FOREX TREND FORECASTING BASED ON LONG SHORT TERM MEMORY AND ITS VARIATIONS WITH HYBRID ACTIVATION FUNCTIONS

A thesis submitted for the degree of Doctor of Philosophy

By

Tianyu Zhou

Department of Computer Science

Brunel University London

SUPERVISORS

Dr Fang Wang

Professor Xiaohui Liu

Abstract

The foreign exchange (Forex) market, as one of the most important financial markets in the globe, has attracted many investors. In order to support forex traders' trading decisions, accurately predicting the forex prices has continued to be a popular but challenging topic. Due to the high complexity of the forex market, it is always a question of how effective the forex prediction could be.

With the rapid development on machine learning in the last decades, deep learning has been applied successfully to many areas including the forex market. Consequently, numerous research papers have been published, which aim to improve the accuracy of forex prediction. The Long Short-Term Memory (LSTM) neural network, a kind of artificial neural network, has been widely used, which is specially designed to analyse time series data. Due to its strong learning capability, the LSTM neural network has now been used to predict complex forex trading based on historical data. However, there is a lack of an authoritative and commonly accepted guidance on how to conduct proper forex predictions by using LSTM. The application of deep learning to financial forecasting is still in a developing stage.

This research aims to investigate the feasibility of applying deep learning, particularly the LSTM neural network to the foreign exchange market and to enhance the prediction accuracy via improved LSTM algorithms.

In this thesis, all of the fundamental and technical features related to forex trading have been collected and analysed comprehensively. The influential features are then selected to be used as the inputs for forex prediction. Based on these inputs, a LSTM is specifically built to predict the trends of forex prices, which are identified as a suitable prediction target for forex traders. Notably, a new validation method is also introduced to overcome the problems in the traditional time series validation methods.

Furthermore, a novel LSTM algorithm using hybrid activation functions in the same hidden layer is proposed to improve the prediction accuracy for forex trend predictions. Extensive experiments have been conducted and the experimental results have shown that the performance of the LSTM with hybrid activation functions has outperformed that of the standard LSTM. The generasalibility of the hybrid activation functions based LSTM has also been proved by its successful applications to different ANNs (e.g., RNNs) and datasets.

Keywords: forex, prediction, time series, neural network, artificial intelligence, Long Short-Term Memory (**LSTM), hybrid activation function**

Acknowledgements

Firstly, I would like to express my sincere thanks to my principal supervisor Dr Fang Wang. She made valuable suggestions for my research and gave the right guidance and encouragement when I encountered difficulties in the research. Dr Fang Wang spent a lot of effort and time to guide my research.

I would also like to thank all the researchers who provided help in my research; their research papers gave me great inspirations. Special thanks are given to Jason Brownlee, whose guiding papers on machine learning helped me extensively. His commutations cleared many of my concerns.

Finally, I sincerely thank my parents. Without their encouragement and support, I wouldn't be able to complete this thesis successfully.

Table of Contents

List of Table

List of Figures

Chapter 1 Introduction

1.1 Research Motivation

The foreign exchange (Forex) Market is a global, decentralised and most liquid financial market for exchanging different currencies. Its high trading volume and continuity keep attracting forex investors, and there are still more and more investors switching to the forex market from the stock market. Compared with the stock market, the forex market has three most outstanding features which are very beneficial for investment (Admiral Markets, 2019).

Firstly, the forex market is a 24-hours operating market during weekdays globally. Investors can trade forex at any time during the weekdays. In contrast, the regular trading of the stock market is limited to a couple of hours during a weekday and the open and close times of the stock market is different across different countries. Moreover, the stock market in many countries has its own restrictions. For example, in China, there are T+1 or T+2 trading restrictions to prevent excessive speculations. Secondly, due to the substantial transaction volume, it is impossible for any particular investor, financial group and institution to control such a large forex market. Different from forex, the total number of shares is limited in a stock market, so any trading behaviour from a significant shareholder will affect the prices of the shares greatly. Finally, investors can make profits in both rising or falling markets in forex trading. That is, investors can earn money by ‘go long’ or ‘go short’ separately. However, ‘go short’ is restricted for the stock market in some countries. These three distinctive features make forex investment more and more popular nowadays.

With the increased popularity of the forex market, how to accurately predicting forex trading has become a hot topic in both the academic and financial areas. Accurate forex trading predictions is not only beneficial to individual forex investors, but can also help organisations or governments make hedging decisions, budget estimation, etc.

In general, there are two kinds of forex forecasting methods (Admiral Markets, 2019): fundamental analysis and technical analysis, based on different theories. There are usually three popular theories for fundamental analysis: Interest Rate Parity (IRP) Theory, Purchasing Power Parity (PPP) Theory and Theory of International Indebtedness. The first two theories are more used by the public, so the indexes reflecting the purchasing power and interest rates are often applied in the fundamental analysis for forex forecast. There are also three theories commonly applied to technical analysis. The first theory believes that historical prices comprises everything, that is, any factors affecting the forex market such as economic, political and psychological expectations will be fully reflected in forex prices. The second theory is that exchange rates always change in a regular pattern and the pattern can be identified from the historical data. Finally, the third kind of technical analysis trusts that history will repeat, so forex prices are predictable by evaluating the past prices.

The above introduction has shown that fundamental analysis and technical analysis are based on different theories for forecasting, and these two kinds of analysis methods are usually used separately in practice, even though each method has its own advantages and disadvantages. This research has conducted a systematic study of these two analysis methods, particularly the features or factors involved in each method. It is assumed that a proper combination of the fundamental and technical will make forex prediction better than using either one individually and the following experimental results have proved this.

Machine learning, particularly deep learning, has recently achieved great success in many areas. AlphaGo[1] beat Lee Sedol in March 2016, which was the first time that a computer program beat a 9-dan professional in a five-game of the board game Go match. Since then, a global attention has been raised to artificial intelligence (AI). It is now widely believed that artificial intelligence is able to overbear humans in a series of specific areas. Machine learning, as an intelligent technique of AI has a strong capability to learn from given data automatically. The same capability can be applied to forex forecasting as well, by using the features or factors drawn from both fundamental and technical analyses to make time series predictions.

Artificial Neural Networks (ANNs) as a kind of powerful machine learning technique, is especially considered and developed in this research to make forex time series predictions. An ANN is a mathematical model inspired by the functions of human neural networks (HNNs) in the brain. One of the most important features of artificial neural networks is their capability to handle imperfect data. Forex prices are especially complex data, bearing lots of uncertainties, nonlinearities and non-Gaussian disturbances. To deal with the huge complexity of time series problems such as forex trading, Long Short Term Memory (LSTM) is often applied and it is used as a major method for forex trend forecasting in this research as well.

The details of how to apply LSTM neural networks to forex prediction are fully investigated in this thesis. Moreover, inspired by human neural networks, an evolution of the LSTM neural network, which utilises hybrid activation functions in the same hidden layer, is developed, together with comprehensive evaluations and comparisons of its performance when compared with other artificial neural networks such as Recurrent Neural Networks (RNN) and the classical LSTM.

[1] AlphaGo is an AI program that plays the board game Go, which uses machine learning algorithms to play.

1.2 Research Objectives

The aim of this research is to improve the accuracy of forex trend forecasting by using Long Short Term Memory (LSTM) neural networks. To achieve this, this research has five objectives as listed below:

- To conduct a comprehensive literature review on machine learning, particularly deep learning for forex forecasting;
- To study and select the proper features or indicators which have a significant impact on forex trading; and to decide a suitable prediction target for forex forecasting;
- To design a new validation method to overcome the limitations of current validation methods;
- To design and implement the new validation method based LSTM neural network to predict forex trends;
- To design and develop a novel LSTM algorithm with hybrid activation functions to further improve the forecasting performance of the LSTM.

The first objective aims to have a deep understanding of the related research areas and find suitable datasets for LSTM neural network training. It is well known that data is one of the two most essential components required for machine learning (ML) because ML is a kind of learning based on the given data. For forex forecasting, the training data and particularly the features of the training data need to be strongly pertinent to forex exchange so as to make the learning efficient and effective. For this reason, investigating the existing methods for forex forecasting and discovering the suitable dataset with proper features are an important objective to this research.

The second objective attempts to decide a proper prediction target for forex forecasting. The current studies have found that price predictions and price trend predictions are two different targets and have varied impacts in forex trading. When used in machine learning, they actually belong to different learning problems, corresponding to regression problems and classification problems respectively. Price trend is a more practical and appropriate target when doing machine learning for forex trading, as elaborated in Section 2.1.4. It is hence chosen as the prediction target in this research.

The third objective investigates a reasonable validation method for the complex machine learning for forex time series. Usually, a proper machine learning procedure involves training, validation and testing. The limitation of current validation methods for time series learning is the extremely large time complexity and restriction of new techniques application. To overcome this limitation, a new validation method is designed to allow many neural network optimization techniques that can be applied during neural network training. The feasibility of this new validation method is tested on two forex exchange datasets.

The fourth objective is to build a LSTM neural network for forex forecasting, with the chosen forex features and the new validation method. As a result, the performance of standard LSTM neural network can be used to evaluate the new validation method and the new developed LSTM algorithm (LSTM with Hybrid Activation Functions).

The last objective aims to propose a new LSTM algorithm to improve the prediction accuracy by applying hybrid activation functions for hidden neurons. This is inspired by biological neural networks which employ varied activation functions but perform intelligently in dynamic, changing environments. Two hypotheses of using the hybrid activation functions in the same hidden layer have been developed and evaluated. The performance of the new proposed hybrid LSTM has been tested and proved to outperform both traditional LSTMs and RNNs.

1.3 Research Questions

This research addresses five research questions in total, corresponding the five objectives introduced above. These research questions are:

- What features or factors are useful for forex forecasting, and which features or factors have the most significant impact on forex trading?
- What prediction target is suitable for forex forecasting, in the best interest of forex traders?
- How to deal with the limitations of current validation methods and how to build a new, proper validation method for time series?
- How to predict forex trends by using LSTM neural networks, with properly selected fundamental and technical features and validation method?
- How to build and test LSTMs with hybrid activation functions in the same hidden layer, with an aim to improve forex trend prediction accuracies.

These five research questions are mainly studied in Chapters 3 and 4.

1.4 Research Contributions

There are five main contributions of this research, listed below:

- Traders in forex trading usually use fundamental analysis and/or technical analysis to make their trading decisions, but in scientific research, most studies used only technical data for machine learning for forex forecasting. There is little research on using features from both fundamental and technical data to make forex predictions. This research is the first one that provides a comprehensive investigation of all the features or indicators involved in the fundamental and technical analysis, and based on which, chooses the most influential indicators by using the Decision-Tree-based feature selection method for the following intelligent forex forecasting.

- A proper prediction target is specifically chosen for forex forecasting. After analysing the benefits of different prediction targets, the real demands of forex traders and the properties of ANN learning, it is decided that forex price trends are the most suitable prediction target for LSTMs to learn, in order to maximise the trading profits for traders.
- A new validation method for the time series classification problem is designed to overcome the shortcomings of the traditional validation methods. Traditional time series validation methods are usually time-consuming and have a problem of overfitting. The new validation method transforms time series data to independent non-time series samples and then uses hold out and cross validation to mitigate the overfitting problem. The success of the new validation method has been proved by efficient prediction results on varied forex pair datasets including USD/CNY and USD/GBP.
- The LSTM combined with the new validation method and proper forex features is designed and implemented for forex trend prediction. After extensive hyperparameter tuning, the experimental results have proved that such kind of LSTM has performed better than learning with technical data only and learning with the traditional walk-through validation method.
- The idea of hybrid activation functions (HAF) is proposed in this research, inspired by biological neural networks. When hybrid activation functions are used in the same hidden layer of the LSTM, better predictions are generated than the LSTM with only one activation function. Further studies have also shown that the ratio of the two hybrid activation functions is better to be kept at 9:1. Moreover, the HAFs have been successfully applied to different types of ANNs (e.g., LSTMs and RNNs)

and different datasets (USD/CNY and USD/GBP), demonstrating their great generalisability.

1.5 Thesis Outline

The rest of the thesis is structured as follows.

Chapter 2 reviews the relevant literature from three areas. Firstly, forex trading is reviewed. This starts with introducing the forex market, its properties, and the complexity of forex trading. Two classical forex analysis methods, i.e., fundamental analysis and technical analysis, are then introduced and forex prediction targets are discussed. Secondly, time series problems and machine learning used for time series predictions are explained, particularly those methods for forex predictions. Finally, Long Short Term Memory neural networks and their applications are introduced. Some related machine learning algorithms such as Ant Colony algorithms and Genetic algorithms, are reported as well.

Chapter 3 introduces in detail how to use LSTMs to predict foreign exchange tradings. First, the raw data used in this research is explained, including the indicators or features used by fundamental analysis and technical analysis. Then, a Decision-Tree-based feature selection method is applied to identify the influential features for forex trading, which are consequently used as the inputs for LSTM training. The prediction targets for forex forecasting are also discussed, and a suitable target is decided accordingly. This is followed by the propose of a new validation method which is specially designed to overcome the limitations of the current validation methods for time series data. Finally, a comprehensive LSTM neural network learning process is designed and developed, including pre-data processing, hyperparameter tuning and performance evaluation. Extensive experimental results have shown that the LSTM neural network combined with the new validation method and indicators from both fundamental and technical analysis have performed better than the benchmark algorithms when making forex trend predictions.

Chapter 4 proposes the use of hybrid activation functions in the same hidden layer of the LSTM neural network. Firstly, the activation functions used in LSTM are explained and the biological inspirations behind hybrid activation functions are discussed. Secondly, the most suitable activation functions to use in LSTMs are identified and the design of the proper ratio of the two activation functions used in LSTMs is investigated through extensive experiments. The experimental results have confirmed the performance improvement of LSTM neural networks when equipped with hybrid activation functions to predict for the USD/CNY trading. Finally, the successful applications of hybrid activation functions to the RNN and the USD/GBP pair have proved their generalisability.

Chapter 5 summarises the research and specifies the limitations of the research, together with some suggestions for the future work.

Chapter 2 Literature Review

2.1 Forex Prediction

It is generally accepted that the foreign exchange market (Forex) is the largest and most liquid financial market in the world. It was estimated that there were up to $5.1 trillion transactions per day in 2016 and 2017 (Moore, Schrimpf, & Sushko, 2016; Habibullah, 2017). Because of the substantial transactions, the exchange rates of different currency pairs fluctuate vigorously throughout a day (Soulas & Shasha, 2013). As a result, the forex market has an enormous impact on modern global economies in terms of economic growth, global interest rates, and financial stability (Krušković & Maričić, 2015). Hayward (2018) pointed out that, considering the high volume of daily trades, investors and financial institutions can make tremendous profits if they can speculate and predict the changes in forex exchange rates correctly. This section introduces the forex market and its properties in Section 2.1.1 and the complexity of the forex market in connection with exchange rate forecasting is discussed in Section 2.1.2.

2.1.1 The Forex Market and its Properties

The forex market is a globally decentralised sector that specialises in currency trading (Abraham & Chowdhury, 2001; Habibullah, 2017). The bank of International Settlement estimated that the daily exchange in the forex market in 2013 was about $5.3 trillion. Considering the substantial turnover rates, the forex market has been considered to be the largest trading platform in the world. When compared with the New York Stock Market, the forex market is said to be over 160 times larger (Talebi, Hoang, & Gavrilova, 2014). Over the years, the forex market was limited to central and commercial funds and also to the hedge funds as an avenue of currency trading.

Nonetheless, with increased technological growth, the forex market has become accessible to retail currency traders in recent years (Talebi, Hoang, & Gavrilova, 2014).

Despite the technological advancement and electronic revolution, Rime (2003) pointed out that the majority of the forex market properties remain the same. For instance, the market remains decentralised with continuous trading and high liquidity (Rime, 2003). A trading day in the forex market begins when dealers open the market in Asia and Australia. The trading activity then follows in Europe including the London, Frankfurt and Paris markets. When the New York market opens, the Europe market will close in the afternoon. These simultaneous market operations suggest that there is hardly any time of the day when the market formally closes, although there is a slowdown in the market activity between 19:00 and 22:00 GMT when New York traders stop and Sydney traders start to open their market (King, Osler, & Rime, 2011). The market liquidity tends to be at the highest when both the New York and London markets are open. Nevertheless, to individual currency pairs, the liquidity is often high when traders trade in their respective local times (King, Osler, & Rime, 2011).

The forex market is also characterised by another unique aspect where the US dollar is the dominant currency used in more than 75% of all spot transactions. The dominance of the dollar emphasises its role in the market when dealing with minor currency pairs where the dollar is used as a major currency (also known as the vehicle currency). For instance, a forex trade from Mexican Pesos to Japanese Yen (JPY) would take into consideration the use of two currencies where one involves the exchange of the Mexican Peso to the dollar and then exchanging the dollar into the JPY. Using a vehicle or major currency in forex trading ensures that liquidity falls within narrow margins and in the process reduces the transaction costs (King, Osler, & Rime, 2011). Within the Eurozone, 46% of the forex trades involve the Euro being used as a major currency. The other commonly used currencies include the JPY and the GBP that account for 20% and 14% of all transactions, respectively. These four currencies are called major currencies in the forex market (King, Osler, & Rime, 2011).

Over time, the conventions that govern quotations of various currencies have remained relatively stable. Most of the forex rates are presented as units of other currencies required to purchase 1 USD. However, the exception applies to the base currencies such as NZD, AUD, GBP, and EUR (that is, GBP/USD = USD per GBP). The forex rates are also presented into five significant figures with the smallest and the final digit called a pip (King, Osler, & Rime, 2011).

Cai, Chen & Fang (2012) pointed out that the CNY/USD pair continued to be one of the most important economic indexes globally too. Huang (2016) further added that, considering China's rapid growth and the increasing globalisation, the Yuan had already become a vital economy index indicator in the global markets. A recent report by the Triennial Central Bank (2016) indicated that the most traded currencies by value were ranked as USD, EUR, JPY, GBP, AUD, CAD, CHF, and CNY. In this case, the CNY was eighth-placed as one of the most traded currencies, and the report showed that the CNY had already exceeded the NZD, another one of the major global currencies. Moreover, in October 1st 2016, the CNY became a world reserve currency, meaning that it is the currency that is available in substantial amounts for institutions or governments who use it in their foreign exchange, transactions, and international investments. A reserve currency is also synonymous with hard or safe-haven currency. According to the IMF's special drawing rights, the CNY represents about 10.92% of the currency basket (Williamson, 2009; Mayeda, 2015). As a result, this makes the CNY a 3rd reserve currency after the USD and the EUR (IMF, 2018). Moreover, Huang (2016) noted that, considering the particularity of the Chinese economy, the USD/CNY exchange rates tend to exhibit diverse patterns depending on the time of the trade and the political reforms or economic policies in a given time period. In light of these facts, there is a need to research more on the USD and CNY currency pair in the forex market.

2.1.2 Complexity of the Forex Market

The literature shows that forex market prediction continues to present substantial challenges in terms of making appropriate price predictions, even with the help of historical trading data. AbuHamad et al. (2013) noted that the continued growth of liquidity and participants in the forex market had introduced more challenges in decision-making for brokers and traders, when they identified factors that were likely to influence price movements. Time series data of trading as it applies to finance is often noisy, dynamic, and inherently chaotic. The intrinsic noise of the information results from incomplete information of forex data from past market behaviour, and it is hard to fully capture the dependency between past and future prices (Philip et al., 2011). Philip et al. (2011) added that it was difficult to predict with finance time series data. The difficulty is because of market volatility, noisy market environment, and also the complex nature of the market. The above observations have also been made by Yu, Wang and Lai (2005) in their review of literature where they confirmed that it was not easy to predict the forex rates as a result of noise and volatility. Yu, Wang and Lai (2005) introduced that the limitation of conventional prediction techniques could explain the complexity in predicting forex changes. This limitation has induced interest among business practitioners and academic researchers to design reliable forex prediction models. Recently, artificial intelligence (AI) based models such as Artificial Neural Networks (ANN) have been identified as crucial in modern forecasting research (Yu, Wang & Lai, 2005). Yao and Tan (2000) also noticed that forex prediction is has become one of the popular research topics across the academic and practitioner cycles.

