechanism. Each model underwent training for 100 epochs. The experiment was repeated ten times, and the average results were calculated. These steps were replicated for each of the four datasets. The findings are presented in Table 4.3. Initially, this study examined the cross-entropy loss of each model, which is a relevant metric for classification problems. The cross-entropy loss quantifies the disparity between the predicted and true probability distributions. The loss function is defined as follows:

$$loss(x, class) = -x[class] + log\left(\sum_{j} exp\,(x[j])\right), \tag{23}$$

where x represents the input and class refers to the category index. The LSTM model yielded a training loss of 0.8435, whereas the MRM model achieved a training loss of 0.5088. These results demonstrated the effectiveness of the proposed method in enhancing performance under high multicollinearity conditions.

Table 4.8 Comparison of performance results for classification LSTM and Classification MRM.

Evaluation Metric	**LSTM**	**MRM**
Mean of Loss Function	0.8435	0.5088
Std of Loss Function	0.0241	0.0750
Mean of Accuracy	45.80%	45.10%
Std of Accuracy	2.67%	2.67%
Mean of Returns (pips)	1747	2787
Std of Returns (pips)	468	607

Subsequently, the impact of the MRM model on trading performance was evaluated. The LSTM model exhibited an average trading return of 1747 pips, whereas the MRM model achieved a higher mean trading return of 2787 pips. This corresponded to an additional profit of 59.53% during the testing period. To provide a visual representation of the improvement in trading return across different foreign exchange pairs, Figure 4.3 presents a bar chart.

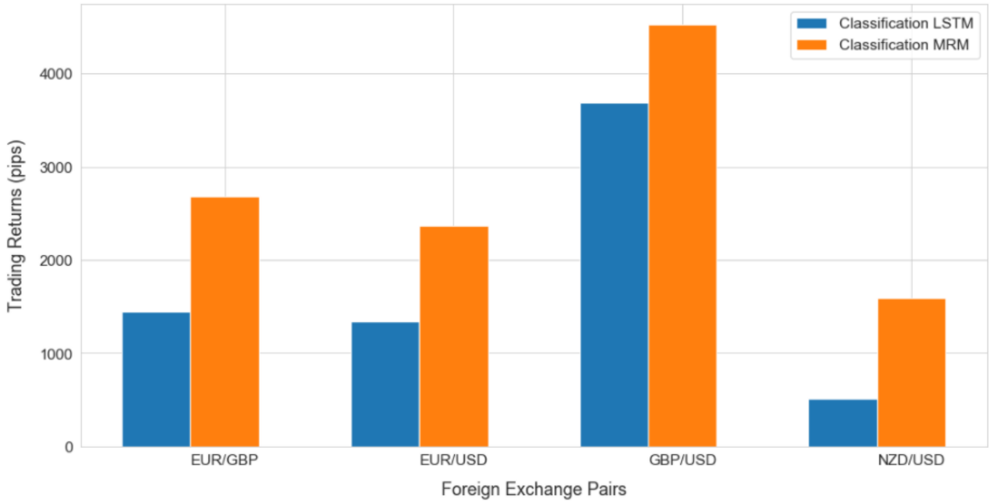


Figure 4.3 Comparison of Trading Returns between classification LSTM and classification MRM.

While there was a significant improvement in returns, the same cannot be said for accuracy. The baseline model demonstrated an average accuracy of 45.8%, while the proposed MRM model achieved an accuracy of 45.1%. This finding aligns with the literature review, which suggests that accuracy alone does not guarantee higher returns. The accuracy paradox and inconsistency profit bias contribute to this discrepancy. Accuracy fails to differentiate between classes and tends to favor the majority class, often overlooking the less important class. Therefore, in addition to accuracy, trading returns are evaluated.

The results indicate that the MRM model outperforms the LSTM model in identifying profitable trades. In other words, the MRM model exhibits higher precision and recall for each predicted class, highlighting its precision in high multicollinearity scenarios. The model only engages in trades when the predicted label is 1, indicating a buy signal. Consequently, this approach leads to higher returns per trade.