However, many researchers have now realised that forex forecasting poses substantial experimental and theoretical problems since the abandonment of the fixed exchange rate, deregulation of forex rate (Abraham & Chowdhury, 2001) and the uptake of the floating exchange system since the 1970s (Yao & Tan, 2000; Aliber, 2000). The uncertainty in forex prediction became even more complex with the General Agreement on Tariffs and Trade (GATT), which proposed the need to liberalise trade (Yao and Tan, 2000). Today, the forex market's uncertainty results from highly correlated psychological, political, and economic factors (Yao & Tan, 2000). Therefore, the interaction of the different factors that influence the forex market dynamics is highly complicated. Similar to the stock market, forex information displays unique seasonal cycles and trends. To get appropriate forex forecasting, designing a suitable model that can recombine, extrapolate and identify trend changes has become a significant problem (Yao & Tan, 2000). Forex forecasting is further complicated when too much noise in the forex data and the nonstationary features of the data keep changes over time (Philip et al., 2011).

As further discussed in section 2.2.3, ANNs have been proposed as more effective forecasting techniques because they are generally noise-tolerant. Moreover, ANNs are capable to learn complex data with corrupted or incomplete information. Neural networks are usually considered more flexible as they can learn from dynamic systems via retraining processes using new information (Philip et al., 2011). Based on the past literature, it becomes evident that the process of predicting forex change is complicated. To overcome the difficulties of predicting forex price movements, practitioners and researchers have developed different predicting methods. The conventional and basic forecasting methods will be discussed in the next section including technical and fundamental analysis.

2.1.3 Forex Analysis

Over the years, forex forecasting has been widely undertaken by using either technical or fundamental analyses (Yao & Tan, 2000). Fundamental analysis works to analyse the primary factors that affect forex change, and these include the level of a country's economic growth, market expectations, situation analysis and political circumstances of countries, regions and the world. In contrast, technical analysis focuses on predicting the future trends of the exchange rates by studying their past exchange rates using research methods in statistics, psychology, and other disciplines. The two analysis techniques are further discussed in the next subsections.

2.1.3.1.1 Fundamental Analysis

Fundamental analysis of forex involves using political, macro-economic, micro-economic and social factors to predict the price movement of currencies (Nassirtoussi et al., 2011). The fundamental analysis in the forex market is limited to examining the state of a country's economy in addition to assessing different factors such as employment, interest rates, manufacturing, and international trade and their relative impact on the value of the related national currency. Chatrath et al. (2014) pointed out that in fundamental analysis, analysts explored the fundamental data obtained from different sources such as economic data release and news, and then made assumptions based on the obtained data. Nonetheless, Nassirtoussi et al. (2015) elaborated that the process of automating fundamental analysis remained a challenging task due to the inadequate availability of fundamental data. Some sources of fundamental information include numeric and textualization sources such as regular financial reports and macro-economic data from governments and banks (Nassirtoussi et al., 2015). Chatrath et al. (2014) explored the potential effect that the news release had on jumps and co-jumps. 5-minute inter-day information was used for the selected currency pairs for the period from 2005 to 2010. Results indicated that news arrivals influenced currency jumps in the forex market (Chatrath et al., 2014). Between 9% and 15% of currency jumps were directly related to announcements from the U.S. government. Chatrath et al. (2014) added that news could help explain between 22% and 56% of the 5-minute jump returns.

Lui and Mole (1998) pointed out that interest rates and related news such as bond prices and monetary aggregates are some of the essential fundamental Forex analysis indicators. Interest rates refer to the value that is charged by central banks when lending money to private banks and individuals and they are central tools that are used to regulate inflation. Inflation is another vital fundament analysis indicator, and it is related to the latest inflation reports about the fluctuations in the costs of goods over a certain period. Abardanell and Bushee (1997) suggested that most of the fundamental indicators (but not all) have certain economic justification in their application to forex prediction and GDP and inflation are some of the main macroeconomic factors.

The Gross domestic product (GDP) is an essential fundamental measure in exchange rates. The level of GDP significantly differs from country to country (De Zwart et al., 2009). For this reason, analysing the relative GDP growth rate can be considered as an essential factor in forecasting forex change. The assumption is that as the higher GDP growth contributes to a high income, there will be rising monetary demands resulting in a stronger currency. De Zwart et al. (2009) postulates that investors can take long positions if there is observable GDP growth in past 12 months, and place for short positions where the GDP is lesser compared with that of the US GDP.

The other economic indicators such as the balance of payments, market correlations such as oil price ratio and gold price ratio, productivity, and purchasing manager's

Index (PMI) based on employment environment, supplier deliveries, production, inventory and new orders are important factors that influence fundamental analysis as well(Korczak, Hernes, & Bac, 2016). Consumer price index (CPI) applied in measuring household average prices and Producer Price Index (PPI) that measures the production costs of goods instead of their costs also influence fundamental analysis (Korczak, Hernes, & Bac, 2016). The PPI also includes the measurement of volatile items such as food and energy (Khan et al., 2018). Employment reports affect exchange rates too, because the level of a working population directly affects the spending patterns and economic growth (Bakhshi & Ebrahimi, 2016).

Some fundamental analysis theories, which are used by long-term foreign currency traders, link the fundamental indicators of forex with the exchange rate. These theories include the Interest Rate Parity Theory by Keynes and the theory of Purchasing Power Parity by Gustav Cassel. By definition, the Interest Rate Parity (IRP) theory seeks to clarify how the exchange rate and the interest rate levels are related. The IRP argues that investors are willing to buy whatever currency has a high-interest rate, which will push the currency up (Levich, 2011). Nwiado and LeneeTorbira (2016) indicate that the IRP assesses the equality of interest rates across different nations due to changes in bilateral exchange rates. The important economic indicators include interest rate, inflation and GDP. The IRP is an arbitrage condition that remains unchanged when the international financial markets reach a state of equilibrium. At the core of the no-arbitrage condition is the assumption that differentials in bilateral interest rates are obliterated after adjustment in expected spot rates and forward exchange rates. The singular assumption drives the entire concept of IRP in international market exchange rates (Nwiado & LeneeTorbira, 2016).

In contrast, purchasing power parity (PPP) postulates that changes in the rates of exchange between two domestic currencies depend on the local purchasing power.

The PPP asserts that the prices of the comparative goods are not influenced by the rates of exchange and the change in the exchange rate is not relative to proportional inflation neither(Hyrina & Serletis, 2010). PPP is crucial because it has remained a cornerstone for exchange rate models across international economic models, and also influenced policy implications. Since the PPP provides a benchmark for exchange rates, it affects policymakers and arbitragers (Hyrina & Serletis, 2010). As asserted by Hassan and Salim (2011), the purchasing exchange rate that is used in the PPP conversion is equal to the ratio of respective currencies (that is the reciprocal of the existing price levels).

The general findings by Chatrath et al. (2014) included that fundamentals play a central role in influencing equity and bond prices. Nassirtoussi et al. (2011) reported that the fundamental indicators could be used to predict the forex when mixing with other data. Kaltwasser (2010) added that determining the value of fundamentals largely depends on perceptions by analysts. In contrast to fundamental analysis, there is another type of analysis called technical analysis, which is discussed in the next section.

2.1.3.2 Technical Analysis

This forex analysis technique requires the application of past or historical price movements to inform decisions in making forex predictions. Taylor and Allen (1992) noted that technical analysis provides trade advice or predictions based on past prices, while analysts do not necessarily take into consideration fundamental analysis aspects or some other economic factors. In some cases, technical analysis uses some quantitative summaries on past price movements, including moving averages or oscillators (momentum indicators) (Taylor and Allen, 1992). Nowadays, many electronic trading platforms based on technical analysis, such as MetaTrader 4, have been widely used to provide forex predictions (Abednego et al., 2018). Neely (1997) pointed out that technical analysis uses historical price movements that are summed up in the form of forecast future trends and price charts to make forex predictions. During the late 1800s, Charles Dow developed the technical analysis which has been used in making professional trading and investment decisions (Neely, 1997).

In 1991, Pring (1991) studies the concepts behind technical information evaluation. The basis of technical analysis is limited to assessing the past trends of prices and that the changes are often triggered when investor attitudes are affected by psychological, political, monetary or economic forces. Consequently, Pring (1991) noticed that technical analysis is anchored on the concept of mass psychology (a crowd) in action. Therefore, technical analysis tends to predict future price movements based on crowd psychology perceptions where there are potential pessimism, panic, fear, greed, optimism and overconfidence (Pring, 1991; Neely, 1997). The technical analysis indicators used in Forex are in general grouped into four categories, namely strength or volume indicators, trend indicators, momentum indicators and volatility indicators (Yazdi & Lashkari, 2013).

Technical analysts tend to believe that future prices can be anticipated by studying historical data (Masry, 2017). This is in contrast to the random walk hypothesis, which argues that the change in price is independent of any past actions or trends. Similarly, the efficient market supports that historical price movements cannot guarantee abnormal returns. As suggested by the efficient market hypothesis, any expectations in abnormal return movement will be broken almost as soon as investors enter the market (Masry, 2017). So it is always quite difficult for an investor to deploy reliable technical predictions in the forex market.

2.1.3.3 Combined Analysis

In practice, most investors use both technical and fundamental approaches to make trading decision, although financial journalists rely more on fundamental analysis in forex prediction. Interestingly, lots of studies only use either technical or fundamental analysis alone to predict the forex, and there are very rare studies that use information from both technical and fundamental analysis for predictions, as real investors do. It is therefore worth studying intelligent techniques using combined analysis method to take advantages of both fundamental and technical analysis. This research will address the research gap, as illustrated below. Eng et al. (2008) reported that ANNs could be applied to foreign exchange rate predictions. Economic fundamental data were added to the ANN inputs and the results revealed that economic fundamentals were essential in exchange rate movements although the relationships between the fundamental factors were unclear. Nassirtoussi et al. (2011) created a new approach to fundamental data manipulation to identify the associations between external information and market behaviour. The experiments identified the potential connections between price movements in the USD/GBP currency pair and fundamental datasets. Nassirtoussi et al. (2015) developed a technique to predict intraday forex movements after the release of breaking financial news and achieved a high directional-accuracy of up to 83.33%.

Some other researchers have used technical data to investigate the possible impacts of various kernel functions that have a polynomial or radial basis on prediction performances in the forex market (Kamruzzaman, Sarker, and Ahmad, 2003). Kamruzzaman, Sarker, and Ahmad (2003) also studied the effect of regularisation parameters on Australian forex prediction. Polynomial kernel generated the best results when working on forecasting trends, whereas radial basis only produced acceptable results. Yao and Tan (2000) documented their practical results when the

ANN technique was applied to forex forecasting based time series data and technical analysis indicators. Results by Yao and Tan (2000) revealed that accurate forex forecasting could be attained and enable investors to make profits through intelligent sample analysis.

The exchange rates between CHF and USD was also studied. It is found out that it was difficult to use an efficient market to make profits when the neural network or technical indicators were used. Chan and Teong (1995) used neural networks to predict technical indicators and generate trading signals even before technical indicators can do. This approach gives traders an advantage by allowing them to change their trading strategies before executing a trade. Lee et al. (2014) used features extracted from trend patterns to create and predict the next day's trends. Hidden Markov Model was used to learn historical pattern trends and applied it to forecast the following day's trends. The historical forex data of AUS and EUD against the USD in 2011 was used in the modelling process, and the 2012 and 2013 data were applied to evaluate the model. The results showed that the new technology generated very good forex forecasting in the evaluation.

2.1.4 Price and Trend Predictions

Recently there is a strong opinion in the forex forecasting research area that predicting price changes is of little help to forex investors. Instead, to predict the change in the forex trend will be more helpful. For example, if a price prediction is made for the EUR/JPY pair at 1.48 and the investor again wants to predict the forex price five days later, the price will be about 1.46 or 1.5, implying that in both hypothetical cases, there is the same error of 0.02. Therefore, improving the accuracy of price prediction alone is not sufficient to help forex investors. Instead, forex investors can benefit more from trend predictions than from price predictions, where the primary focus is to assess whether the price is likely to go up or down. Knowing such information beforehand will help investors make essential decisions on whether they need to buy or sell their currencies in the forex market.

Brown, Mundkowsky, & Shiu (2013) sought to predict next-second price movements in the EUR/USD pair by using depth as a feature. This research attempted to overcome the limitations of short-term price movements, which were thought to follow a random walk and hence difficult to predict. The findings revealed that there was enough imbalance in depth and it is possible to make accurate price predictions in the forex market. Moreover, the researchers trained and tested a Markov Model where they demonstrated that the predictability potential could address potential losses resulting in profits. Kondratenko and Kuperin (2003) used neural networks to predict trends in foreign exchange rates. Time series data with technical indicators, e.g., the moving average, were introduced to the ANN to acquire underlying rules of forex exchange rate movements. Researchers revealed that the trained RNN was able to predict different currency pairs including the USD, EUR, GBP, CHF, and JPY. The results showed that neural networks could forecast prices with acceptable accuracy (Kondratenko & Kuperin, 2003).

Galeshchuk and Mukherjee (2017) identified the weaknesses of applying shallow neural networks to time series data when predicting the prices of long-term exchange rates. This implied that shallow neural networks were not very effective at forecasting forex trends. For this reason, using such shallow neural networks to predict forex prices cannot make a sustained profit. The researchers proposed to use machine learning classifiers and found that they generated better results in trend prediction than price prediction. The success of deep networks in predicting market changes was attributed to their capacity in learning abstract features from raw data. For instance, deep convolutional neural networks performed well in predicting exchange rates of

JPY/USD, GBP/USD, and EUR/USD currency pairs. Therefore, in this thesis, the trend will be used in the forex forecasting since it gives better exchange rate estimations when compared with price prediction.

2.1.5 Prediction Time-Period

It is very important to decide how many days after today to predict in forex trend forecasting. Different prediction time periods (forecast horizons) will result in different prediction accuracies (Yu, Wang, & Lai, 2010). Lai & Nakamori (2003) pointed out that, when predicting the price trends on the next day, the 3rd day and the 5th day, made by ANNs performed better than some other models such as random walk. On the contrary, ANNs performed worse when predicting for the 10th day and the 30th day. Chun and Kim (2004) applied the Lyapunov model to identify the best forecast horizons based on loss of information, and the result showed that the application of artificial intelligence to financial forecasting had a better performance on short term horizon. Accurate forecasts by ANNs can only be attained by using suitable forecast horizons. Generally, ANNs can give reliable forecasts when undertaking short-term and medium-term predictions.

The short term refers typically to 1 to 3 steps, medium-term refers to 4-8 steps, and long-term indicates more than eight steps. The term ‘step’ is used to refer to the frequency of information updating, which can either be in days, weeks, months, or quarters. Yu, Wang, and Lai (2010) investigated 45 articles on short-, medium-, and long-term predictions, including 39 studies on short-term predictions, 15 studies on medium-term forecasting and six studies on long-term forex predictions. Out of the 39 articles, 27 papers used both middle-term and short-term predictions. The conclusion drawn from the research reveals that using long-horizon forecasts potentially contributes to negative predictions, and short-term predictions result in better predictions (Yu, Wang, and Lai (2010).

Maknickien and Maknickas (2012) performed a 5-day prediction and obtained an accuracy of 78% when predicting for the EUR/USD day trading. Sidehabi and Tandungan (2016) undertook a performance evaluation on statistical and machine learning processes based on three time periods: 1, 5, and 30 days. For 1 and 5 days, Adaptive Spline Threshold Autoregression (ASTAR) showed better results in term of Close and High variables, because the data showed periodic patterns. Genetic Algorithm-Neural Network (GA-NN) gave contrary results, although it displayed better results when predicting Low and Open prices. When performing long-term predictions, there were conflicting results when GA-NN gave better High and Open price predictions and ASTAR worked better on High and Close prices (Sidehabi and Tandungan, 2016). Mańdziuk and Rajkiewicz (2016) proposed a neuro-genetic system for trading on the forex market for short-term investments on the EUR/USD currency pair. A small subset of the EUR/USD trading was used as the input data. Considering the complex correlations among forex features, such as forex price and technical indicators, the GA was found to be more robust for short-term time periods such as a 5-day horizon. Thomason (1999) also observed that predicting for the 5th day horizon was more suitable for daily forex trading. Similar findings were provided by Ghazali et al. (2008), in which RPNNs generated higher profits within the short-term period of 5 days, when compared with some others models such as PSNNs, FLNNs, and MLPs under similar settings.

2.2 Machine Learning for Time Series Prediction and Forex Prediction

The current literature has extensively examined the topic of forecasting and exchange rate modelling. The availability of numerous modelling techniques actually indicates the challenging nature of finding a representative model that fully describes the forex market fluctuations (Yao & Tan, 2000). Forex prediction challenges were initially reported by Meese and Rogoff (1983) when they compared sample predictions from both time series models and structural models. The researchers found that, although these models fit very well with the samples, none of them made better predictions than the random walk with root mean squared forecast error. Since 1983, many researchers and traders have tried to refine the time series models in order to improve exchange rate predictions but there is still a long way to go.

2.2.1 Time Series and Forex Time Series

According to Box et al. (2015), a time series refers to a set of observations Xt that have been recorded sequentially in time order t. Fuller (2009) defined a time series as successful data points that have been graphed, listed or indexed in time order. When numerical data are gathered in a specific sequence, at regular intervals and over a period of time, the data is called a discrete-time series (Angadi and Kulkarni, 2015). Li (1991) pointed out that time series can be regarded as white noises. One typical example is the daily prices of the stock market at the Dow Jones Industrial Average, when the market closes. Therefore, time-series data is inevitably involved in almost all forex prediction analyses.

Bleikh & Young (2014) suggested that forex forecasting based on time series should take previously recorded events into consideration. In such assessments, regression analysis was applied to test how the historical data had influenced past trends in a different time series. Montgomery, Jennings, & Kalachi (2015) indicated that, in terms of orders, there was a temporal order of the time series information that distinguishes it from the cross-sectional one, which often lacks similar natural ordering. When stochastic models are used in time series, they primarily reflect that observations which are close together are more related than those far apart from each other (Montgomery, Jennings, & Kalachi, 2015). Besides, Box et al. (2015) noticed that time series have also been widely used in different disciplines and topics, including empirical applications to stock index forecasting using support vector machines (SVM) (Huang & Wu, 2008; Kazem et al., 2013) and artificial neural networks (Kourentzes, Barrow & Crone, 2014; Kamruzzaman & Sarker, 2003).

In the forex market, the time series data structure has been shown to be an important analysis tool considering the difficulties in forecasting the forex changes (Brockwell & Richard, 2002; Kamruzzaman & Sarker, 2003; Bao, Yue, & Rao, 2017). In the past literature, there has been substantial research that has been undertaken using time series-based approaches to improve forex prediction (Pincak, 2013). Tlegenova (2015) modelled the annual exchange rates by comparing three currency pairs, namely the EUR/KZT, SGD/KZT and USD/KZT based on eight-year data from 2006 to 2014 from Kazakhstan. The researchers used the ARIMA model in predicting the annual forex rates and compared the model’s accuracy using RMSE, MAPE, and MAE. The findings from the study showed that, for the three currencies, using the ARIMA model by using MAPE was the most effective in forecasting annual currency exchange rates. Nanayakkara et al. (2014) sought to develop and compare the accuracy of using GARCH as a time series model in predicting daily currency exchange rate between the USD and Sri Lankan Rupee (USD/LKR). The moving average technical indicators and past lagged observations were taken into consideration as explanatory variables. The ANN model was found to have better performance in predicting the currency exchange rates than the GARCH model. The above studies have demonstrated that time series can provide valuable information forex prediction.