Moving forward, the proposed method was tested on a regression model to explore potential improvements. Table 4.4 presents the performance results using the same metrics. Although there was a minor improvement of 2.12% in accuracy, a significant enhancement is observed in trading return. The mean trading return for the LSTM model is 751 pips, while the MRM model achieved a mean trading return of 1216 pips. This corresponds to an additional profit of 61.92% during the testing period. Figure 4.4 presents a bar chart to visually depict the improvement in trading return for each foreign exchange pair. In all cases, classification resulted in higher returns, with the most substantial improvement observed in the EUR/GBP pair, where the baseline LSTM model barely achieved profitability.

Table 4.9 Comparison of performance result between regression LSTM and regression MRM.

Evaluation Metric	LSTM	MRM
Mean of Accuracy	22.47%	24.59%
Std of Accuracy	2.61%	2.41%
Mean of Returns (pips)	751	1216
Std of Returns (pips)	451	512

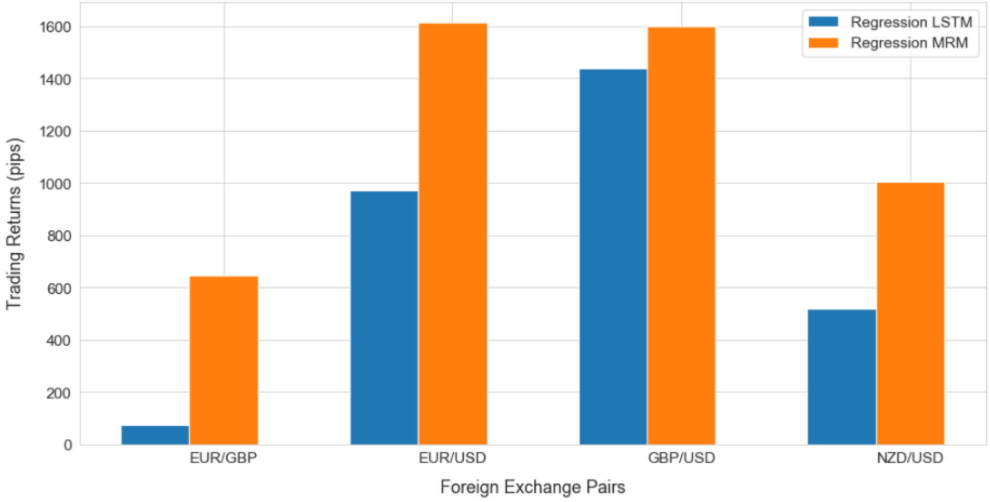


Figure 4.4 Comparison of Trading Returns between regression LSTM and regression MRM.

4.6. Discussion

The experiments conducted in this study confirmed that classification outperforms regression in high multicollinearity scenarios, exhibiting higher prediction accuracy. One possible explanation for this finding is that multiclass labels can contain more information. Instead of estimating a single point, classification models estimate both the upper and lower bounds of possible prices. From a mathematical perspective, framing financial forecasting as a classification problem may be easier in terms of model fitting compared to point estimation problems. This is because a classification task aggregates an interval of continuous points into a single point in the sample space. The proposed model only enters a trade when it has a high level of confidence in achieving significant profits.

Moreover, the MRM model demonstrated its effectiveness in enhancing the baseline LSTM model. By incorporating the proposed extensions, the model was able to improve precision and, consequently, the profitability of algorithmic trading models. While the LSTM model is proficient in capturing temporal effects, it fails to consider the interaction between data. The results indicate that neural networks with the proposed extensions can effectively learn the relevance and redundancy present in financial variables as intended. Furthermore, all potential information is retained as no variables are removed during the model building process. Although initially developed for classification, the proposed model also exhibited similar improvements in regression tasks.

Table 4.5 provides a comprehensive overview of the results obtained for each foreign exchange pair and model. One inference drawn from these findings is the significance of problem formulation in algorithmic trading. On average, changing the problem formulation yields better returns compared to changing the model itself. Consequently, there is still ample room for innovation in financial forecasting, particularly regarding the prediction target aspect.