2.2.2 Time Series Forecasting

Forecasting using time series refers to using data sequences to measure the same thing over an order period (Velicer, & Fava, 2003; Chatfield, 2016). Alternatively, time series can be defined as a series of numeric values, each with its own time-stem defined by a name and a set of labelled dimensions (Angadi, & Kulkarni, 2015; Tay, 2017). In the Forex market, autonomous trading algorithms continuously collect data on market fluctuations over a period of time (Angadi, & Kulkarni, 2015). Collecting time series data enables researchers to analyse the past, monitor the present, and predict the future (Chatfield, 2016). There exist some prediction methods to predict the forex time series. Before explaining these methods, the training set and testing set need to be explained in advance. The training set is used for applying data analysis method to make a prediction, and the testing set is used for evaluating the performance of the method. Because a significant amount of data is needed for data analysis, training set usually accounts for 70-90% of total data, and testing set account 10-30%.

Firstly, the data can be analysed by using simple naïve forecast approach where the last day's value of the training set can be used to estimate the value of the following day by same value (Goodwin, 2014). The formula for the naïve forecast model can be presented as:

$$\hat{y}_{t+1} = y_t \qquad \textit{(2.1)}$$

This is the most basic time series forecasting equation. However, a mean square error test, which shows the error between the prediction and the real value in the testing data set, often shows that this naïve forecast method performs poorly in many situations, especially for those that have large variations between two time-units.

There is another fundamental time series prediction method, called simple average. The simple average method can be used to estimate the value of the next day by using the average of all of the previous days, as presented in the following equation:

$$\hat{y}_{x+1} = \frac{1}{x}\sum_{i=1}^{x} y_i \qquad (2.2)$$

However, the performance of the simple average method is even worse when there is a significant increase or decrease in the whole time period. For this reason, the simple average approach can be improved by taking the average value of a short time period in the past, i.e., the moving average. The simple moving average has shown improved predictions and it can be presented as (Johnston et al., 1999; Nau, 2014):

$$\widehat{y_l} = \frac{1}{p}\left(y_{i-1} + y_{i-2} + y_{i-3} \dots y_{i-p}\right) \qquad (2.3)$$

where p is the number of time units in the short time period.

The model can be further improved by adding more weights to the values of more recent data points, i.e., Weighted Moving Average (WMA) (Nugrahani, Adi & Suseno, 2018). The principle of WMA is that the more recent data points play a more important role in forecasting.

$$\widehat{y_l} = \frac{1}{m}\left(w_1 * y_{i-1} + w_2 * y_{i-2} + w_3 * y_{i-3} \cdots\cdots + w_m * y_{i-m}\right) \qquad (2.4)$$

When proper weights are added within a specific period and make all the values added to 1, it becomes possible to mathematically define the importance of recent data points (Nau, 2014). Both the simple average and the weighted moving average are on the opposite end of the time series methods in terms of how they approach a problem. Exponential smoothing can be used to incorporate ideas from both techniques by attaching more weights to recent data points (Nau, 2014). The simple exponential smoothing method can be presented as:

$$\hat{y}_{T+1|T} = \alpha y_t + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} \dots. \qquad (2.5)$$

where, in this case, α represents the alpha smoothing parameter. However, the simple exponential smoothing method fails to take into account the variations existed in the data. Holts Linear Trend makes it possible to apply simple exponential smoothing to forecasting the data with the trend and the level. The trend refers to the general pattern of values in a given time, and the level denotes the average value in the series (Nau, 2014). Holt's linear method improves the forecast using three equations: forecast, trend, and level equations. The specific equations can be presented as,

$$\text{Forecast equation, } \hat{y}_{t+h|t} = l_t + hb_t \tag{2.6}$$

$$\text{Level equation, } l_t = \alpha y_t + (1-\alpha)\,(l_{t-1} + b_{t-1}) \tag{2.7}$$

$$\text{Trend Equation } b_t = \beta * (l_t - l_{t-1})\,(1-\beta)b_{t-1} \tag{2.8}$$

Although the model works under some circumstances, it does not account for data variations perfectly. An additional improvement is to add seasonality since some trends depend on seasons (Gahirwal, 2013; Venkateswarlu & Sarma, 2017), called Holt's Winter Method.

$$\begin{aligned}
&\text{Level equation, } L_t = \alpha(y_t - s_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}) \\
&\text{Trend equation, } b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \\
&\text{Forecast equation, } F_{t+k} = L_t + kb_t + S_{t+k-s} \\
&\text{Seasonality equation, } S_t = \gamma(y_t - L_t) + (1-\gamma)S_{t-s}
\end{aligned} \tag{2.9}$$

So far, all these methods only take into account univariate time series data, which means the data has only single time-dependent variables. In forex prediction, however, the value of one variable not only depends on the past but also is also affected by other variables. These dependencies need to be considered for predicting future values and one common method that can be used is the Vector Autoregression (VAR) method based on the linear function of its historical values and the past values of other inputs (Chandra & AI-Deek, 2009). Machine learning techniques are often used to be trained to learn from the data sequences and LSTMs can learn from longer sequences of data (Schaefer, Udluft, & Zimmermann, 2008). In Forex prediction, the time series is multivariate data, including high, low, open and close prices, which are just suitable for machine learning techniques such as RNNs or LSTMs to learn.

The common prediction models used for multivariate time series include vector autoregression (VAR) (Chandra & AI-Deek, 2009), Recurrent neural network (RNN) (Rehman, Khan, & Mahmud, 2014), and LSTM (Heryadi et al., 2017; Kim & Won, 2018). Due to the complexity of forex, the selection of a suitable method becomes essential in obtaining appropriate prediction results when predicting forex time series. A model based on time series data can be linear or non-linear based on the nature of the previous values (He, Xie & Lai, 2008; De Oliveira, & Ludermir, 2016). Artificial intelligence presents a new statistical learning technique that has received growing attention for financial forecasting (Cao & Tay, 2003). Today, several AI-based techniques are being used in forex prediction, as discussed in the next subsection.

2.2.3 Current Methods Used for Forex Prediction

Researchers and practitioners have attempted to apply various prediction techniques in the forex market over the years. Philip et al. (2011) pointed out that the methods can be differentiated based on the techniques they hold. Classic time series prediction methods largely dominated traditional statistical models such as Box-Jenkins (Findley et al. 2002). As White (1989) noted, there is a complementary relationship between the ANN-based time series analysis and conventional statistical prediction methods. However, Refenes et al. (1994) further pointed out that traditional statistical approaches for time series prediction appear to have reached their limitations when being applied to nonlinear data such as stock indices. In efforts to address the limitations that conventional statistical methods have, Yao and Tan (2000) noted that the artificial neural network technology had demonstrated significant capabilities for complex predictions where the problem involves recognition, classification, and predictions.

For instance, Idval and Johnson pointed out that Markov models tend to be unstable when used as a trading prediction tool, especially on forex data that has many factors that influence the final results (as cited in Philip et al., 2011). Idval and Johnson (2008) also reported that Hidden Markov models hardly improve the output results. Dunis et al. (2011) investigated the stability and robust nature of three Neural Network architectures and other five traditional architectures in predicting the EUR/USD currency pair. The results revealed that Higher Order Neural Network (HONN) and Multilayer Perceptron (MLP), RNN, and Psi Sigma models had better predictions than traditional models in predicting EUR/USD over eight years.

Philip et al. (2011) developed an ANN for predicting forex change to correct the instability and uncertainty experienced with the use of forex data. In developing their model, the artificial neural network foreign exchange rate forecasting model (AFERFM), categorised it into two classes, namely data training and forecasting. When training the network, the researchers used backpropagation to approximate inputs. The input was transformed into a standard range [0,1] using the Sigmoid Activation Function (SAF). During learning the weights were allocated values between [-0.1, 0.1] at random to get outputs that were constant throughout the training process. In order to make the training process efficient, hyperbolic tangent was used in denoting the SAF and increasing the learning rate. Backpropagation was used for Feed-Forward Networks, while forecasting was done using Multilayer Perceptron Networks. The AFERFM model showed an accuracy rate of 81.2% compared with the previously best-known Hidden Markov foreign exchange rate forecasting model (HFERFM) that recorded 69.9% accuracy. Therefore, the new AFERFM model provided a better and improved outcome in predicting foreign exchange rates (Philip, 2011).

Yu, Wang, and Lai (2005) postulated a new nonlinear forex prediction technique that incorporated generalised linear auto-regression (GLAR) using ANNs to ameliorate forecasting performance and obtain reliable forex prediction findings. The effectiveness of the proposed model was assessed using individual forecasting models (ANN and GLAR) in addition to linear combination and hybrid models. The empirical results from the study revealed that the nonlinear ensemble model predicted better compared to findings obtained using other approaches in the study. The findings revealed that the nonlinear ensemble model postulated in this study can be an alternative prediction tool in forex to attain better prediction accuracies and also in improving forecasting quality (Yu, Wang, & Lai, 2005).

De Matos (1994) contrasted the reliability of using the recurrent network and multilayer feed-forward network (MLFN) based on how effective the two models could predict the Japanese Yen futures. The findings revealed that feedforward networks generated lower prediction errors. The models outperformed the simple naive model that only used the previous month's price. Kuan and Liu (1995) showed the predictive effectiveness of RNNs and MLFNs when forecasting on different forex datasets. The results revealed that the predictive stochastic complexity (PSC) criteria are suitable for comparing neural networks in predicting forex rates. By using PSC, neural network models showed the potential for better market predicting ability compared with random walk models. Hsu, Hsu &, Tenorio (1995) created a clustering ANN platform to forecast the direction of USD/DEM movement in the forex market. Their findings revealed that their new model attained better prediction results compared with other prediction methods (as cited in Chen & Leung, 2004).

Tenti (1996) used the RNN framework for determining changes in exchange rates. Their study sought to address the growing application of ANNs in forex prediction; the result showed its efficiency and profitability. The researchers compared three RNNs based models on their capacity to predict the Deutsche mark accurately. The findings revealed that all three models generated profitable forecasting in real-time trading. El Shazly and El Shazly (1999) developed a hybrid model that combined genetic algorithm and ANN training for four currencies based on 3-month spot transactions. The four currencies were the Swiss franc, Japanese yen, German mark, and the British pound. The practical findings revealed that the forecast from the model outperformed for both futures rates and forward rates in terms of precision. Leung, Chen and Daouk (2000) compared the prediction accuracy of general regression neural network (GRNNs) and MLFNs. The findings revealed GRNNs had a greater prediction outcome compared with MLFNs for different currency pairs. Zhang and Berardi (2001) used ensemble techniques in forex prediction and proposed the use of serial partitioning and systematic methods to construct ensemble models which contained various NN structures. Findings revealed that the ensemble network consistently outperformed single network architectures in Forex prediction.

Recently, better hybrid forex prediction models have been introduced where they integrate ANN methods with several burgeoning and conventional forecasting techniques such as the time series models and econometrical models to enhance the forex prediction accuracy. The existing literature has several examples of combining ANNs in forex prediction with conventional time series prediction methods, particularly the auto-regression integrated moving averages (ARIMA) method. For example, Yu et al. (2005) pointed out the need to have a nonlinear ensemble prediction platform that integrated GLARs with ANNs in improving the accuracy of forecast results. The new model was compared with ANNs and GLARs respectively, and the results indicated that forecasting using the nonlinear ensemble model gave better findings in terms of forecasting accuracy.

Tseng, Yu, and Tzeng (2002) postulated a SARIMABP model which incorporated a back-propagation (BP) neural network and a SARIMA (seasonal ARIMA) model to forecast forex data. Findings from the study revealed that SARIMABP generated better findings compared to those obtained from SARIMA. The values of MAPE, MAE, and MSE were all found to be low in the case with SARIMABP. Besides, the SARIMABP performed better than other proposed techniques, including their turning point predictions, MAPE, MAE, and MSE. Zhang (2003) developed a hybrid model that combined ANN and ARIMA models to exploit the potential strengths from both nonlinear and linear models in time forecasting series. Findings from these studies on real datasets revealed that the proposed model provided a reliable and effective way of improving forex predictions than the models used individually. Altavilla and De Grauwe (2010) observed that linear models appeared to perform better at short-term predictions, mainly when long-term equilibrium deviations were small. In comparison, nonlinear models dominated long-term horizons, mainly when deviations are large.

However, one of the limitations of these techniques is that they cannot remember long term information. As such, this study proposes to use LSTM, which can remember long-term information compared with other artificial intelligence systems. Fischer and Krauss (2018) used LSTM in predicting the movement and direction of the S&P 500 between 1992 and 2015. The researchers reported that LSTM was more effective because, unlike the linear models that were unable to perform nonlinear activation functions, LSTMs was essential in nonlinear time series where nonlinearity is learned from the massive financial data, making predictions much more accurate. Considering its advantages, LSTM is selected as a choice for forex forecasting in this dissertation and the details are explained below.

2.3 Long Short Term Memory (LSTM)

This section introduces Long Short Term Memory (LSTM). The advantage of the LSTM is attributed to its capacity to learn from complex data and fast and effective learning capabilities (Greff et al., 2017).

2.3.1 LSTM

LSTM is an RNN architecture that remembers information over arbitrary intervals and able to avoid the gradient explosion and gradient vanishing problems in RNN training to a significant degree (Wöllmer et al., 2010; Sundermeyer, Schlüter, & Hermann, 2012; Xu & Xia, 2018). Moreover, it can sufficiently make use of large data in categorising and predicting as well as in clustering evaluation (Xu & Xia, 2018). Azzouni and Pujolle (2017) noted that LSTM could effectively play a central part in categorising, processing, and predicting time series when time lags with unspecified durations were taken into consideration. One of the critical advantages of LSTM compared with traditional ANNs is that LSTM can recall past information over a long time (Salehinejad et al., 2017). RNNs address this problem as well because they contain loops in the structure that make it possible to allow past information to persist (Rawal & Miikkulainen, 2016; Kumar, Goomer, & Singh, 2018). Figure 2.1 illustrates the concept of loops that exist in RNNs.

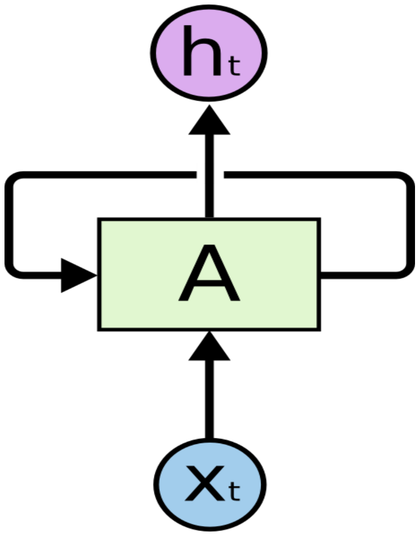


Figure 2.1 RNNs have loops (Olah, 2015)

Figure 2.1 shows a block of neural network (A), with input Xt and output ht. The loop in Figure 2 makes it possible to pass information from one step of the network to the next (Dinarelli & Tellier, 2016; Kumar, Goomer, & Singh, 2018). An RNN can be considered as a neural network of numerous duplicates, which relay specific information in sequences (Navarin et al., 2017). Figure 2.2 shows an unrolled RNN

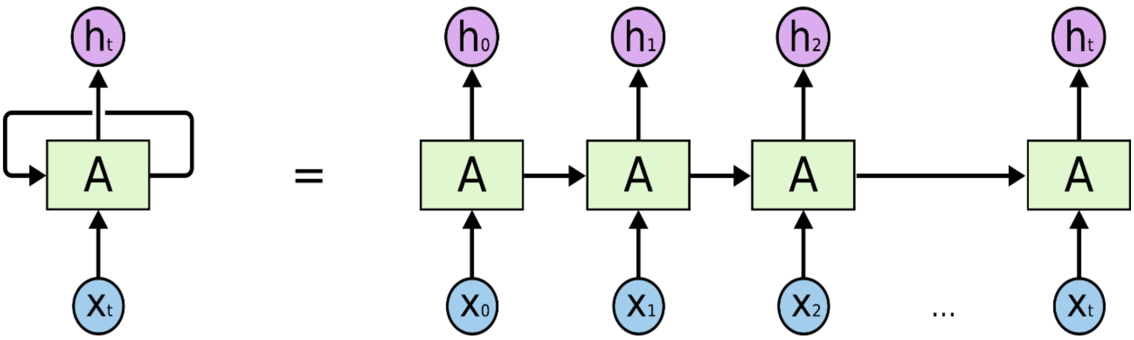


Figure 2.2 An unrolled RNN (Olah, 2015)

The chain-like nature of the RNN shows that they are closely related to lists and sequences. For such data, they are the natural architecture of ANN to use. Recently, there have been many successful applications of RNNs to various domains such as speech synthesis (Zen & Sak, 2015; Wu & King, 2016; Wu, Watts, & King, 2016; Chen et al., 1998; An, Ling, & Dai, 2017), phoneme classification (Koizumi et al., 1996;

Graves, Fernández, Schmidhuber, 2005), imagine processing (Gelenbe et al., 1997; Wang et al., 2016), language processing (Yin et al., 2017; Young et al., 2018; Zazo et al., 2016) and handwriting recognition (Doetsch et al., 2014; Messina & Louradour, 2015). LSTMs have also been used in the analysis of audio and video data (Fan et al., 2016; Ullah et al., 2018) and protein assembly prediction (Sønderby & Winther, 2014). In summary, LSTM networks present a state-of-the-art technology that can be applied to many complex problems including forex forecasting.

Central to this success is the use of LSTM that works best compared with the standard versions of RNNs (Gers et al., 2000; Gers, 2001; Bianchi et al., 2017). Almost all effective performances on RNNs are achieved using LSTMs. A main advantage of RNNs is their ability to link historical data to existing forecast tasks. However, this ability depends on the amount of missing information between the available data and the required data. When the gap is limited, it becomes easier for RNNs to learn based on the historical data they are fed. In contrast, when the gap is large, RNNs become increasingly unable to learn in terms of establishing correlations between the past, present, and future data (Lipton, Berkowitz & Elkan, 2015; Wilcox et al., 2018). The problematic learning process was examined in detail by Bengio et al. (1994) and Hochreiter (1997), which identified why it is difficult for RNNs to learn including their lack of additional information about the past to recall.

Fortunately, LSTMs do not suffer from memory loss, different from what RNNs do. LSTMs can learn long-term dependencies (Hochreiter & Schmidhuber, 1997; Zhang et al., 2018). Hochreiter and Schmidhuber (1997) first introduced the LSTM concept, which was then popularised by other researchers such as Gers, Schmidhuber, & Cummins (2000), Gers (2001) and Cummins, Gers, and Schmidhuber, (1999). LSTMs are more effective than standard RNNs, making them more suitable in many applications.

2.3.2 LSTM Applications

The literature has substantially studies the applications of LSTMs in various fields. Pang et al. (2018) postulated a deep LSTM using an embedded layer for predicting stock prices. The findings revealed that this LSTM network with an embedded layer had achieved an accuracy of 57.2% when predicting A-shares in the Shanghai stock market, and 52.4% when predicting individual stocks. Gao (2016) used LSTM to forecast stock prices from 6 different industries in the US stock market. Out of 359 stocks, the model showed an accuracy rate of 54.83%. McNally et al. (2018) used an RNN-LSTM model in predicting bitcoin prices and optimised the model by using Bayesian. The results had a 52% accuracy, and the task was achieved at different degrees of success through LSTMs and Bayesian optimised RNNs.

Chung and Shin (2018) created a new stock market prediction model and used past data to train the model. The researchers used a deep learning LSTM, considering its excellent learning capabilities. The hybrid model was based on LSTM and genetic algorithm. The LSTM architectural factors and time window size were estimated using a trial and error approach. The research investigated the sequential characteristics of the market in the Korea Stock Price Index. Findings from the study revealed that the hybrid LSTM model and genetic algorithm outperformed the benchmark model. Chen et al. (2016) designed a LSTM network to predict stock prices at Intel Corporation. The results revealed that LSTM was able to accurately forecast the following year's stock prices, especially when there was a lack of trend in the stock prices. Arévalo et al. (2016) trained a 5-layer network to forecast Apple stock prices. The results showed an 81% success when trading and 66% directional accuracy. Bao, Yue & Rao (2017) proposed a forecast model for data time series that included LSTM, wavelet transformation, and stacked autoencoders. Results showed that the network with other hidden layers performed better compared with canonical LSTMs and RNNs. Takeuchi & Lee (2013) used a 5-layer encoder to improve the trading strategy momentum and obtained about 45.93% accuracy in terms of annualised returns.

Palangi et al. (2016) created a model to address sentence embedding using LSTM cells. The results showed that the LSTM technique outperformed other web-based document retrieval systems. Bakker (2007) also used LSTMs for time series analysis and reported that the network was powerful in processing persistent information through back-propagation and time series analysis. Wang, Velswamy, and Huang (2017) used LSTM networks to improve heating ventilation through reinforcement learning. The technique improved thermal comfort in addition to optimising energy consumption by an average of 15% in terms of thermal comfort, and 2.5% in terms of energy efficiency. Wielgosz, Skoczeń, and Mertik (2017) used LSTM in the modelling of voltage-time series in magnets and found optimal results after 16-steps history buffer.

2.3.3 Evolutionary LSTM

This section discusses how evolutionary computation such as genetic algorithms (GAs) and ant colony optimisations (ACOs) has been used to evolve LSTMs, particularly for hyperparameters adjustment. Moreover, the advantages and shortcomings of using the two evolution methods in LSTM/RNN are discussed.

2.3.3.1 Genetic Algorithm

Holland (1975) proposed that a GA is a stochastic and metaheuristic optimisation model inspired by natural evolution. Armano, Marchesi, and Murru (2005) pointed out that GAs incorporate operations that imitate natural evolution and genetic principles, including mutation and crossover. The primary characteristic upon which GA is based is attributed to the chromosome population. Each chromosome is a potential solution to the identified problem, and in most cases, they are presented as binary strings. The generation of the chromosomes at the beginning is a random process, and in the process, those that can give better results are allowed to reproduce (Armano, Marchesi, & Murru, 2005). In the literature, GAs have been used in optimising LSTM/RNN networks too.

Chung and Shin (2018) used GAs in exploring the optimum architectural features (including a specific time window that can be introduced into the LSTM networks) and derives findings via the GA genetic search. The study intended to develop new stock market prediction models based on available financial information. The researchers created a hybrid approach to integrate GA and LSTM. The model was applied for estimating the window size and architectural factors of the LSTM network. The proposed hybrid model was evaluated using data obtained from the daily Korea Stock Price Index (KOSPI). The results obtained from the research found that the proposed framework that combined GA and LSTM network performed better when compared with the current benchmark models (Chung & Shin, 2018).

In particular, Chung and Shin (2018) observed that the best time window size for LSTMs was ten. The performance of the GA-LSTM network was evaluated using MAPE, MAE and MSE of the real closing prices. Similar performance measurements have also been used and reported in the literature to provide means of determining the effectiveness of the models in stock market forecasting. The prediction of the benchmark model, was only 209.45, whereas the MSE predicted by the combined GA-LSTM was 181.88, with a prediction improvement by 13.11%. For the MAE prediction, the benchmark model obtained 11.71, and the combined model had 10.21, with an improvement of 12.8% (Chung & Shin, 2018).

Kim and Han (2000) used GAs to determine and discretise the assembly weights of an ANN in addition to mitigating feature space complexity, when performing stock price predictions. Results revealed that GAs concurrently enhanced the connection weights between thresholds and layers for discretising features by 10-11% for the holdout data. The GA-evolved weights were reported to avoid the shortcomings caused by gradient descent algorithms and reduce the dimensionality of the feature space by removing irrelevant factors. As a consequence, the GA-based ANN outperformed the conventional techniques.

Rather et al. (2015) postulated a novel and a robust hybrid model that can be used in predicting stock returns. The model contained two linear models (Exponential smoothing, autoregressive moving average) and a non-linear model based on RNN. The GA was used to training the model and findings indicated that RNNs produced better predictions than the linear models. The MAE and MSE averages for the 25 stocks generated by RNNs was 0.0107 and 0.0028, respectively. For the hybrid prediction model, the average of MSE and the average of MAE were 0.0013 and 0.0047, respectively. Obviously, RNNs generated more optimal estimations compared with the linear models.

2.3.3.2 Ant Colony Optimization

ElSaid et al. (2018) explored the application of ant colony optimisation (ACO) to improving LSTM recurrent neural networks through refined cell structures. Large flight data used for training was acquired from the flight records of past airline vibrations. The results confirmed the effectiveness of the ACO-based LSTM, when about 1,000 variations of the LSTM structure were generated and evaluated. The new evolved LSTM cells were observed to have a better performance for prediction (up to 1.3%). Meanwhile, the MPE was reduced from 6.38% to 5.01% when exploring potential engine vibration problems up to next 10 seconds. The number of weights was also significantly reduced from 21,170 to 11,650. During the LSTM training, ElSaid et al. (2018) noticed that ACO was able to optimize LSTM, which helped it perform substantially better than the traditional Nonlinear Box-Jenkins (NBJ) model, Nonlinear AutoRegression with Exogenous (NARX) inputs and Nonlinear Output Error (NOE), with error rates of 9.77%, 8,47%, and 11.45%, respectively.

ElSaid et al. (2018) investigated multiple LSTM RNN structures in previous studies. LSTM RNNs were observed to give more robust and generalizable solutions than analytical assessments of engine vibrations. The limitations of analytical calculations were due to the need for prior knowledge of specific empirical engine parameters, and as a result, the technique cannot be generalised across different engines. In their study, ElSaid et al. (2018) developed a parallel version of the ACO neuroevolution model to optimise the LSTM cell structure of the most efficient LSTM RNN found in the past studies. The training process throughout the evolution of the LSTM RNN was achieved using vast flight data records obtained from aircraft that suffered from excessive vibrations. 1000 diverse LSTM structures were evolved using 168 computing cores in 4 days, resulting in a 1.35% improvement on predicting disproportionate vibrations, when the prediction error was reduced from 5.51% to 4.17%.

2.4 Chapter Summary

This chapter reviews the work related to forex forecasting. It firstly introduces the complexity of forex prediction. As a very complex time series, forex has been studied for many years, resulting in various theories and methods. For this reason, the investigation of the existing methods to predict forex prices has been conducted. Through the investigation, it is found out that current forex forecasting is mainly based on fundamental analysis and technical analysis, which are the two widely accepted categories of forex analysis. Each category utilises a different set of indicators for prediction. Both sets of indicators have been proven useful in practice, but there is no existing study on using indicators from both sets to make forex predictions, which leaves a clear research gap in the literature. In addition to indicators, different prediction targets have been discussed for forex prediction.

Secondly, the chapter reports the existing statistical models for time series data. After that, a series of studies on various prediction methods have been reviewed. The research has found that traditional statistical models cannot work well with fundamental and technical indicators. For this reason, machine learning prediction methods, which can learn with various forex features, have been reported.

Finally, machine learning methods for forex forecasting have been explained. Particularly, Long Short Term Memories (LSTM), which is most suitable for time series predictions, has been introduced, together with its applications. In order to improve its performance, LSTM has been combined with evolutionary methods including and ACO. However, further investigation on evolutionary LSTM is required for better forex forecasting.

Chapter 3 Forex Time Series Forecasting based on Long Short Term Memory (LSTM)

This chapter presents how to forecast forex trends by using Long Short Term Memory (LSTM). Section 3.1 introduces the preparatory work for LSTM neural network training, including the raw data collection and its rationale, feature selection and prediction target analysis. In Section 3.2, a newly designed validation method is introduced. At the same time, the current validation method and its limitations are explained. In the last section, the LSTM neural network and its detailed network design for forex forecasting are explained. The performance of applying LSTM to forex trend prediction is stated in this section as well.

3.1 Data Collection

3.1.1 Forex Pairs

There are nearly 200 countries in the whole world and many countries have their own currencies. Although there are some countries that use the same currency, such as European countries using EUR, there are still a massive number of currencies in use. Accurate predictions for currency exchange prices help not only forex traders earn profits, but also international companies make correct trading strategies.

This research uses the USD/CNY forex pair as an example for forex trend forecasting. With the rapid development of the Chinese economy, the currency exchange between USD and CNY has become more and more frequent (Cai et al., 2012), ever since 1978 when China resumed foreign trades re-sumed. Due to the particularity of China's economy, the mechanism of the USD/CNY exchange is different from the other major currency exchange mechanisms, mainly because the Chinese Yuan exchange rate is more independent from other currencies (Huang, 2016). There have been plenty of literature on USD/CNY forecasting, including Cai et al.,2012; Liu, 2010; Wang & Xie, 2013; Yin & Chen, 2016; Jiang & Wu, 2016 and Liu et al., 2009. This indicates that the USD and CNY exchange rate has become an important research topic in the forex market.

3.1.2 Forex Indicators and Data Collection

There are typically four prices units recorded daily; they are open price, high price, low price and close price. In general, the close price stands for the price level for a forex pair and the other units are mainly used for technical analysis. This research uses daily close prices of the USD/CNY forex pair from 01/2007 to 12/2017 as the historical data to study. To make more precise predictions, information from both fundamental analysis and technical analysis is used in this research.

3.1.2.1 Fundamental Analysis Data

As introduced in the literature review in Chapter 2, fundamental analysis is a method used for forex prediction, in which the global economic situations are analysed. There are four types of data used in fundamental analysis in this research: macroeconomic indicators, interest rate, forex pairs with six countries and volatility caused by expected events. The details for these four types of data are explained in the following sub-sections, together with their associations with forex price changes.

3.1.2.1.1 Macroeconomic Indicators

Bank of China (2008), the bank with the longest history in China, provides a list of macroeconomic indicators related to forex change, as listed in Table 3.1. This list actually covers all the macroeconomic indicators that were used in other researches. Whereas the other studies used only part of the list for forex forecasting, this research investigates all the indicators in this list as the macroeconomic indicators in order to provide a comprehensive overview and analysis of all the possible indicators involved.

To ensure authenticity, all the data of the macroeconomic indicators mentioned in this list are collected from official government or authoritative financial websites. Data on these sources have different updating frequencies, as shown in Table 3.1. It is worth noting that most macroeconomic data come with a pair, which has information from both USA and China data, because the forex pair is related to two countries. However, some data has only information from the USA, because the corresponding information from China is absent on the Internet.

Table 3.1 Macroeconomic indicators

Indicators Name	*Resource*	*Update Frequency*
Unemployment rate USA	*https://data.bls.gov/timeseries/LNS14000000*	*Monthly*
Unemployment rate CN	*http://data.stats.gov.cn/easyquery.htm?cn=C01*	*Yearly*
Nonfarm payroll employment USA	*https://www.bls.gov/web/empsit/cesnaicsrev.htm#2007*	*Monthly*
GNI USA	*https://data.worldbank.org/indicator/NY.GNP.ATLS.CD?locations=CN*	*Yearly*
GNI CN		*Yearly*
GDP USA	*https://bea.gov/iTable/iTable.cfm?reqid=19&step=2#reqid=19&step=3&isuri=1&1910=x&0=-9&1921=survey&1903=5&1904=2007&1905=2018&1906=q&1911=0*	*Quarterly*
GDP CN	*http://data.stats.gov.cn/easyquery.htm?cn=B01*	*Quarterly*
PPI USA	*https://www.bls.gov/bls/news-release/ppi.htm*	*Monthly*
PPI CN	*http://data.stats.gov.cn/easyquery.htm?cn=A01*	*Monthly*
CPI USA	https://cn.investing.com/economic-calendar/cpi-733	*Monthly*
CPI CN	http://data.stats.gov.cn/easyquery.htm?cn=A01	*Monthly*
Personal Income USA	*https://www.bea.gov/iTable/iTable.cfm?reqid=19&step=2#reqid=19&step=3&isuri=1&1910=x&0=-*	*Quarterly*

	9&1921=survey&1903=58&1904=2007&1905=2018&1906=q&1911=0	
Personal Income CN urban	http://data.stats.gov.cn/easyquery.htm?cn=B01&zb=A0501&sj=2018A	Quarterly
Personal Income CN non-urban		Quarterly
Personal Consumption Expenditures USA	https://www.bea.gov/iTable/iTable.cfm?reqid=19&step=2#reqid=19&step=3&isuri=1&1910=x&0=-9&1921=survey&1903=66&1904=2007&1905=2018&1906=q&1911=0	Quarterly
Personal Consumption Expenditures CN urban	http://data.stats.gov.cn/easyquery.htm?cn=B01	Quarterly
Personal Consumption Expenditures CN non-urban		Quarterly
Consumer Confidence Index USA	https://cn.investing.com/economic-calendar/cb-consumer-confidence-48	Monthly
Consumer Confidence Index CN	http://data.eastmoney.com/cjsj/consumerconfidenceindex.aspx?p=7	Monthly
Industrial Production USA	https://www.federalreserve.gov/releases/G17/default.htm	Monthly
Industrial Production	http://data.stats.gov.cn/easyquery.htm?cn=A01&zb=A010801&sj=201805	Monthly
Housing Starts	https://www.census.gov/construction/nrc/pdf/startssa.pdf	Monthly
PMI	http://data.stats.gov.cn/easyquery.htm?cn=A01	Monthly
NAPM	https://cn.investing.com/economic-calendar/ism-manufacturing-pmi-173	Monthly
Retail Sales Index USA	https://www.census.gov/econ/currentdata/dbsearch?program=MRTS&startYear=2007&endYear=2018&categories=44X72&dataType=MPCSM&geoLevel=US&adjusted=1	Monthly

	&notAdjusted=1&submit=GET+DATA&releaseScheduleId=	
Retail Sales Index CN	*http://www.stats.gov.cn/english/*	Monthly
Foreign trade USA	*https://www.bea.gov/newsreleases/international/trade/trad_time_series.xlsx*	Monthly
Foreign trade CN	*https://cn.investing.com/economic-calendar/chinese-trade-balance-466*	Monthly
Factory Order USA	*https://www.census.gov/econ/currentdata/dbsearch?program=M3&startYear=2007&endYear=2018&categories=MTM&dataType=NO&geoLevel=US&adjusted=1&submit=GET+DATA&releaseScheduleId=*	Monthly
Durable Good Orders USA	*https://www.census.gov/econ/currentdata/dbsearch?program=M3&startYear=2007&endYear=2018&categories=MDM&dataType=NO&geoLevel=US&adjusted=1&submit=GET+DATA&releaseScheduleId=*	Monthly
Capacity Utilization USA	*https://www.federalreserve.gov/releases/g17/default.htm*	Monthly
Current Account USA	*https://www.bea.gov/iTable/iTable.cfm?ReqID=62&step=1#reqid=62&step=6&isuri=1&6221=1&6220=1,2,3,4,5,6,7,8,9,10,11,12&6210=1&6200=1&6224=&6223=&6222=0&6230=1*	Monthly
Current Account CN	*http://data.stats.gov.cn/easyquery.htm?cn=A01*	Monthly
Business Inventory USA	*https://www.census.gov/econ/currentdata/dbsearch?program=MTIS&startYear=2007&endYear=2018&categories=TOTBUS&dataType=IM&geoLevel=US&adjusted=1&notAdjusted=1&submit=GET+DATA&releaseScheduleId=*	Monthly

3.1.2.1.2 Interest Rate

According to the Interest Rate Parity (IRP) theory proposed by John Keynes in 1923, the interest rate is an important indicator that can affect the change of foreign currencies (Nwiado & LeneeTorbira, 2016). IRP theory believes that there is a strong relationship between interest rates and the exchange rates; because a high interest rate of a currency will push up the value of the currency (Nwiado & LeneeTorbira, 2016). For this reason, the Federal Funds Rate in the USA and the SHIBOR RATE in China have been selected to indicate the interest rate levels of USD and CNY respectively. The resources and update frequencies of the interest rates in the USA and China are reported in Table 3.2.

Table 3.2 Sources of the interest rates in the USA and China

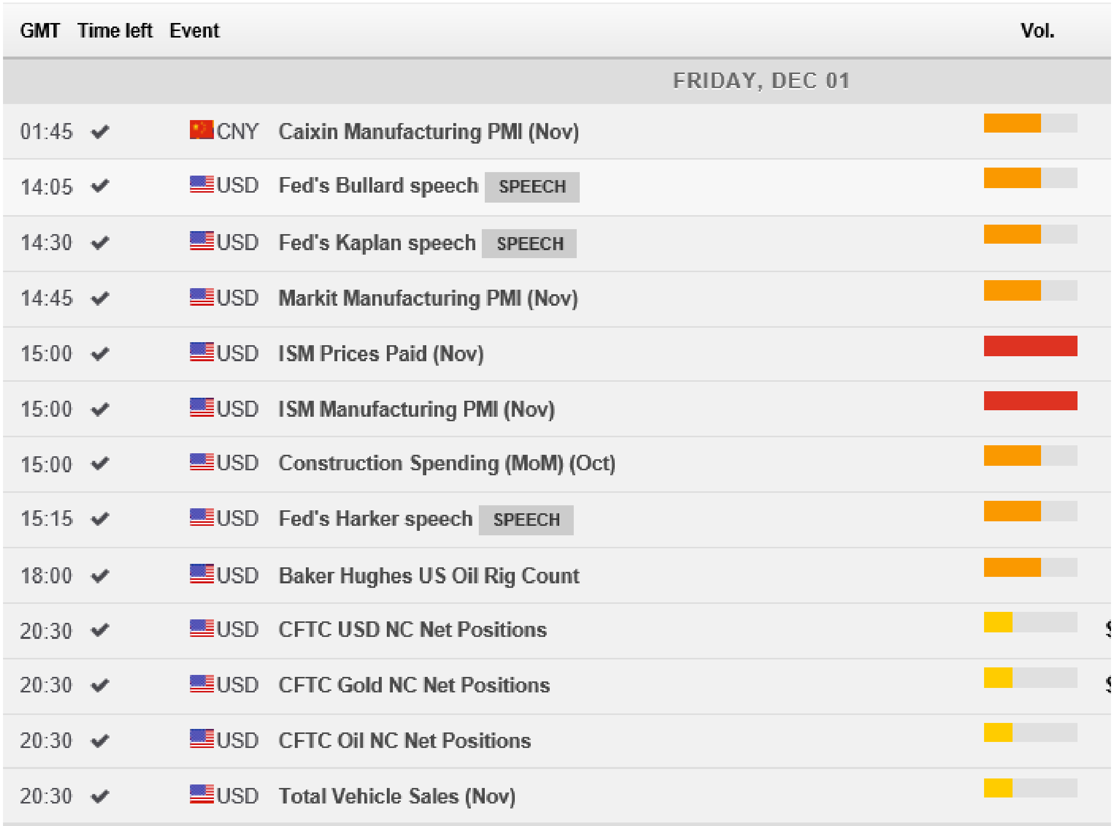
Interest rate	Data sources	Frequency
USA Federal Funds Rate	http://www.macrotrends.net/2015/fed-funds-rate-historical-chartFederal Funds Rate	Daily
CN SHIBOR	http://www.shibor.org/	Daily

3.1.2.1.3 Volatility Caused by Event Expectations

It is well-known that forex exchange is greatly affected by the occurrences of related events, such as the release of important news and economic reports. Because such news and reports are usually unpredictable, there is a belief that forex is unable to be predicted and forex prediction is a random walk problem.

Moreover, the change in currency prices, caused by news or reports, is often very swift, that is, 'fast come fast go'. This results in high volatility in the forex market, which is defined as a measure of dispersion around the mean or average return of a security (Wagner, 2018). Volatility expected, also called forecasting volatility, is an expectation of future volatility, which can be predicted by statistical methods from historical data (Satchell & Knight, 2011). In addition to statistical methods, rich practical experiences can help in predicting the actual volatility (Wagner, 2018). In this research, the expected volatility levels or importance degrees of the economic events affecting the

CNY and USD have been collected from www.fxstreet.com. Each event has its own volatility expected level, and there are four levels of volatility: ‘Very low’, ‘Low’, ‘Medium’ and ‘High’ (as shown in Figure 3.1).