Table 4.10 Compilation of performance results for every foreign exchange pairs.

		Metric	**EUR/GBP**	**EUR/USD**	**GBP/USD**	**NZD/USD**
Regression	LSTM	Accuracy	18.9%	23.9%	22.3%	24.7%
		Return	74	973	1438	519
	MRM	Accuracy	20.0%	24.6%	24.5%	29.3%
		Return	646	1613	1600	1006
Classification	LSTM	Accuracy	45.0%	47.1%	45.8%	45.4%
		Return	1445	1340	3689	515
	MRM	Accuracy	43.9%	46.1%	47.3%	43.1%
		Return	2677	2364	4521	1587

CHAPTER 5

CONCLUSION

5. Conclusions

It is important for financial markets to have efficient price discovery, and with advancements in technology, financial forecasting and algorithmic trading have played an increasingly significant role in achieving this goal. One major challenge faced in financial forecasting is multicollinearity, which arises due to the abundance of data and can lead to unreliable predictions from models. In this research, an algorithmic trading process for the foreign exchange markets was developed, with technical indicators generated as the predictive variables. The presence of multicollinearity in financial datasets was confirmed through the use of the VIF.

The existing approaches in the literature to mitigate multicollinearity can be categorized into variable selection and modified estimators. Variable selection aims to reduce the number of predictors to the most relevant ones, potentially reducing noise but also removing the potential incremental predictive value. Most widely used modified estimators are based on traditional statistical models rather than neural networks. Therefore, this research focuses on a neural network approach and problem formulation, with the objective of comparing the performance of mitigating multicollinearity between the Classification Neural Network and the Regression Neural Network.

1. **To compare the performance of mitigating multicollinearity between Classification Neural Network and Regression Neural Network.**

The experiment demonstrated that the classification approach achieved a 23.33% higher accuracy compared to regression. Classification proved to be more effective in predicting the future direction of movement than a point estimation.

2. **To investigate the potential improvement in performance of proposed method over Neural Network.**
    a. **To investigate the potential improvement in prediction accuracy of proposed attention mechanism and embeddings over neural network in the presence of multicollinearity.**
    b. **To investigate the potential improvement in trading returns of proposed attention mechanism and embeddings over neural network in the presence of multicollinearity.**

The proposed MRM model did not improve the prediction accuracy of the baseline model, as the difference in mean and standard deviation of accuracy was comparable across all four datasets. However, the MRM model exhibited a 59.53% higher trading return. This experiment revealed that accuracy does not always translate into higher returns in an algorithmic trading simulation. The proposed model demonstrated higher profitability and returns despite having the same accuracy, indicating that MRM can enhance precision in financial forecasting and mitigate reliability issues associated with high multicollinearity data.

3. **To investigate the potential improvement in performance of proposed method on regression.**

The results showed that the MRM model achieved a 61.92% higher trading return in regression as well. The effects of the proposed method on regression were similar to those observed in classification, with higher precision in prediction leading to higher returns over the testing period. Moreover, it was evident that changing the problem formulation yielded greater marginal improvement compared to enhancing the predictive model.

Our model introduces an attention module to identify relevant variables and utilizes a correlation-based embedding to model redundancy within the variables. Unlike feature selection methods, the proposed method does not remove variables, making it more effective in prediction as it avoids the risk of discarding relevant features. Furthermore, neural networks have the ability to uncover nonlinear relationships that statistical approaches often fail to capture.

In future work, it would be of interest to expand the features beyond price-based technical indicators and incorporate fundamental data and news data, which are also commonly used in algorithmic trading models and exhibit high multicollinearity. Assessing the performance of the proposed MRM in higher-dimensional datasets with the addition of these features would be valuable. Additionally, experimental exploration of alternative measures, aside from correlation, as proxies for redundancy in features could be conducted.

References

Alexander, C. (2008). *Market risk analysis, practical financial econometrics* (Vol. 2): John Wiley & Sons.