GMT	Time left	Event		Vol.
		FRIDAY, DEC 01		
01:45	✔	CNY	Caixin Manufacturing PMI (Nov)	
14:05	✔	USD	Fed's Bullard speech SPEECH	
14:30	✔	USD	Fed's Kaplan speech SPEECH	
14:45	✔	USD	Markit Manufacturing PMI (Nov)	
15:00	✔	USD	ISM Prices Paid (Nov)	
15:00	✔	USD	ISM Manufacturing PMI (Nov)	
15:00	✔	USD	Construction Spending (MoM) (Oct)	
15:15	✔	USD	Fed's Harker speech SPEECH	
18:00	✔	USD	Baker Hughes US Oil Rig Count	
20:30	✔	USD	CFTC USD NC Net Positions	
20:30	✔	USD	CFTC Gold NC Net Positions	
20:30	✔	USD	CFTC Oil NC Net Positions	
20:30	✔	USD	Total Vehicle Sales (Nov)	

Figure 3.1 Examples of expected volatility levels for economic events

The total number of events at each volatility level is counted daily. For example, on 01/12/2017, there were no events that possessed an expected volatility level marked as ‘Very low’, four events that had a low level of expected volatility, seven events had a medium level of expected volatility and two events marked as ‘High’ level of expected volatility. The higher the level of expected volatility a specific day has, the more change the forex price is likely to have. The expected volatility levels and their update frequencies are listed in Table 3.3.

Table 3.3 List of expected volatility levels.

Event indicator-expected volatility level	Data Sources	Frequency
Very Low	*https://www.forexcrunch.com/live-forex-calendar/*	Daily
Low		Daily
Medium		Daily
High		Daily

3.1.2.1.4 Forex Pairs between CNY and Other Currencies

The most-traded currencies, by value, are ranked as USD, EUR, JPY, GBP, AUD, CAD, CHF and CNY (Hall-Smith, 2018). According to PPP, which was introduced in Chapter 2, the currency exchange pairs between the CNY and the other most-traded currencies can reflect the international trading situations between China and the other countries and indirectly reflect the economy in China. For this reason, the forex pairs between the CNY and the currencies of the other six countries, i.e., EUR, GBP, CAD, AUD, JPY and CHF, are used to estimate the Chinese economic (Table 3.4). The economic strength of China is reflected by its currency trading with the other most-traded currencies. The study of these forex pairs is also helpful for predicting the prices of the USD/CNY forex pair.

Table 3.4 Important forex pairs between the CNY and the other most-traded currencies.

Forex Pairs	***Data Sources***	**Update Frequency**
Close prices of EURCNY	https://www.investing.com/	Daily

Close prices of GBPCNY		
Close prices of CADCNY		
Close prices of AUDCNY		
Close prices of CNYJPY		
Close prices of CHFCNY		

3.1.2.1.5 Summary of the Fundamental Analysis Indicators

In this sub-section, forty-six indicators for fundamental analysis are collected and discussed. Among those 46 indicators, 34 are macroeconomic indicators, 2 are interest rates, 4 indicate varied expected volatility levels and 6 are the exchange rates between CNY and other currencies.

3.1.2.2 Technical Analysis Data

In the forex market, there are usually five fundamental units recorded daily as technical data, i.e., open and close prices, the highest and the lowest prices and the trading volume. However, trading volume is uncountable because the forex is an OTC (over-the-counter) market, which means trading is conducted without supervision. For this reason, it is impossible to record the total trading volume around the world. Therefore, only four prices from technical analysis are used in this research, that is, the opening price, closing price, the highest price and the lowest price in a day.

Based on these four prices, There are many derived technical indicators. Generally, technical indicators are used to analyse the relationships between different elements via statistics and aim to predict the close price of a day. As introduced in the literature review of Chapter 2, there are four groups of technical indicators, i.e., volume, trend, momentum and volatility. Because the forex trading volume cannot be collected, the other three groups of technical indicators are used in this research. Admiral Markets (2019) gave the most-popular indicators in each group, as explained below.

3.1.2.2.1 Trend Indicators

According to Admiral Markets (2019), there are five indicators grouped as trend indicators, as follows:

- **Moving Average**

A moving average (MA) is a continuously calculated value of the arithmetic mean of the price over a specified period. The formula of MA is

$$\text{N day's MA} = \text{The sum of the closing prices of past N days/ N} \qquad (3.1)$$

In forex time series, MA stands for the trend of the prices. In contrast to daily exchange prices, MA is considered to be fairly stable. The trend of MA only changes when there is a significant change in price. Furthermore, MA can be counted in different time periods, such as five or ten days. Considering the prediction target in this research, five- and ten- (working) day MAs, which stand for one week and two weeks respectively, have been calculated as one of the trend indicators.

- **Average Directional Index**

The average directional index (ADX) usually reflects the strength of price changes rather than the direction of the trend. The ADX contains two accompanying indicators,

+DI and -DI; the price is moving up when +DI>-DI and the price is moving down when -DI>+DI. Usually, when the ADX reading is below 20, the trend is weak; when the ADX is over 25, the trend is strong, and this is a good opportunity to buy or sell (Mitchell, 2019). Eq. 3.2 shows the formula for ADX:

$$+\mathrm{DI} = \frac{Smoothed{+}DM}{ATR} * 100 \tag{3.2}$$

$$-\mathrm{DI} = \frac{Smoothed{-}DM}{ATR} * 100$$

$$\mathrm{DX} = \frac{+DI - -DI}{+DI + -DI}$$

$$\mathrm{ADX} = \frac{(PriorADX*9)+CurrentDX}{10},$$

where

+DM = Current High - Previous High

-DM = Previous Low - Current Low

Smoothed +/-DM = sum of 10 periods of DM - (sum of 10 periods DM / 10) + Current DM

ATR = Average ten days True Range[2]

The time period of the ADX is usually 14, which represents two weeks. However, in forex, the market closes in weekends, so ten days ADX is used. Moreover, it is worth noting that the first ADX = sum of 10 DX/10.

In conclusion, three indicators have been used—+DI 10, -DI 10 and ADX.

- **<u>Ichimoku Kinko Hyo</u>**

The Ichimoku Kinko Hyo indicator is a diagram with six graphic elements, which are intended to help identify where the support and resistance lie. They also show whether the market is trending. The six graphic elements are explained below (Admiral Markets, 2019):

[2] Calculation of ATR will be reported in Section 3.2.2.3.1 Average True Range(ATR) with 14 periods.

Tenkan-sen: $(\text{highest high} + \text{lowest low})/2$, averaged over the last seven periods

Kijun-Sen: $(\text{highest high} + \text{lowest low})/2$, averaged over the last 22 periods

Senkou span A: $(\text{Tenkan} - \text{sen} + \text{Kijun} - \text{sen})/2$, plotted 22 periods ahead

Senkou span B: $(\text{highest high} + \text{lowest low})/2$, averaged over the last 44 periods and plotted 22 periods ahead

Chikou span: plots the current closing price, 22 periods back

Komo: the gap between Senkou Span A and Senkou Span B

Traditionally, Ichimoku Kinko Hyo used (9,26,52) as its period because, in 1930's Japan, there were six working days per week. Due to the indicator being based on (1.5 weeks, one month and two months), the period in this research has been changed to (7,22,44).

It is worth noting that the Chikou Span moves data back 26 periods to a previous time. This means that previous data use future data as an indicator. As such, the Chikou Span will not be used in this research so as to avoid using future data. Additionally, Komo cannot be represented by numbers, as it is just a graph to help the investor analyse. Therefore, Komo will also not be used.

- **<u>Moving Average Convergence/Divergence</u>**

Moving average convergence/divergence (MACD) is a trend-following and -capturing momentum indicator. The MACD was created by Gerald Appeal in the late 1970s. Ye et al. (2016) gave the calculation for MACD:

$$EMA_{12} = \frac{11}{13} \times EMA_{12} of\ last\ day + \frac{2}{13} \times the\ closing\ price\ of\ day \tag{3.3}$$

$$EMA_{26} = \frac{25}{27} \times EMA_{26} of\ last\ day + \frac{2}{27} \times the\ closing\ price\ of\ day$$

$$\mathrm{DIF} = EMA_{12} - EMA_{26}$$

$$\mathrm{DEA} = \frac{8}{10} \times DEA\ of\ last\ day + \frac{2}{10} \times DIF$$

$$\mathrm{MACD} = (\mathrm{DIF} - \mathrm{DEA}) \times 2$$

Note that

$$EMA_n = \text{SUM of the close price for past n} \frac{\text{days}}{\text{n}}.$$

As a result, three accompanying indicators will be used in this research: DIF, DEA and MACD.

■ Parabolic Stop and Reverse Indicator

The Parabolic Stop and Reverse (PSAR) indicator is used by a trader to determine trend directions and potential reversals in price. The calculation of PSAR is very complex, and, as such, Eq. 3.4—introduced by Ursell (2015)—is used as a guide for calculating PSAR.

$$\text{Rising PSAR} = \text{Prior PSAR} + [\text{Prior AF (Prior EP} - \text{Prior PSAR)}] \qquad (3.4)$$

$$\text{Falling PSAR} = \text{Prior PSAR} - [\text{Prior AF (Prior EP} - \text{Prior PSAR)}],$$

where

- AF = Acceleration Factor. It starts at 0.02 and increases by 0.02, up to a maximum of 0.2, each time the extreme point makes a new low (falling SAR) or high (rising SAR)
- EP = Extreme Point. The lowest low in the current downtrend (falling SAR) or the highest high in the current uptrend (rising SAR)

As an important indicator, PSARis used in this research.

3.1.2.2.2 Momentum Indicators

There are three indicators categorised as momentum indicators, reported by Admiral Markets (2019). They are listed and explained below:

- **Relative Strength Index**

The relative strength index (RSI) is a momentum indicator which measures the magnitude of the change of price (Chen, 2019). In this research, RSI has been calculated in a 5- and 14-day period. For the first RSI, the formula is

$$\text{RSI}(\text{STEP one}) = 100 - [\frac{100}{1+\frac{Ave\ Gain}{Ave\ Loss}}]. \tag{3.5}$$

The Ave Gain or Ave Loss is the average percentage gain or loss, respectively, during a previous period. The positive values are used to calculate the average losses. After the first RSI, the later RSI calculated as

$$\begin{aligned} &\text{N days RSI}(\text{STEP two}) \\ &= 100 - [\frac{100}{1+\frac{Previous\ Ave\ Gain \times (N-1) + Current\ Gain}{Previous\ Ave\ Loss \times (N-1) + Current\ Loss}}] \end{aligned} \tag{3.6}$$

- **Stochastic Oscillator**

A stochastic oscillator is a momentum indicator comparing a particular closing price to the closing price in a period. Hayes (2019) gives the formula for Stochastic Oscillator:

$$\%\text{K} = \left(\frac{C-L14}{H14-L14}\right) \times 100, \tag{3.7}$$

where

C is the latest closing price;

L14 is the lowest price in the past 14 days;

H14 is the highest price in the past 14 days;

%K is the slow stochastic indicator.

Another fast-stochastic indicator is called %D = 3 days MA of %K. When the %K crosses through %D, it indicates transaction signals. Both %K and %D are used in this research.

- **Williams % range**

The equation of Williams % range (%R), used to indicate the entry points and exit points, is represented as Eq.3.8:

$$\text{Williams \% R} = \tag{3.8}$$

$$(\text{Nth High} - \text{Close price})/(\text{Nth High} - \text{Nth Low})\ \text{x}\ -100,$$

where 'Nth High' stands for the highest high, and 'Nth Low' stands for the lowest low. In this research, N=14 will be used.

3.1.2.2.3 Volatility Indicators

There are also three indicators grouped as volatility indicators reported by Admiral Markets (2019). They are listed and explained below:

- **Average True Range**

The average true range (ATR) is used to show the market volatility. It is a measure of volatility introduced by Wilder (1978). The function is shown below:

$$\text{ATR (first)} = \frac{1}{N}\sum_{i=1}^{n} TR_i \tag{3.9}$$

$$\text{ATR} = \frac{Previous\ ATR \times 13}{14} + \frac{TR}{14},$$

where

$\text{TRi} =$

$\max[(\text{high} - \text{low}), \text{abs}(\text{high} - \text{previous close}), \text{abs}(\text{low} - \text{previous close})]$

$\text{n} = \textit{the time period employed (which is } 14 \textit{ days in this research)}.$

- **Bollinger Bands**

Bollinger Bands (BBs) can be applied in all financial markets and used in most time frames. They consist of three lines: MA and two lines which are the double values of standard deviation away from MA. The formula of Bollinger Bands (Bollinger, 1992) is shown in Eq. 3.10 and a period of 20 days is used:

$$\text{Upper band} = \text{MA} + 2 \times \text{Standard deviation} \quad (3.10)$$

$$\text{Middle band} = \text{MA}$$

$$\text{Lower band} = \text{MA} - 2 \times \text{Standard deviation},$$

where standard deviation = $\sqrt{\frac{\sum_{j=1}^{n}(X_j - MA)^2}{n}}$, n=20

As a result, there will be three indicators, i.e. Upper, Middle and Lower bands, to be used from BBs.

■ **<u>Sample Standard Deviation</u>**

The sample standard deviation (SSD) is used to measure the variation between the current price and the mean. SSD is a statistical measurement in finance to reflect past information. Different from the standard deviation, SSD uses N − 1 instead of N.

$$\text{Sample standard deviation} = \sqrt{\frac{\sum_{j=1}^{n}(X_j - MA)^2}{n-1}}, \text{n=20} \quad (3.11)$$

3.1.2.2.4 Technical Analysis Indicators Summary

As the basis of the technical analysis indicator, the Open, High, Low, Close and Change prices have also been considered as shown in Table 3.5. So in total there are 24 indicators from the technical analysis which can be used in this research (shown in

Table 3.6).

Table 3.5. Four basic prices of forex

The basic unit for USD/CNY price	**Close USDCNY**
	Open USDCNY
	High USDCNY
	Low USDCNY

	Change

Table 3.6 The technical indicators used in this research

Group of technical indicators	Technical indicator	Accompanying indicators
Trend indicators	Moving Average	5 days MA
		10 days MA
	Average Directional Index	10 days +DI
		10 days -DI
		Adx
	Ichimoku Kinko Hyo	Tenkan-sen
		Kijun-sen
		Senkou span A
		Senkou Span B
	Moving Average Convergence/Divergence	DIF
		DEA
		MACD
	Parabolic Stop and Reverse Indicator	PSAR
		PSAR TREND
Momentum	Relative Strength Index (RSI)	5 days Relative strength
		14 days Relative strength
	Stochastic Oscillator	%k
		%d

	Williams % range (%R)	%R
Volatility	Average True Range (ATR)	14 days ATR
	Bollinger Bands (BB)	Middle Bollinger band
		Upper Bollinger band
		Lower Bollinger band
	Sample Standard Deviation	Standard deviation

3.1.3 Feature Analysis and Selection

3.1.3.1 Feature Analysis

This research believes that using both fundamental and technical indicators will improve the performance and reliability for forex prediction. The indicators used in fundamental analysis can be deemed as the causes for price changes in forex; and the indicators used in technical analysis are the results of continuous price changes, which contain statistical analysis of the past already.

3.1.3.2 Feature Selection

There are 75 features/indicators in total when putting all the fundamental and technical analysis indicators together, which are all considered to have a connection with forex analysis and prediction. These features, however, have varied functionalities and usefulness in forex prediction so it is essential to investigate, first, which features are the most influential. Moreover, such a large set of features will affect the following ANN training so feature selection needs to be applied to deal with such high dimensional data.

Feature Selection is a critical step in data analysis (Asaithambi, 2018), because it can reduce data dimensionality. Data with high dimensionality often causes problems in practice, such as

- taking longer time for training, and
- overfitting problems

The purpose of feature selection is to remove the features which may have little use in the system operation. Fewer features may help reduce the complexity of a system and mitigate the overfitting problem.

'*The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.*'

----- Guyon and André, 2003

There are three types of feature selection methods: Filter, Wrapper and Embedded methods. The Filter method measures the associations between the features and system output (e.g., prediction target), by calculating the variance of each feature. The Wrapper method selects the features by using Sequential Feature Selection, by adding a feature each time to test the system performance. The Embedded method include L1, L2 regularisation methods and Decision-Tree-based method, which calculates the importance score of each feature and selects the features with higher importance scores (Grabczewski & Jankowski, 2005). Compared with all other methods, the Decision-Tree-based feature selection method is well known for its ability to select the features which are important for classification (Grabczewski & Jankowski, 2005). In this research, the forex trend prediction belongs to the classification problem, and the main reason for using the Decision-Tree-based feature selection is that it can discover nonlinear relationships and interactions between variables. There are other advantages of Decision-Tree-based feature selection, such as requiring less effort for data preparation and being easy to interpret and explain (Floares et al., 2016).

In this research, Decision-Tree-based feature selection is achieved by using Decision-Tree-based estimators from sci-kit-learn, which is a machine-learning library (Pedregosa et al., 2011) and 'SelectFromModel' from sci-kit-learn to select features.

In order to implement Decision-Tree-based feature selection, two pieces of code are used from sci-kit-learn:

```
clf = ExtraTreesClassifier()
clf = clf.fit(X, Y),
```

where X is the feature set and Y is the target which needs to be calculated from features. In this research, all 75 features collected in Section 3.1.1 are used as inputs and the forex trends are the targets, which can be used to measure the importances of different features. Examples of X and Y are shown in Table 3.7.

Table 3.7. Examples of feature selection data

	X					Y
Date	Feature 1	Feature 2	... (35 data)	Feature 38	Feature 39	Trend (Target)
25/12/2017	7.7997	7.8051	... (35 data)	7.7973	0.007696	-1
26/12/2017	7.796	7.7986	... (35 data)	7.7931	0.007725	-1
27/12/2017	6.5418	6.5598	... (35 data)	6.5568	0.017958	1
28/12/2017	6.5542	6.5598	... (35 data)	6.5334	0.018775	0

By applying Decision-Tree-based feature selection to the 75 features, the importance score of each indicator is listed in Table 3.8.

Table 3.8. The importance score of each indicator

Index name	**Importance score**
Unemployment rate USA	**0.00649246**
Unemployment rate CN	**0.00205029**
Nonfarm payroll employment USA	**0.00936303**
GNI USA	**0.00180899**
GNI CN	**0.00444975**
GDP USA	0.00378036
GDP CN	0.00430666
PPI USA	0.00566725
PPI CN	0.00486067
CPI USA	0.00426461
CPI CN	0.00448945
Personal Income USA	0.00456449
Personal Income CN urban	0.00389173
Personal Income CN non-urban	0.00334244
Personal Consumption Expenditures USA	0.00378587
Personal Consumption Expenditures CN urban	0.0058493
Personal Consumption Expenditures CN non-urban	0.00436745
Consumer Confidence Index USA	0.00524387
Consumer Confidence Index CN	0.00691644
Industrial Production USA	0.00737917
Industrial Production	0.00849877
Housing Starts	0.00728903
PMI	0.00727565
NAPM	0.00542057

Retail Sales Index USA	0.00516431
Retail Sales Index CN	0.00579099
Foreign trade USA	0.00576476
Foreign trade CN	0.0066199
Factory Order USA	0.00508577
Durable Good Orders USA	0.00542791
Capacity Utilization USA	0.00756163
Current Account USA	0.00368812
Current Account CN	0.00675397
Business Inventory USA	0.00514179
USA Federal Funds Rate	0.01434494
CN shibor	0.02122336
Very low volatility expected level	0.00755902
Low volatility expected level	0.0144256
Medium volatility expected level	0.01573557
High volatility expected level	0.0163292
OPENUSDCNY	0.01766726
HIGHUSDCNY	0.02497798
LOWUSDCNY	0.0210636
change	0.01728699
EURCNY	0.01853005
GBPCNY	0.0250449
CADCNY	0.01984544
AUDCNY	0.01846976
CNYJPY	0.01970475

CHFCNY	0.0160411
5-day MA	0.0199682
10-day MA	0.02040288
+DI10	0.02163546
-DI10	0.02386905
Adx	0.02104654
Tenkan-sen	0.01818315
Kijun-sen	0.01626759
Senkou span A	0.02037011
Senkou Span B	0.01708821
Dif	0.02381489
Dea	0.02307371
macd	0.02357935
PSAR	0.02022879
PSAR TREND	0.01080886
relative strength5	0.02076135
relative strength14	0.02746579
%k	0.02321379
%d	0.02166815
%R	0.01977334
Middle Bollinger band	0.02281278
upper Bollinger band	0.01881956
lower Bollinger band	0.02153124
standard deviation	0.02487522
USDCNY	0.02449895
TR14	0.02363601

There is no a standard rule in the literature about how to choose a threshold (the lowest importance score) for feature selection. Thresholds are usually decided depending on the applications and most often the designer's experiences. In this research, 0.01434494 has been chosen as the threshold to distinguish important and non-important features because there is an obvious gap between 0.01080886 to 0.01434494, as shown in Figure 3.2, which ranks all the importance scores of 75 features from low to high.

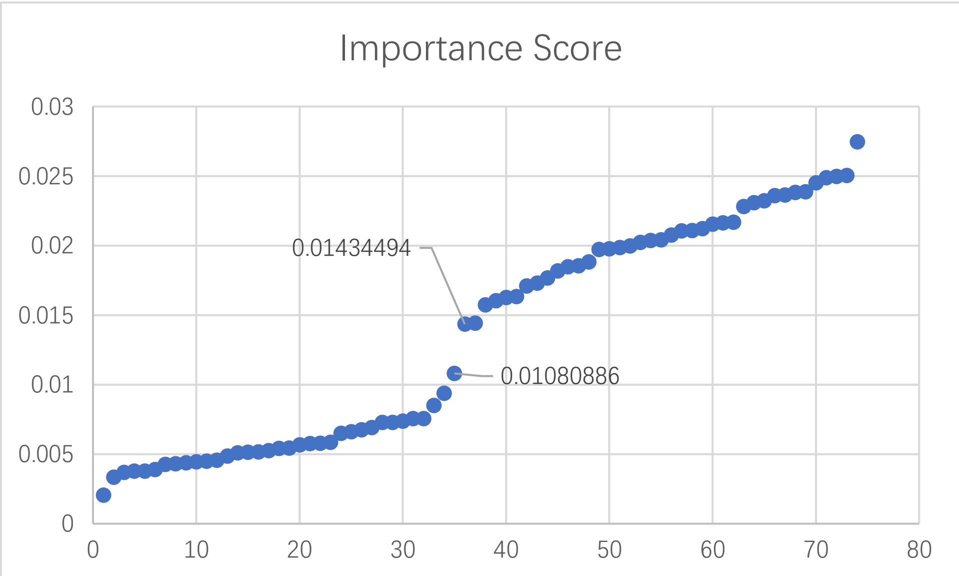


Figure 3.2 Plot of ranked importance scores

Consequently, 39 features show higher importances ranging from 0.01434494 (USA Federal Funds Rate) to 0.02746579 (14-day relative strength), whereas the others have much lower importances ranging from 0.00180899 to 0.01080886. As a result, the selected important features (as listed in Table 3.9)are used as the inputs for forex forecasting in the remainder of the thesis.

Table 3.9. Selected indicators used in this research

Selected indicators used in this research		
USA Federal Funds Rate	Close price AUDCNY	Dea
CN shibor	Close price CNYJPY	Macd
Low volatility expected level	Close price CHFCNY	PSAR
Medium volatility expected level	5 day MA	relative strength5
High volatility expected level	10 day MA	relative strength14
Close price USDCNY	+DI10	%k
Open price USDCNY	-DI10	%d
High price USDCNY	Adx	%R
Low price USDCNY	Tenkan-sen	Middle Bollinger band
Change USDCNY	Kijun-sen	Upper Bollinger band
Close price EURCNY	Senkou span A	Lower Bollinger band
Close price GBPCNY	Senkou Span B	Standard deviation
Close price CADCNY	Dif	TR14

3.1.4 Prediction Target Analysis

The aim of forex forecasting is usually to provide a highly accurate forex prediction for better trading. Most of the existing work on forex prediction is to forecast the exact price for the next day (Pujari et al., 2018; Pincak, 2013). However, according to Jakob Aungiers, who is the ex-Vice President in HSBC Asset Management's Global Investment Strategy team and the Founder & CEO of Altum Intelligence, ‘Running the data on a single point-by-point prediction gives something that matches the returns pretty closely. However, this is deceptive!’ (Aungiers, 2016, p. 1). This is because, in the so-called point-by-point prediction, there is always a strong correlation between the current price and the previous price — that is, predict price(t) ≈ actual price(t-1); hence, the next day’s price prediction easily ‘sticks’ to the previous day’s price (Figure 3.3).

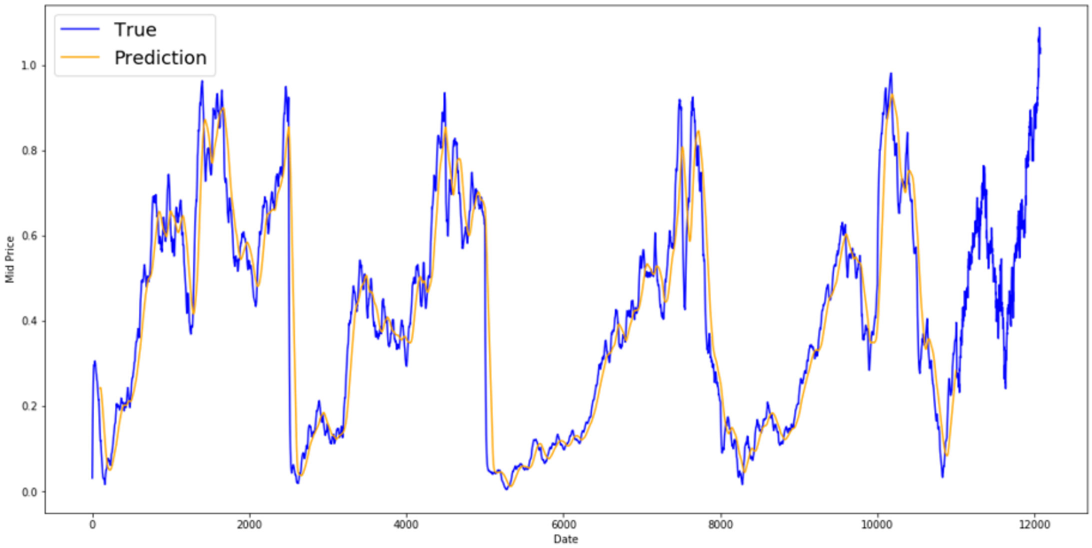


Figure 3.3 An example of the point-by-point prediction (Ganegedara,2018)

The reason for the point-by-point prediction problem can be explained in two ways:

1. Aungiers (2016) explained that the prediction point was based on the entire set of previous data. For this reason, the prediction methods usually do not need to learn the information or pattern in the time series; they just learn that the prediction point needs to be close to the previous point.

2. This problem can also be explained mathematically. When using machine-learning, the Mean Squared Error (MSE) K.mean(K.square(y_pred - y_true) or Mean Absolute Error (MAE) K.mean(K.abs(y_pred - y_true) are generally used as loss functions in neural networks when predicting the next value. The process of training a neural network is the process of optimizing the loss function. Because of the relatively low

fluctuations in the forex prices of two consecutive days, it is easier to get a very low loss value we predict price(t) = price(t-1).

Very recently, there was a suggestion to predict the trend of forex rather than precise forex prices, which would be more practical and more useful (Admiral Markets, 2019). Usually, traders are not interested in accurate exchange rates (Kondratenko & Kuperin, 2003). Instead, they are more interested in the movement directions of exchange rates or forex trends, which is the focus of the research here.

There are three possible outputs when predicting the future trend of a forex pair: ‘increase’, ‘decrease’ and ‘no change’. Financial data has a particular property—the more frequently the used data is, the worse the signal/noise ratio is (Kondratenko & Kuperin, 2003). That is, the forecast can be worse when predicting daily data, compared with predicting weekly data, because there are more noises presented in the daily data. This has also been proven by our experiments, when using LSTMs to predict the forex trend of the next day, which was only 49.3% accurate. The prediction of the weekly trend, however, has generated more positive results. The fifth-day prediction accuracy (five working days in a week) was 78.28% and the tenth-day prediction accuracy was 86.6%.

Figure 3.4 shows the prediction accuracy against the number of the predicted days in the future, by using the box and whisker plot. A box and whisker chart is widely used in statistics, demonstrating the distribution of data by showing the maximum, minimum, mean and 25% and 75% of the data range.

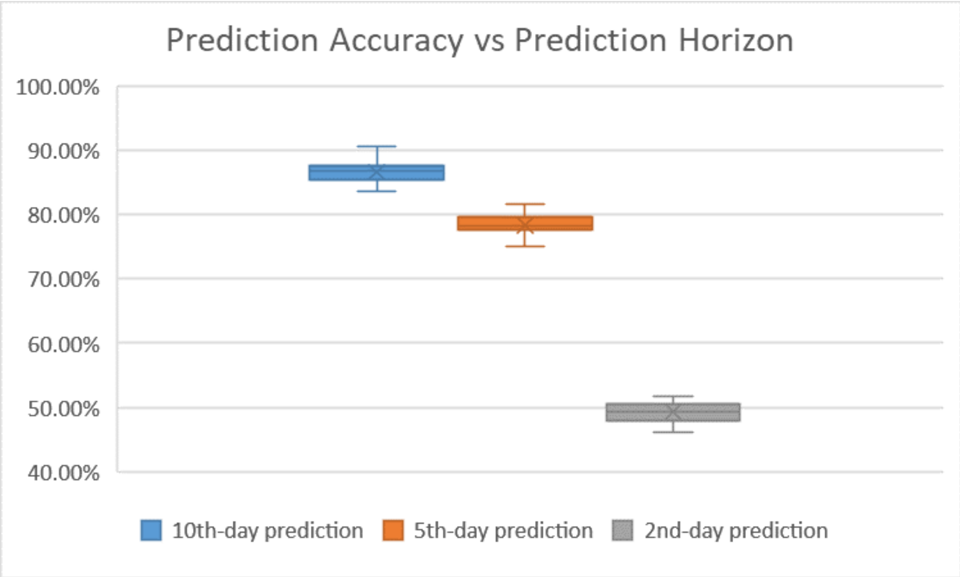


Figure 3.4 Prediction Accuracy vs Prediction Horizon

The result shows that the longer the forecasting horizon, the higher the prediction accuracy. However, a long forecasting horizon will lead to a low trading frequency and, hence, probably less profits. The prediction frequency means the numbers of predictions run in a fixed time period. In general, if the prediction result is that the trend will go up, traders will buy the low-price forex pair and sell it after a few days to earn a profit. On the contrary, if the prediction result is that the trend will go down, traders can 'go short' the forex pair to earn a profit. As a result, there is only one trading in a forecasting horizon, so there are limited times of trading in a fixed time length of forecasting. For example, when comparing the fifth-day prediction with the tenth-day prediction, the former is twice as many trading times as the latter in a certain time period. Usually, the prediction horizon is decided depending on the trader's trading strategy. In this research, the fifth-day prediction has been chosen as the prediction horizon.

3.1.5 Raw Input Data

As explained in Section3.1.3.2, 39 features are selected for forex forecasting in this research. The data of forex trading from 12/01/2007 to 28/12/2017 has been collected, resulting in 2,860 values for each feature. Table 3.10 shows some example values of the feature “Open price”.

Table 3.10. Example data of feature “Open Price”

Date	Open Price
12/01/2007	7.7997
13/01/2007	7.796
…..(2856 data)	…..(2856 data)
27/12/2017	6.5418
28/12/2017	6.5542

For 39 features, there are 2,860 * 39 data in total, as illustrated in Table 3.11.

Table 3.11. Raw data with all features

Date	Feature 1	Feature 2	... (35 data)	Feature 38	Feature 39
12/01/2007	7.7997	7.8051	... (35 data)	7.7973	0.007696
13/01/2007	7.796	7.7986	... (35 data)	7.7931	0.007725
… (2856 data)	... (2856 data)	... (2856 data)	... (2856 * 35 data)	... (2856 data)	... (2856 data)

27/12/2017	6.5418	6.5598	... (35 data)	6.5568	0.017958
28/12/2017	6.5542	6.5598	... (35 data)	6.5334	0.018775

The Closing prices of USD/CNY from 12/01/2007 to 28/12/2017 are used to mark the trend of this forex pair. The trend from the current day to the fifth day is processed as the output data. In this study, there are three kinds of trends: up, down and same (Table 3.12), corresponding to increase', 'decrease' and 'no change' as introduced before.

Up: the closing price of the fifth-day – the closing price of the current day >0 (marked as 1)

Down: the closing price of the fifth-day – the closing price of the current day <0 (marked as -1)

Same: the closing price of the fifth-day – the closing price of the current day =0 (marked as 0)

Table 3.12 Prediction target

Date	Trend
12/01/2007	-1
13/01/2007	-1
…..(2856 data)	…..(2856 data)
27/12/2017	-1
28/12/2017	-1

3.2 New Validation Method for Time Series training

In supervised learning (e.g., ANN training and evaluation), the training data set needs to be split into three sub-sets: training set, validation set and test set. The training set is used to train the machine learning method such as artificial neural networks, i.e., updating their weights and biases through the backpropagation algorithm. The validation set is used to tune the hyperparameters when designing the neural networks and selecting the best ANN structure for training. The test set is to evaluate the performance of the trained neural network on unseen data. Because a time series is a sequence of data, there is a big difference in data splitting between time series and non-time series data. This section starts by introducing the current methods for data splitting for both time series data and non-time series data.

3.2.1 Validation Method for Non-Time Series Data

There are two general validation methods for non-time series data. They are holdout validation and cross validation.

3.2.1.1 Holdout Validation

Holdout validation method (Arlot & Celisse, 2010) initially split the whole data set to a training set and a test set. The ratio of the sizes of the training set and test set is 9:1, 5:1 or 2:1 generally. Then the training set is split into a sub-training set and a validation set. The ratio of the sizes of the sub-training set and validation set is also 9:1, 5:1 or 2:1 generally (Figure 3.5). Worth to note that, the sub-training set is usually called a training set in most research.

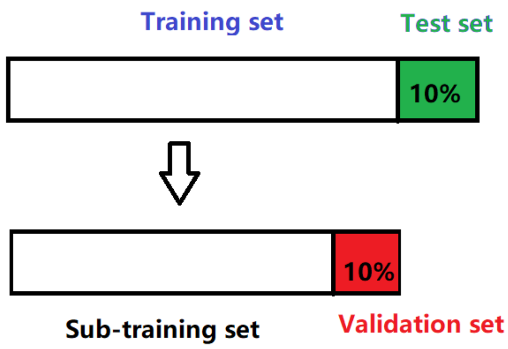


Figure 3.5 Hold-out validation

The holdout validation method is a widely used validation method, but it has its own limitations: it is more suitable for large data sets because this method only uses a part of the data to represent the performance of the whole dataset.

3.2.1.2 K-fold Cross Validation

Due to the limitations of the holdout validation method, another validation method, K-fold cross validation has been used nowadays. The letter ‘K’ in K-fold cross validation refers to divide a dataset into K equal parts.

‘The choice of k is usually 5 or 10, but there is no formal rule. As K gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller’ ------- Kuhn & Johnson, 2013, p. 70

The experiment needs to run K times by using every part of the data set once as a test set (Figure 3.6). The test set is the set only used after training; it is used to test the performance of the algorithm on unseen data. Except the test set, the rest of the dataset is used as a training set in the experiments.

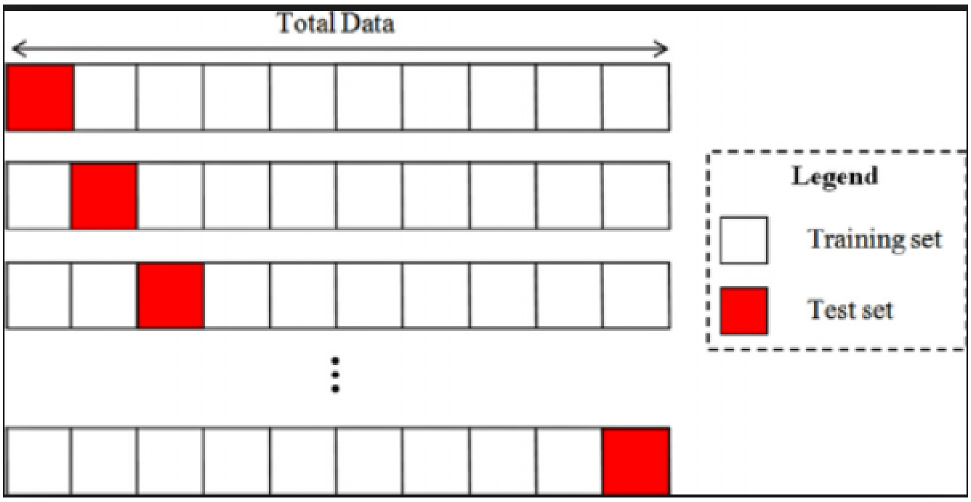


Figure 3.6 10-fold cross-validation

In general, cross-validation can be used to evaluate the performance of an algorithm (Rahman, 2017) and is especially suitable for a dataset which has limited samples. Compared with the traditional holdout validation, the K-fold cross-validation allows each part of the dataset to be tested. Moreover, K-fold cross-validation is useful to avoid the over-fitting problem when the dataset is small. Cross-validation not only can evaluate the performance of an algorithm, but can also be used for tuning hyperparameters. When it is used to tune hyperparameters, the whole data set needs to be splitted into a training set and test set at first, and then cross validation can be applied to the training set to divide it into sequences of k equal parts of training and validation (Figure 3.7).

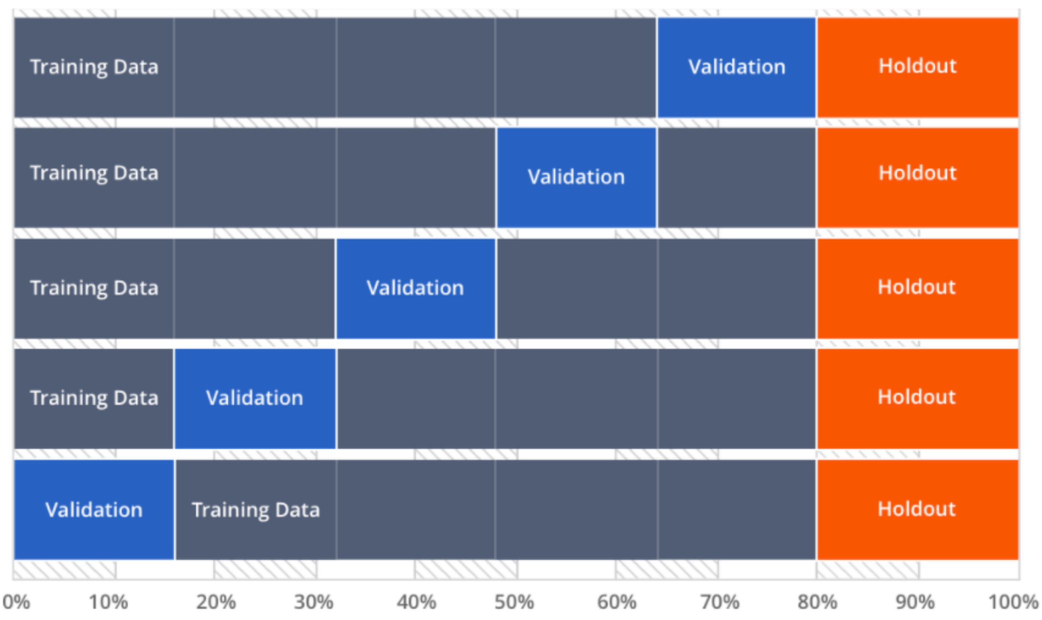


Figure 3.7 5-fold cross validation for hyperparameter tuning

3.2.2 Validation Method for Time Series

Both holdout validation and cross validation methods assume that all the observations are independent, but this assumption doesn't apply to time series In time series, the observations have an order and they cannot be shuffled. To keep the order of observations, the walk-forward validation method is widely used for time series. The walk-forward validation method is inspired by cross validation. So it is also called cross validation in time series data. The basic idea of walk-forward validation is o simulate the real situation of making a prediction, that is, to predict only a one-time point at each time and add this time point to the training set for the next prediction.