Algamal, Z. Y. (2018). Biased estimators in Poisson regression model in the presence of multicollinearity: A subject review. *Al-Qadisiyah Journal for Administrative and Economic Sciences, 20*(1), 37-43.

Alin, A. (2010). Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(3), 370-374.

Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018). *Evaluation of bidirectional LSTM for short-and long-term stock market prediction.* Paper presented at the 2018 9th international conference on information and communication systems (ICICS).

Andrews, J. L., & McNicholas, P. D. (2014). Variable selection for clustering and classification. *Journal of Classification, 31*(2), 136-153.

Arashi, M., Norouzirad, M., Roozbeh, M., & Mamode Khan, N. (2021). A High-Dimensional Counterpart for the Ridge Estimator in Multicollinear Situations. *Mathematics, 9*(23), 3057.

Askin, R. G. (1982). Multicollinearity in regression: Review and examples. *Journal of Forecasting, 1*(3), 281-292.

Assaf, A. G., Tsionas, M., & Tasiopoulos, A. (2019). Diagnosing and correcting the effects of multicollinearity: Bayesian implications of ridge regression. *Tourism Management, 71*, 1-8.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barndorff-Nielsen, O. E., & Shephard, N. (2005). Variation, jumps, market frictions and high frequency data in financial econometrics.

Barndorff-Nielsen, O. E., & Shephard, N. (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64*(2), 253-280.

Beck, T. (2008). The econometrics of finance and growth.

Belsley, D. A., Kuh, E., & Welsch, R. E. (2005). *Regression diagnostics: Identifying influential data and sources of collinearity*: John Wiley & Sons.

Bento, P., Pombo, J., Calado, M., & Mariano, S. (2018). A bat optimized neural network and wavelet transform approach for short-term price forecasting. *Applied energy, 210*, 88-97.

Bies, R. R., Muldoon, M. F., Pollock, B. G., Manuck, S., Smith, G., & Sale, M. E. (2006). A genetic algorithm-based, hybrid machine learning approach to model selection. *Journal of pharmacokinetics and pharmacodynamics, 33*(2), 195.

Bollinger, J. (1992). Using bollinger bands. *Stocks & Commodities, 10*(2), 47-51.

Brooks, C. (2019). *STATA Guide for Introductory Econometrics for Finance*: Cambridge University Press.

Brownlees, C. T., & Gallo, G. M. (2006). Financial econometric analysis at ultra-high frequency: Data handling concerns. *Computational statistics & data analysis, 51*(4), 2232-2245.

Campbell, J. Y., Lo, A., & MacKinlay, C. (1997). The Econometrics of Financial Markets," Princeton University Press, Princeton. *New Jersey: MacKinlay*.

Castillo, F. A., & Villa, C. M. (2005). *Symbolic regression in multicollinearity problems.* Paper presented at the Proceedings of the 7th annual conference on Genetic and evolutionary computation.

Chan, J. Y. L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z. W., & Chen, Y. L. (2022). Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review. *Mathematics, 10*(8), 1283.

Chan, J. Y. L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z. W., . . . Chen, Y. L. (2022). A Correlation-Embedded Attention Module to Mitigate Multicollinearity: An Algorithmic Trading Application. *Mathematics, 10*(8), 1231.

Chandrasekhar, C., Bagyalakshmi, H., Srinivasan, M., & Gallo, M. (2016). Partial ridge regression under multicollinearity. *Journal of Applied Statistics, 43*(13), 2462-2473.

Chen, A. S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers & Operations Research, 30*(6), 901-923.

Chen, C. W., Tsai, Y. H., Chang, F. R., & Lin, W. C. (2020). Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Systems, 37*(5), e12553.

Chen, G., Chen, Y., & Fushimi, T. (2021). *Application of deep learning to algorithmic trading*.

Chen, S. H., Chang, C. L., & Du, Y. R. (2012). Agent-based economic models and econometrics. *The Knowledge Engineering Review, 27*(2), 187-219.

Chen, Y., & Hao, Y. (2020). A novel framework for stock trading signals forecasting. *Soft Computing, 24*(16), 12111-12130.