The detailed process of walk-forward validation can be explained in Figure 3.8. The blue points are the training data, and the red points are the test data. After each training, the red point is added to the blue set as new training data form a new training set for the next training.

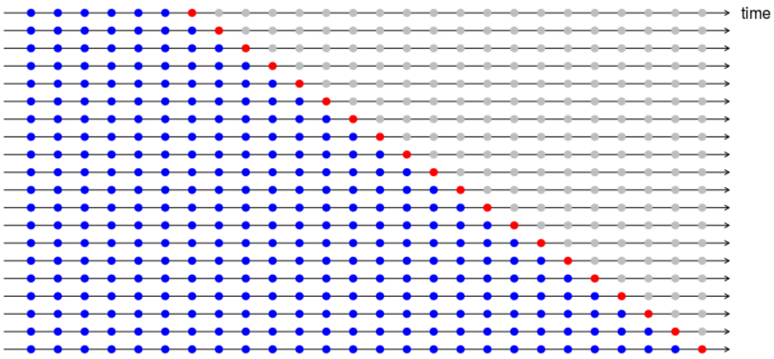


Figure 3.8 Walk-forward validation

3.2.3 The Limitations of Walk-forward Validation

There are three limitations of walk-forward validation,

1. The walk-forward validation requires extremely long time to evaluate an algorithm or a method. For example, if there are 100 time points to be evaluated, the training and evaluation experiments need to be repeated for 100 times to get the average performance for training on all time points. In general, there are thousands or even more time points in a time series. Spending a long time for validation is unreasonable.
2. Moreover, the hyperparameter tuning often needs to be done at the same time. It is well known that tuning hyperparameters is a very tedious process. Usually, it takes a week or even one month to obtain reasonable values for all hyperparameters. By using walk-forward validation, the time of tuning hyperparameters might be extended to a couple of months.
3. Walk-forward validation uses time points to compose a test set. As a result, the accuracy cannot be calculated by just one prediction. For this reason, many

techniques based on the performance of the validation set cannot be used in time series, such as early stopping.

As a result, a new validation method for LSTMs has been developed, as explained in the next section.

3.2.4 New Validation Method for LSTMs

The new validation method is specially designed for LSTMs or other RNNs. Basically, there are three steps:

Step 1: Transform the time series data to the proper format for supervised learning. Such a transformation generates proper input and output samples and makes all samples independent with each other. The following gives an example to help understand the transformation.

Table 3.13 is a multiple-feature time series:

Table 3.13 Time series

Time	Feature A	Feature B	Feature C	Feature D
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10

There are 10 observations in total and there are 4 features in each observation. If the prediction target is the value of feature D at the next time point, the time series and the target should be a list like below:

Table 3.14 Time series with a target

Time	Feature A	Feature B	Feature C	Feature D	Target
1	A1	B1	C1	D1	D2
2	A2	B2	C2	D2	D3
3	A3	B3	C3	D3	D4
4	A4	B4	C4	D4	D5
5	A5	B5	C5	D5	D6
6	A6	B6	C6	D6	D7
7	A7	B7	C7	D7	D8
8	A8	B8	C8	D8	D9
9	A9	B9	C9	D9	D10
10	A10	B10	C10	D10	D11

If the next prediction is dependent on a short time series in the past (e.g., the last 3-time steps), such a short time series and the prediction target then compose a sample. Table 3.15 shows the samples for 3-time steps training, derived from Table 3.14.

Table 3.15 Samples for 3-time steps training

Number of sample						
Sample 1	Time	Feature A	Feature B	Feature C	Feature D	Target
	1	A1	B1	C1	D1	D4
	2	A2	B2	C2	D2	
	3	A3	B3	C3	D3	
Sample 2	Time	Feature A	Feature B	Feature C	Feature D	Target
	2	A2	B2	C2	D2	D5
	3	A3	B3	C3	D3	
	4	A4	B4	C4	D4	
Sample 3	Time	Feature A	Feature B	Feature C	Feature D	Target
	3	A3	B3	C3	D3	D6
	4	A4	B4	C4	D4	
	5	A5	B5	C5	D5	
Sample 4	Time	Feature A	Feature B	Feature C	Feature D	Target
	4	A4	B4	C4	D4	D7
	5	A5	B5	C5	D5	
	6	A6	B6	C6	D6	
Sample 5	Time	Feature A	Feature B	Feature C	Feature D	Target
	5	A5	B5	C5	D5	D8
	6	A6	B6	C6	D6	
	7	A7	B7	C7	D7	
Sample 6	Time	Feature A	Feature B	Feature C	Feature D	Target
	6	A6	B6	C6	D6	D9
	7	A7	B7	C7	D7	
	8	A8	B8	C8	D8	
Sample 7	Time	Feature A	Feature B	Feature C	Feature D	Target
	7	A7	B7	C7	D7	D10
	8	A8	B8	C8	D8	
	9	A9	B9	C9	D9	
Sample 8	Time	Feature A	Feature B	Feature C	Feature D	Target
	8	A8	B8	C8	D8	D11
	9	A9	B9	C9	D9	
	10	A10	B10	C10	D10	

Step 2: Shuffle all the samples. Shuffling the data randomly allows all hidden information or patterns in the time series distributed into all the sub-sets for the following training and validation. Moreover, random shuffling can help mitigate the overfitting problem effectively.

Step 3: Apply the holdout validation and cross validation methods to split the samples into a training set, validation set and test set.

By applying such a new validation method, the time required for tuning hyperparameters and performance evaluation is shortened dramatically. Moreover, cross validation and the early stopping technique can be applied to tune hyperparameters and select proper models for the next iteration's training. An experiment has been undertaken to compare the new validation method with the walk-forward validation.

The last 10% of the collected forex data of the USD/CNY pair has been taken out as the test set; the other 90% of the data has been used to test these two validation methods respectively. The results have shown that the new validation method has achieved an accuracy of 76%, which is higher than that of the walk-forward validation method (71%), as shown inFigure 3.9.

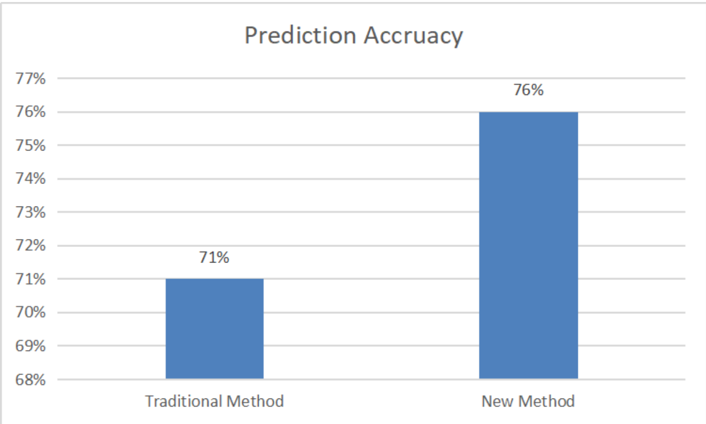


Figure 3.9 Comparison of the walk-forward validation Method with the new validation method for time series

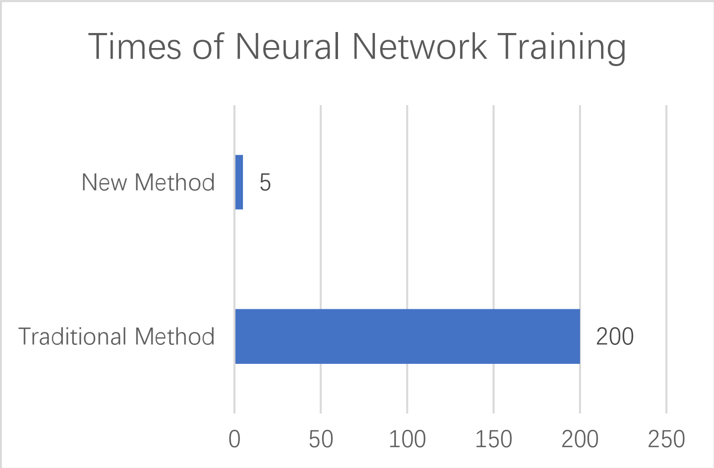


Figure 3.10 Iterations of Neural Network Training: the new validation method vs the walk-forward validation

Furthermore, the time spent on hyperparameter tuning by walk-forward validation was estimated to be 7 months, it was only 5 days by the new validation method (Figure 3.10). As we said before, walk-forward validation takes extremely lone time in its practical use. The experimental results demonstrate that the new validation method outperformed the traditional walk-forward validation method.

3.2.5 Test of the New Validation Method on a Different Dataset (USD/GBP)

To further evaluate the new validation method, a different dataset has been used to test its performance. This time, the USD/GBP forex pair has been considered. The open, high, low and close prices of this forex pair have been collected from Investing.com, and its technical indicators have been calculated accordingly. There are in total 28 features chosen for USD/CNY after feature selection, as follows:

OPENUSDGBP	Dif	-DI10	%d
HIGHUSDGBP	Dea	Adx	%R
LOWUSDGBP	macd	Tenkan-sen	Middle Bollinger Band
Change	PSAR	Kijun-sen	Upper Bollinger Band
5-day MA	relative strength5	Senkou span A	Lower Bollinger Band

10-day MA	relative strength14	Senkou Span B	Standard Deviation
+DI10	%k	TR14	USDGBP

The prediction target for USD/GBP is set to predict the price trend on the 5th-day. There are 5385 days in record for USD/GBP trading and each day has 28 features/indicators. Again, in order to compare the performances between two validation methods, ten per cent of the total samples are used as test data, i.e., the latest 538 days. In order to evaluate the performances of the traditional and new validation methods, traditional walk-forward validation method is applied to calculate the prediction accuracy and compare the performance. Worth to note that, if the new validation method is proved to be useful, it will be used to evaluate the neural network's performance as further performance evaluation. Both the traditional validation method and the new validation methods are applied to build and train the neural network, including hyperparameter tuning and model selection. The details of hyperparameter tuning and model selection will be explained later in Section 3.3.3.2. The experimental results have shown that the prediction accuracy by using the traditional validation method on USD/GBP trend prediction is 70%. In contrast, the prediction accuracy by using the new validation method is relatively higher, which is 74.1% (Figure 3.11).

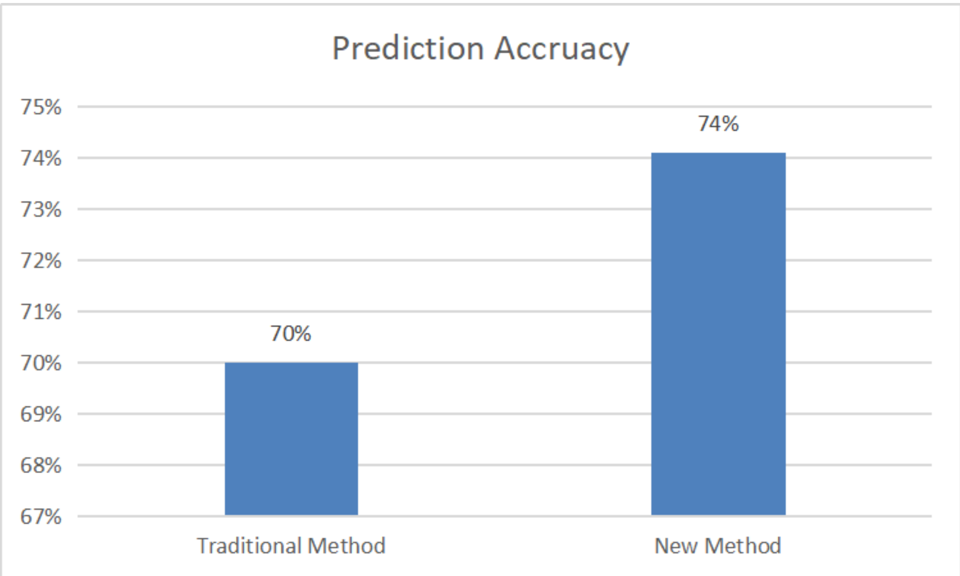


Figure 3.11 Traditional validation method vs the new validation method on USD/GBP

The new validation method has shown improved prediction accuracies for both the USD/GDP forex pair, and the USD/CNY forex pair, which is a 4% and 5% increase for each pair respectively when compared with the traditional validation method. Such improvements prove the feasibility and effectiveness of the new validation method for time series.

3.3 LSTM Training and Model Selection for Forex Prediction

3.3.1 LSTM and Learning Process

3.3.1.1 LSTM

LSTMs are used in this research to predict forex trends, because forex data is a typical time-series. A LSTM is a type of recurrent neural network (RNN) that is capable of learning the order dependence between items in a sequence; it is specially designed for time series. Compared with traditional RNNs, exploding and vanishing gradient problems can be easily solved in LSTMs (Wöllmer et al., 2010).

LSTMs are specially developed to address the problem of long-term dependencies. That is, they can remember information in past long time periods. All RNNs have repeating module of NNs. In conventional RNNs, the repeating module has a modest structure such as the tanh (hyperbolic tangent) layer (Malik & Kumar, 2018), as shown in Figure 3.12.

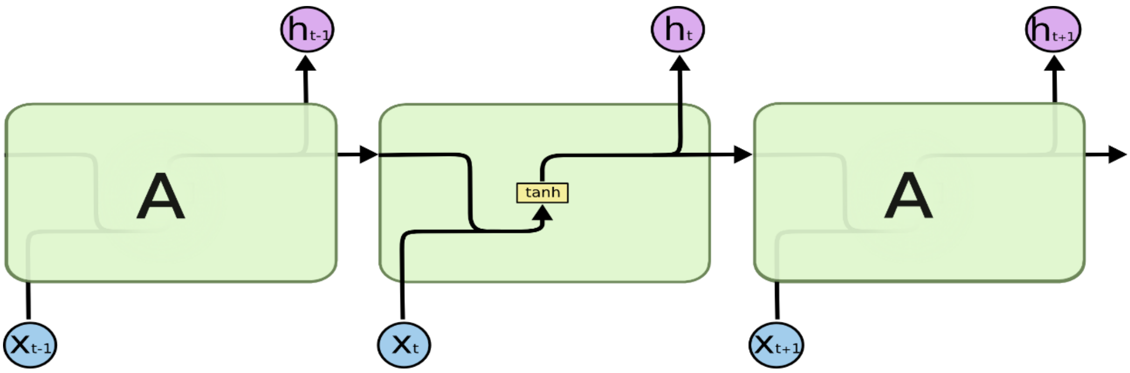


Figure 3.12 A single tanh layer as the repeating module in a standard RNN (Olah, 2015)

LSTMs have similar structures, although the repeating modules contain different structures (Graves et al., 2005; Fischer & Krauss, 2018). Instead of having one neural network layer, an LSTM module has four NN layers that interact in a unique way, as shown in Figure 3.13.

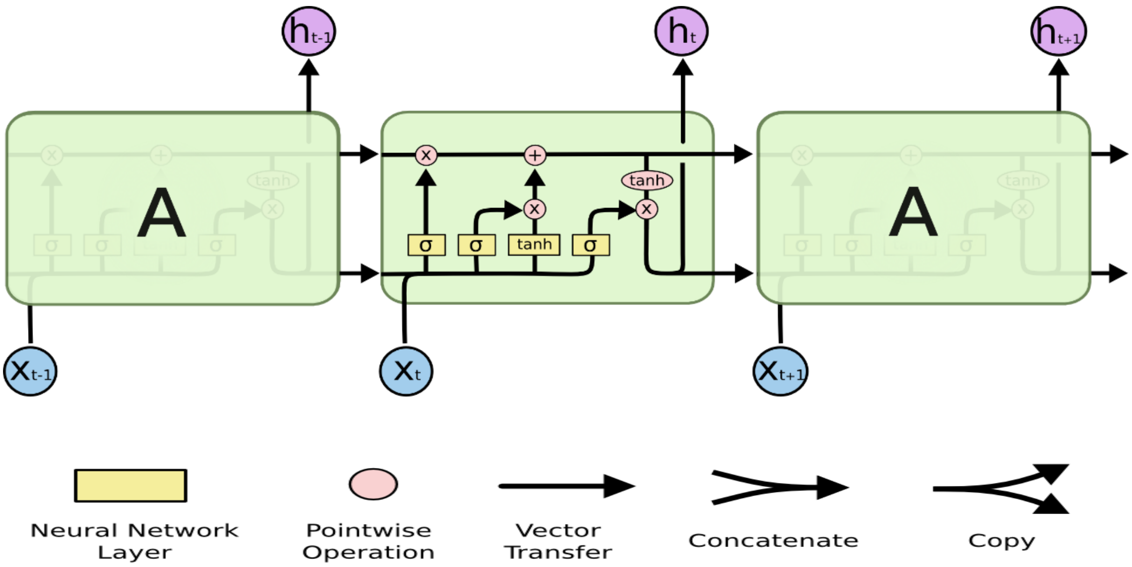


Figure 3.13 Four interacting layers in an LSTM structure (Olah, 2015)

In Figure 3.13, each line carries a vector from an input to the next output. The pointwise operations such as vector additions are shown by pink circles. The yellow boxes represent neural network layers. The merging lines indicate concatenation and the forking lines demonstrate that the content is copied and transferred to diverse locations.

The primary idea behind LSTMs is related to cell states, as shown in Figure 3.14 with a horizontal line running across the diagram. The cell state runs through the entire chain but with minor interactions and makes information more accessible to flow around unchanged. LSTMs are also able to incorporate or eliminate data to the cell state, which is closely monitored by the gates (Hochreiter & Schmidhuber, 1997; Bianchi et al. 2017).

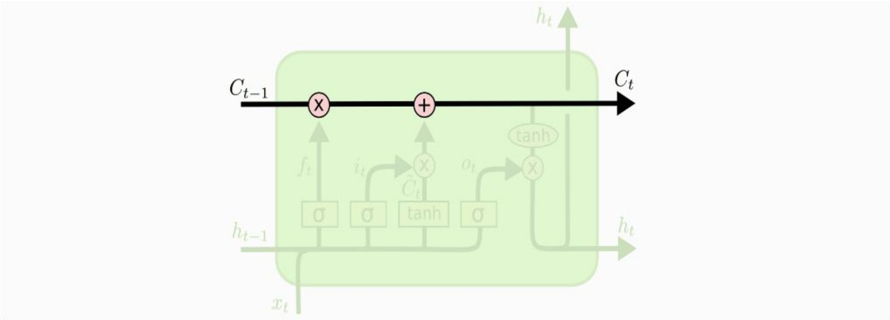


Figure 3.14 Cell state (Olah, 2015)

Figure 3.15 shows the structure of a gate, which plays a role of optionally letting information go through. The gates contain a sigmoid neural layer and a pointwise operation multiplication (Narayan & Roe, 2018; Zhang et al., 2019).

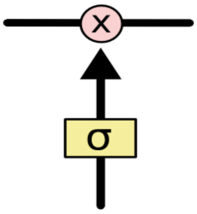


Figure 3.15 Gate structure in LSTMs (Olah, 2015)

The sigmoid layer functions give outputs between 0 and 1, demonstrating how much information is allowed to pass through. When the value is 0, nothing can be let through; when the value is 1, everything is allowed to go through. In the LSTM structure, there are three such gates that control and protect the cell states (Bianchi et al., 2017; Kim & Won, 2018).

Step 1 of the LSTM learning is to decide the specific information that will be eliminated from the cell state. This is achieved by using the sigmoid layer or the “forget gate” to checking ht-1 (previously hidden states) and xt (input) and all the numbers in the cell state Ct-1 (Gers et al., 1999). The Forget Gate equation is presented as ft = σ(Wt [ht-1, Xt] + bf) (Chung & Shin, 2018), as shown in Figure 3.16.