Chiriac, R., & Voev, V. (2011). Modelling and forecasting multivariate realized volatility. *Journal of Applied Econometrics, 26*(6), 922-947.

Chong, I. G., & Jun, C. H. (2005). Performance of some variable selection methods when multicollinearity is present. *Chemometrics and intelligent laboratory systems, 78*(1-2), 103-112.

Christoffersen, P. F., & Diebold, F. X. (2000). How relevant is volatility forecasting for financial risk management? *Review of Economics and Statistics, 82*(1), 12-22.

Cook, J. (2021). *6 Most Popular Currencies for Trading*. Investopedia. https://www.investopedia.com/articles/forex/11/popular-currencies-and-why-theyre-traded.asp

Duzan, H., & Shariff, N. S. B. M. (2015). Ridge regression for solving the multicollinearity problem: review of methods and models. *Journal of Applied Science*.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics, 32*(2), 407-499.

Elman, J. L. (1990). Finding structure in time. *Cognitive science, 14*(2), 179-211.

Engle, R. (2001). GARCH 101: The use of ARCH/GARCH models in applied econometrics. *Journal of economic perspectives, 15*(4), 157-168.

Engle, R. (2004). Risk and volatility: Econometric models and financial practice. *American economic review, 94*(3), 405-420.

Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications, 29*(4), 927-940.

Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70*(5), 849-911.

Garg, A., & Tai, K. (2012). *Comparison of regression analysis, artificial neural network and genetic programming in handling the multicollinearity problem.* Paper presented at the 2012 Proceedings of International Conference on Modelling, Identification and Control.

Garg, A., & Tai, K. (2013). Comparison of statistical and machine learning methods in modelling of data with multicollinearity. *International Journal of Modelling, Identification and Control, 18*(4), 295-312.

Genuer, R., Poggi, J. M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters, 31*(14), 2225-2236.

Giacomini, R., & Rossi, B. (2010). Forecast comparisons in unstable environments. *Journal of Applied Econometrics, 25*(4), 595-620.

Gorman, J. W., & Toman, R. (1966). Selection of variables for fitting equations to data. *Technometrics, 8*(1), 27-51.

Gu, Y., Liu, J., Chen, Y., Jiang, X., & Yu, H. (2014). TOSELM: timeliness online sequential extreme learning machine. *Neurocomputing, 128*, 119-127.

Guo, L., Hao, J. H., & Liu, M. (2014). An incremental extreme learning machine for online sequential learning problems. *Neurocomputing, 128*, 50-58.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hamaker, H. (1962). On multiple regression analysis. *Statistica Neerlandica, 16*(1), 31-56.

Harris, M. (2008). *Profitability and Systematic Trading: A Quantitative Approach to Profitability, Risk, and Money Management* (Vol. 342): John Wiley & Sons.

Hatami, N., Gavet, Y., & Debayle, J. (2018). *Classification of time-series images using deep convolutional neural networks.* Paper presented at the Tenth international conference on machine vision (ICMV 2017).

Hocking, R. R., & Leslie, R. (1967). Selection of the best subset in regression analysis. *Technometrics, 9*(4), 531-540.

Hördahl, P., Tristani, O., & Vestin, D. (2006). A joint econometric model of macroeconomic and term-structure dynamics. *Journal of Econometrics, 131*(1-2), 405-444.

Horel, A. (1962). Applications of ridge analysis toregression problems. *Chem. Eng. Progress., 58*, 54-59.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications, 129*, 273-285.

Hua, Y. (2020). *An efficient traffic classification scheme using embedded feature selection and lightgbm.* Paper presented at the 2020 Information Communication Technologies Conference (ICTC).

Huang, J., & Yang, H. (2014). A two-parameter estimator in the negative binomial regression model. *Journal of Statistical Computation and Simulation, 84*(1), 124-134.

Huynh, H. T., & Won, Y. (2011). Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks. *Pattern Recognition Letters, 32*(14), 1930-1935.

Iba, H., & Sasaki, T. (1999). *Using genetic programming to predict financial data.* Paper presented at the Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406).

Inan, D., & Erdogan, B. E. (2013). Liu-type logistic estimator. *Communications in Statistics-Simulation and Computation, 42*(7), 1578-1586.

Jasiak, J. (2001). *Financial econometrics: problems, models, and methods*: Princeton University Press.

Jasic, T., & Wood, D. (2004). The profitability of daily stock market indices trades based on neural network predictions: Case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965–1999. *Applied Financial Economics, 14*(4), 285-297.

Kashid, D., & Kulkarni, S. (2002). A more general criterion for subset selection in multiple linear regression. *Communications in Statistics-Theory and Methods, 31*(5), 795-811.

Katrutsa, A., & Strijov, V. (2015). Stress test procedure for feature selection algorithms. *Chemometrics and intelligent laboratory systems, 142*, 172-183.

Katrutsa, A., & Strijov, V. (2017). Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications, 76*, 1-11.

Kejian, L. (1993). A new class of blased estimate in linear regression. *Communications in Statistics-Theory and Methods, 22*(2), 393-402.

Kibria, B., & Lukman, A. F. (2020). A new ridge-type estimator for the linear regression model: Simulations and applications. *Scientifica, 2020*.

Kim, J. M., Wang, N., Liu, Y., & Park, K. (2020). Residual control chart for binary response with multicollinearity covariates by neural network model. *Symmetry, 12*(3), 381.

Kim, R., So, C. H., Jeong, M., Lee, S., Kim, J., & Kang, J. (2019). Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*.

Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PloS one, 14*(2), e0212320.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koop, G., & Korobilis, D. (2012). Forecasting inflation using dynamic model averaging. *International Economic Review, 53*(3), 867-886.

Krishnaveni, P., Swarnam, S., & Prabakaran, V. (2019). An empirical study to analyse overbought and oversold periods of shares listed in CNX Bankex. *International Journal of Management, IT and Engineering, 9*(3), 155-167.

Lafi, S., & Kaneene, J. (1992). An explanation of the use of principal-components analysis to detect and correct for multicollinearity. *Preventive Veterinary Medicine, 13*(4), 261-275.

Larabi-Marie-Sainte, S. (2021). Outlier Detection Based Feature Selection Exploiting Bio-Inspired Optimization Algorithms. *Applied Sciences, 11*(15), 6769.

Lavery, M. R., Acharya, P., Sivo, S. A., & Xu, L. (2019). Number of predictors and multicollinearity: What are their effects on error and bias in regression? *Communications in Statistics-Simulation and Computation, 48*(1), 27-38.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Lee, M. C. (2022). Research on the Feasibility of Applying GRU and Attention Mechanism Combined with Technical Indicators in Stock Trading Strategies. *Applied Sciences, 12*(3), 1007.

Leung, M. T., Daouk, H., & Chen, A.-S. (2000). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of forecasting, 16*(2), 173-190.

Li, C., Wang, H., Wang, J., Tai, Y., & Yang, F. (2018). *Multicollinearity problem of CPM communication signals and its suppression method with PLS algorithm.* Paper presented at the Proceedings of the Thirteenth ACM International Conference on Underwater Networks & Systems.

Liu, G., & Wang, X. (2019). A new metric for individual stock trend prediction. *Engineering Applications of Artificial Intelligence, 82*, 1-12.

Liu, K. (2003). Using Liu-type estimator to combat collinearity. *Communications in Statistics-Theory and Methods, 32*(5), 1009-1020.

Lucey, B. M., & Muckley, C. (2011). Robust global stock market interdependencies. *International Review of Financial Analysis, 20*(4), 215-224.

M'ng, J. C. P., & Aziz, A. A. (2016). Using neural networks to enhance technical trading rule returns: A case with KLCI. *Athens J. Bus. Econ, 2*, 63-70.

Mahadi, M., Ballal, T., Moinuddin, M., & Al-Saggaf, U. M. (2022). A Recursive Least-Squares with a Time-Varying Regularization Parameter. *Applied Sciences, 12*(4), 2077.

Maitra, S., & Yan, J. (2008). Principle component analysis and partial least squares: Two dimension reduction techniques for regression. *Applying Multivariate Statistical Models, 79*, 79-90.

Mallows, C. (1964). *Choosing variables in a linear regression: A graphical aid.* Paper presented at the Central Regional Meeting of the Institute of Mathematical Statistics, Manhattan, KS, 1964.

Mason, C. H., & Perreault Jr, W. D. (1991). Collinearity, power, and interpretation of multiple regression analysis. *Journal of marketing research, 28*(3), 268-280.

Mills, T. C., & Markellos, R. N. (2008). *The econometric modelling of financial time series*: Cambridge university press.

Mingyue, Q., Cheng, L., & Yu, S. (2016). *Application of the Artifical Neural Network in predicting the direction of stock market index.* Paper presented at the 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS).

Misra, P., & Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *Int J Emerg Technol, 11*, 659-665.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*: John Wiley & Sons.

Mosier, C. I. (1951). I. Problems and designs of cross-validation 1. *Educational and Psychological Measurement, 11*(1), 5-11.

Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). Applied linear statistical models.

Nguyen, V. C., & Ng, C. T. (2020). Variable selection under multicollinearity using modified log penalty. *Journal of Applied Statistics, 47*(2), 201-230.

Nobrega, J. P., & Oliveira, A. L. (2019). A sequential learning method with Kalman filter and extreme learning machine for regression and time series forecasting. *Neurocomputing, 337*, 235-250.

Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic trading. *Computer, 44*(11), 61-69.

Obite, C., Olewuezi, N., Ugwuanyim, G., & Bartholomew, D. (2020). Multicollinearity effect in regression analysis: A feed forward artificial neural network approach. *Asian journal of probability and statistics, 6*(1), 22-33.

Olson, D., & Mossman, C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of forecasting, 19*(3), 453-465.

Paolella, M. S., & Taschini, L. (2008). An econometric analysis of emission allowance prices. *Journal of Banking & Finance, 32*(10), 2022-2032.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics, 160*(1), 246-256.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence, 27*(8), 1226-1238.

Qaraad, M., Amjad, S., Manhrawy, I. I., Fathi, H., Hassan, B. A., & El Kafrawy, P. (2021). A hybrid feature selection optimization model for high dimension data classification. *IEEE Access, 9*, 42884-42895.

Ralston, A., & Wilf, H. S. (1960). *Mathematical methods for digital computers*. Retrieved from

Rasekhschaffe, K. C., & Jones, R. C. (2019). Machine learning for stock selection. *Financial Analysts Journal, 75*(3), 70-88.

Roozbeh, M., Arashi, M., & Hamzah, N. A. (2020). Generalized cross-validation for simultaneous optimization of tuning parameters in ridge regression. *Iranian Journal of Science and Technology, Transactions A: Science, 44*(2), 473-485.

Roozbeh, M., Babaie-Kafaki, S., & Aminifard, Z. (2022). Improved high-dimensional regression models with matrix approximations applied to the comparative case studies with support vector machines. *Optimization Methods and Software*, 1-18.

Roozbeh, M., Babaie–Kafaki, S., & Aminifard, Z. (2021). A nonlinear mixed–integer programming approach for variable selection in linear regression model. *Communications in Statistics-Simulation and Computation*, 1-12.

Rundo, F. (2019). Deep LSTM with reinforcement learning layer for financial trend prediction in FX high frequency trading systems. *Applied Sciences, 9*(20), 4460.

Schmidt, A. F., & Finan, C. (2018). Linear regression and the normality assumption. *Journal of clinical epidemiology, 98*, 146-151.

Schroeder, M. A., Lander, J., & Levine-Silverman, S. (1990). Diagnosing and dealing with multicollinearity. *Western journal of nursing research, 12*(2), 175-187.

Senawi, A., Wei, H.-L., & Billings, S. A. (2017). A new maximum relevance-minimum multicollinearity (MRmMC) method for feature selection and ranking. *Pattern Recognition, 67*, 47-61.

Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing, 70*, 525-538.

Sezer, O. B., Ozbayoglu, A. M., & Dogdu, E. (2017). *An artificial neural network-based stock trading system using technical analysis and big data framework.* Paper presented at the proceedings of the southeast conference.

Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv preprint arXiv:1803.06386*.

Singh, B., Chaubey, Y., & Dwivedi, T. (1986). An almost unbiased ridge estimator. *Sankhyā: The Indian Journal of Statistics, Series B*, 342-346.

Singh, S. G., & Kumar, S. V. (2021). Dealing with Multicollinearity problem in analysis of side friction characteristics under urban heterogeneous traffic conditions. *Arabian Journal for Science and Engineering, 46*(11), 10739-10755.

Tamura, R., Kobayashi, K., Takano, Y., Miyashiro, R., Nakata, K., & Matsui, T. (2017). Best subset selection for eliminating multicollinearity. *Journal of the Operations Research Society of Japan, 60*(3), 321-336.

Tamura, R., Kobayashi, K., Takano, Y., Miyashiro, R., Nakata, K., & Matsui, T. (2019). Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. *Journal of Global Optimization, 73*(2), 431-446.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267-288.

Treleaven, P., Galas, M., & Lalchand, V. (2013). Algorithmic trading review. *Communications of the ACM, 56*(11), 76-85.

Türkan, S., & Özel, G. (2016). A new modified Jackknifed estimator for the Poisson regression model. *Journal of Applied Statistics, 43*(10), 1892-1905.

Valverde-Albacete, F. J., & Peláez-Moreno, C. (2014). 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PloS one, 9*(1), e84217.

Vo, A., & Yost-Bremm, C. (2020). A high-frequency algorithmic trading strategy for cryptocurrency. *Journal of Computer Information Systems, 60*(6), 555-568.

Vrieze, S. I. (2012). Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods, 17*(2), 228.

Wang, J., Sun, T., Liu, B., Cao, Y., & Zhu, H. (2021). CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets. *arXiv preprint arXiv:2104.04041*.

Wang, W., & Mishra, K. K. (2018). A novel stock trading prediction and recommendation system. *Multimedia Tools and Applications, 77*(4), 4203-4215.

Weisberg, S. (2005). *Applied linear regression* (Vol. 528): John Wiley & Sons.

Willis, M., Hiden, H., Hinchliffe, M., McKay, B., & Barton, G. W. (1997). Systems modelling using genetic programming. *Computers & chemical engineering, 21,* S1161-S1166.

Wold, H. (1982). Soft modeling: the basic design and some extensions. *Systems under indirect observation, 2,* 343.

Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems, 58*(2), 109-130.

Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. *Wireless Personal Communications, 102*(2), 1645-1656.

Ye, Y., Squartini, S., & Piazza, F. (2013). Online sequential extreme learning machine in nonstationary environments. *Neurocomputing, 116,* 94-101.

Yılmaz, M. K., Erdem, O., Eraslan, V., & Arık, E. (2015). Technology upgrades in emerging equity markets: Effects on liquidity and trading activity. *Finance Research Letters, 14,* 87-92.

Yong, B. X., Abdul Rahim, M. R., & Abdullah, A. S. (2017). *A stock market trading system using deep neural network.* Paper presented at the Asian simulation conference.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine, 13*(3), 55-75.

Yuan, H., Zhou, H., Cai, Z., Zhang, S., & Wu, R. (2022). Dynamic Pyramid Attention Networks for multi-orientation object detection. *Journal of Internet Technology, 23*(1), 79-90.

Zeng, L., & Xie, J. (2014). Group variable selection via SCAD-L 2. *Statistics, 48*(1), 49-66.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1253.

Zhang, X., Liu, S., & Zheng, X. (2021). Stock Price Movement Prediction Based on a Deep Factorization Machine and the Attention Mechanism. *Mathematics, 9*(8), 800.

Zhao, N., Xu, Q., Tang, M. L., Jiang, B., Chen, Z., & Wang, H. (2020). High-dimensional variable screening under multicollinearity. *Stat, 9*(1), e272.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications, 67,* 126-139.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301-320.