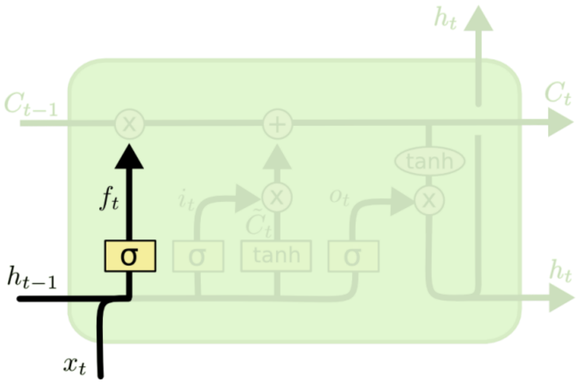
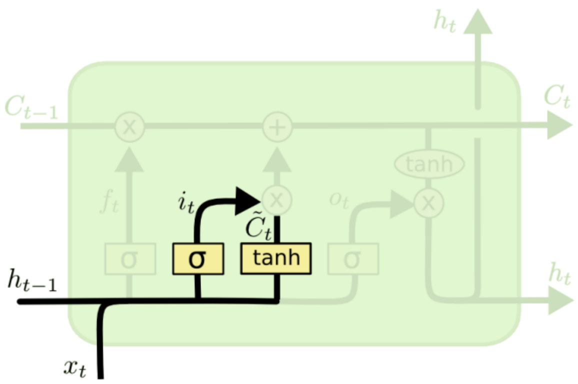


Figure 3.16 Forget Gate (Olah, 2015)

The Remember Vector is often referred to as the forget gate. The output of the forget gate tells the cell state which information to forget. It outputs a number between 0 and 1. If the output of the forget gate is 1, the information is kept in the cell state. In the input cell equation, a sigmoid function is applied to the observations or weighted inputs and the previously hidden state. The Input Gate (Figure 3.17) is presented as it = σ(Wi [ht-1, Xt] + bi), and the Input Modulation Gate equation is presented as Ct = tanh (Wc [ht-1, Xt] + bc) (Chung & Shin, 2018).



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 3.17 Input Gate (Olah, 2015)

The cell updating equation can be represented as $C_t = f_{t-1} * C_{t-1} + i_t * \tilde{c}_t$, where the old state C_{t-1} is multiplied by f_t for forgetting and then added to $i_t * \tilde{c}_t$, which represents a new candidate scaled to an expected degree, as shown in Figure 3.18.

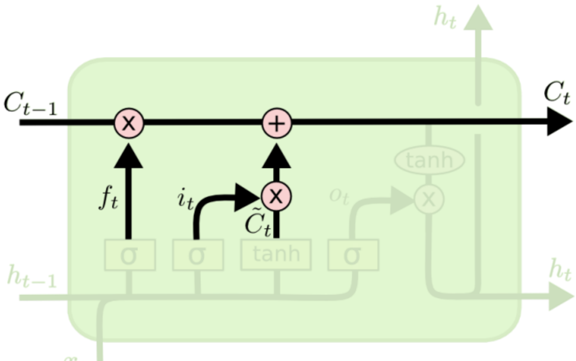


Figure 3.18 Updating the old cell state (Olah, 2015)

The Output Gate, or the Focus Vector (Figure 3.19), discharge all the possible values from the matrix to the subsequent hidden state. Its equation is represented as $O_t = \sigma (W_o [h_{t-1}, X_t] + b_o)$ (Chung & Shin, 2018).

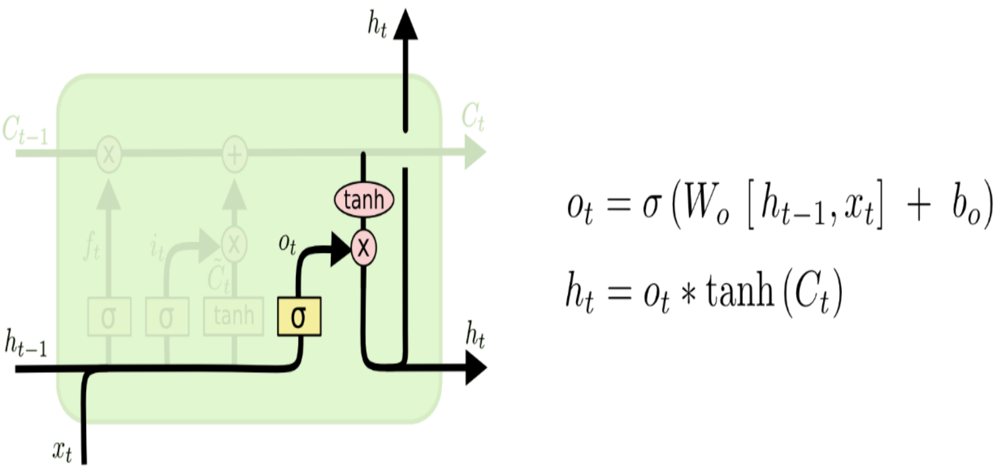


Figure 3.19 The Output Gate. (Olah, 2015)

The Hidden State, also referred to as the Working Memory, functions by accessing the information to be taken to the next sequence. The Hidden State equation is presented as $h_t = O_t \circ \tanh(c_t)$ (Chung & Shin, 2018).

In conclusion, the LSTM architecture can be summarized as the following:

1. The first sigmoid activation function in the Forget Gate evaluates the information to be forgotten based on the previous cell state (C_{t-1});

2. The second sigmoid and first hyperbolic tangent activation function represents the Output Gate and it assesses the information to be forgotten or to be saved in the cell state;

3. The third sigmoid function represents the Output Gate that pass the satisfying information to the following Hidden State.

3.3.1.2 LSTM learning process for forex forecasting

The LSTM learning process for USD/CNY trend forecasting can be expressed as follows.

1. Read samples from the data file;
2. Normalise each input feature to [0,1] by using MinMaxScaler from sci-kit- learn (Pedregosa et al., 2011);
3. Form training samples (100,39), which have the past 100 days and 39 features in each day; In total, 2761 training samples are generated from 2860 records for the USD/CNY training; The input set for training is represented as (2761,100,39), to make forex trend predictions on the fifth days, which are the outputs;
4. Shuffle all training samples;
5. Split the samples to three data sets; there are 2236 samples in the training dataset, 249 samples in the validation dataset and 276 samples in the testing dataset;
6. Train the neural network by using the training set;
7. For each epoch (maximum 1000 in total):

For each iteration (stop until all 2236 samples have been used. 2236/5 = 4.3, so five iterations in total):

i. Feed batch size (512) samples into the neural network:

ii. For each data sample with shape (100,39):

For each time step from 1 to t with shape (39):

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

where the initial values are $c_0 = 0$ and $h_0 = 0$; the operator $\circ$ denotes the Hadamard product (entrywise product). The subscript t refers to the timestep.

$x_t \in \mathbb{R}^d$: Input Vector to the LSTM unit

$f_t \in \mathbb{R}^h$: Forget Gate's activation vector

$i_t \in \mathbb{R}^h$: Input Gate's activation vector

$o_t \in \mathbb{R}^h$: Output Gate's activation vector

$h_t \in \mathbb{R}^h$: Output Vector of the LSTM unit

$c_t \in \mathbb{R}^h$: Cell State Vector

$W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters, which need to be learned during training,

where the superscripts d and h refer to the number of input features and number of hidden units, respectively.

σ_g: sigmoid function

σ_c & σ_h: hyperbolic tangent function.

For each time step from t to 1 with shape (39):

1) Calculate the gradients visa BPTT (**Backpropagation Through Time**)
2) Using gradients to update the weight via adam optimiser

iii. Calculation the loss value and accuracy value for the validation set

8. Stop training if the accuracy of the validation set did not improve for 100 epochs or the number of epochs is 1000;
9. Save the model with the best validation accuracy in the past epoch;
10. Evaluate the test accuracy with the saved model;
11. Repeat Steps 6-11 50 times to get the average accuracy.

3.3.2 Input Data Pre-Processing

The raw USD/CNY trading data collected from 12/01/2007 to 28/12/2017 needs to be pre-processed so as to be used for the LSTM training and testing.

Firstly, to make sure all the features are on a similar scale, all the raw data has been normalised by scaling to [0,1] according to the following equation:

$$scaling\ result\ (X) = \frac{X - min}{max - min},$$

where X is the current value, min is the minimum value in the sequence, and max is the maximum value in the sequence.

For LSTM, the data needs to be further reformed to a three-dimensional matrix (samples, time steps, features) (Brownlee, 2017), where

1. Samples: One sequence is one sample. One sequence means a certain time series data with the number of timesteps and number of features in each timestep
2. Time Steps: One timestep is one point of observation in the sample
3. Features: One feature is one observation at a timestep

After feature selection, the number of features is 39. However, the timesteps and the number of samples need to be explained in detail.

A timestep is the length of the period when one input sample is used to generate an output (price trend). An example of a sample with inputs and output, where timestep = 4, is listed in Table 3.16.

Table 3.16 A sample of input and output

Input for one sample shaped with (1,4,39)						Output for one sample
Date	Feature 1	Feature 2	... (35 data)	Feature 38	Feature 39	-1
25/12/2017	7.7997	7.8051	... (35 data)	7.7973	0.007696	
26/12/2017	7.796	7.7986	... (35 data)	7.7931	0.007725	
27/12/2017	6.5418	6.5598	... (35 data)	6.5568	0.017958	
28/12/2017	6.5542	6.5598	... (35 data)	6.5334	0.018775	

For the time period 12/01/2007 to 28/12/2017, which has 2861 workdays, the number of samples = 2861. The number of timesteps in each sample can be tuned in the experiments to get the best performance. An initial tentative experiment with 10-fold cross-validation has been conducted to test the performance of various timesteps ranging from 1,10, …, 100,110,120 days, respectively (Table 3.17).

Table 3.17. The results from the timesteps-tuning experiment

Cross-Validation													
Time steps	120	110	100	90	80	70	60	50	40	30	20	10	1
Accuracy in cross validation	80.72%	77.11%	81.53%	80.40%	82.33%	78.31%	79.12%	78.00%	81.60%	77.51%	76.71%	73.09%	60.64%
	78.23%	83.47%	80.65%	77.91%	79.84%	79.52%	77.02%	79.84%	77.51%	83.13%	77.51%	78.31%	61.45%
	79.44%	79.03%	79.44%	85.08%	83.47%	80.32%	76.61%	75.81%	78.23%	80.24%	78.71%	76.31%	59.44%
	82.66%	77.42%	82.26%	80.57%	78.63%	77.73%	81.05%	81.85%	75.71%	72.98%	75.90%	71.08%	59.44%
	79.84%	81.45%	83.47%	78.95%	77.02%	77.33%	81.45%	76.92%	76.92%	78.23%	79.84%	74.09%	64.11%
	78.14%	80.57%	82.19%	85.43%	78.95%	84.62%	81.78%	78.95%	78.54%	80.97%	79.84%	73.68%	60.89%
	80.57%	83.40%	81.38%	79.35%	77.33%	79.35%	83.40%	80.97%	77.73%	79.76%	77.24%	74.09%	58.94%
	82.59%	83.40%	82.19%	79.76%	84.21%	80.16%	78.95%	78.95%	78.95%	79.76%	80.08%	76.52%	57.72%

	82.19%	83.40%	82.19%	83.81%	75.71%	78.54%	78.95%	78.95%	80.57%	76.52%	73.58%	78.86%	61.38%
	82.59%	80.57%	81.78%	78.54%	78.95%	82.59%	81.38%	77.33%	78.95%	80.08%	80.49%	76.02%	60.57%
Average	80.70%	80.98%	81.70%	80.98%	79.64%	79.85%	79.97%	78.76%	78.47%	78.92%	77.99%	75.21%	60.46%

The accuracy of the prediction (listed in Table 3.17) is increasing from Timesteps 1 to 100, but it starts to decrease after 100 Timesteps and above (as shown in Figure 3.20). The decrease of the accuracy after 100 timesteps is possible because the hidden pattern of forex time-series changes over time quickly. The hidden information after 100 timesteps gives a negative effect when predicting the forex trend. According to this initial experimental result, 100 days are taken as the timestep to be used in the following LSTM training. Consequently, the initial raw data with data shape (2860,39) has been transferred to (2761,100,39) after the data pre-processing.

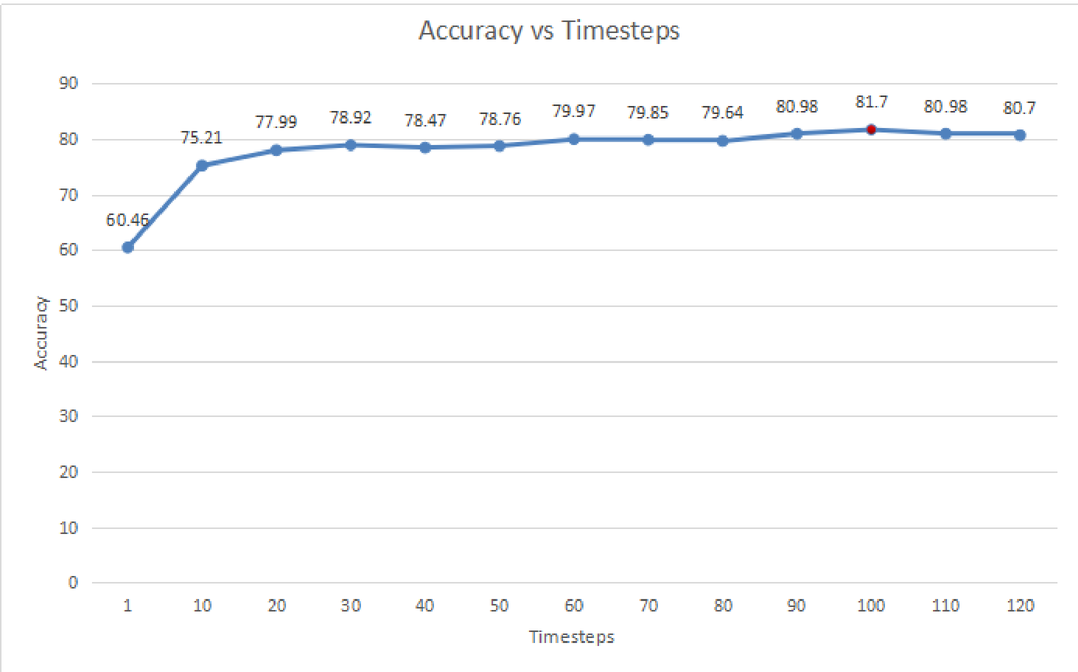


Figure 3.20 Prediction accuracy vs timesteps of LSTM

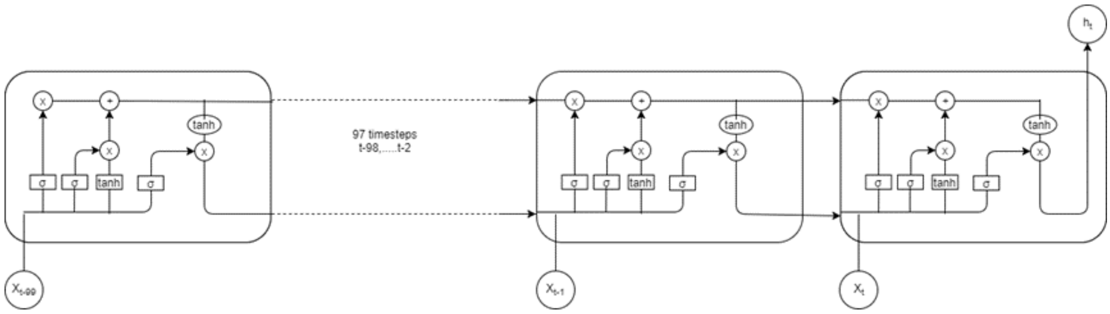


Figure 3.21 Information transferred in each timestep

Figure 3.21 shows how information is transferred in each timestep when LSTM is applied for forex prediction. X_{t-99}, X_{t-98},…… X_t stand for the forex data in the past 100 days, and h_t stands for the trend for five days.

For all 2761 samples of data, a random shuffle needs to be done. Moreover, all 2761 sample is split into three data set, i.e. training set (2236 sample), validation set (249 samples) and test set (276 samples).

3.3.3 Hyperparameter Tuning and Model Selection

3.3.3.1 Experiment Setup

All neural networks were trained using Python 3.6 and Keras (Chollet, 2015) in conjunction with TensorFlow (Abadi, 2015). The operating system and hardware specifications include Windows 10 with Nvidia 1080 Ti GPUs. The training time is boosted by GPU-based TensorFlow with a 1080ti graphics card. The GPU-based TensorFlow is proven to be around 30 times faster than a CPU-based TensorFlow in machine-learning (depends on the hardware).

3.3.3.2 Hyperparameter tuning

For any neural networks, there exists a number of hyperparameters that need to be tuned before formal training. In machine learning, a hyperparameter is different from a parameter. A parameter is tuned during the training process, e.g., Backpropagation; it uses to be a weight or bias in a neural network. A hyperparameter needs to be tuned before training and has different values in different settings, depending on the training data, algorithms and even the software and hardware used. Good hyperparameters can help an algorithm perform better.

In general, the hyperparameters that need to be tuned in an LSTM neural network include the number of timesteps, the number of hidden layers, the number of neurons in each hidden layer, the activation function for each layer, the number of training epochs, batch size, learning rate, optimiser and loss function. In this research, as explained in Section 3.3.2, the timesteps have been tested to be 100; the activation function of a standard LSTM is by default to be hyperbolic tangent (tanh) (Olah, 2015); the optimizer is 'adam', which is reported to be one of the best optimizers and the loss function needs to be 'categorical_crossentropy' because it is a three-way classification problem. Finally, for the training epochs, a technique, called 'early stopping', with the purpose of avoiding the over-fitting problem (Prechelt, 1998) has been applied instead of a fixed number of epochs.

Most of the neural network training process faces an over-fitting problem. That is, during the training process, the accuracy of the training set is always increasing, but the accuracy on the real test set is increasing at the beginning but starts to decrease at some point. This is because the trained ANN model is over-fitting with the training data and hence lost the ability to adapt to new data. For this reason, a validation set is often used to help choose a more generalised model during the training process.

When the accuracy of the validation set is increased to the maximum point during the training process, the model with the highest accuracy on validation set is saved as the final trained model (shown in Figure 3.22).

In this research, the maximum training epoch is set to be 1000 to ensure the training can reach the turning point of overfitting. An epoch toleration of 100 has been set, which means a training process will be stopped to save time if the accuracy on the validation set is not improving and starts to decrease in the past 100 epochs. Consequently, the trained model with the highest accuracy will be saved as the training result (Prechelt, 1998).

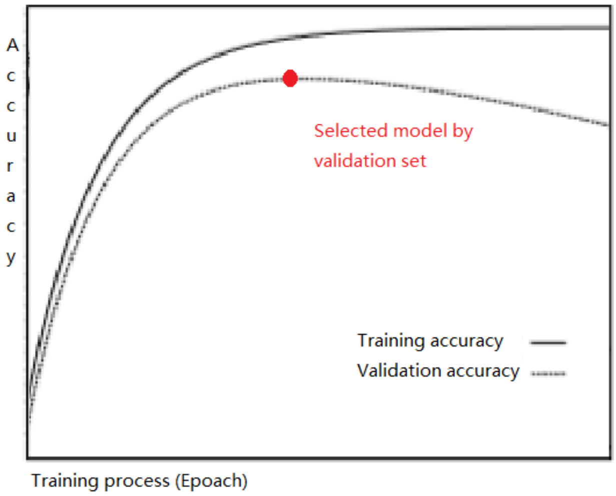


Figure 3.22 Early stop at the red point to avoid over-fitting

K-fold cross validation (Brownlee, 2018) is used in this research and in order to ensure a high accuracy, K = 10 is chosen for the cross-validation. 10% of the total 2761 samples after data reforming are used as the test set to evaluate the performance of the trained LSTM. 10-fold cross-validation is applied to the rest of the samples for hyperparameter tuning(Figure 3.23).

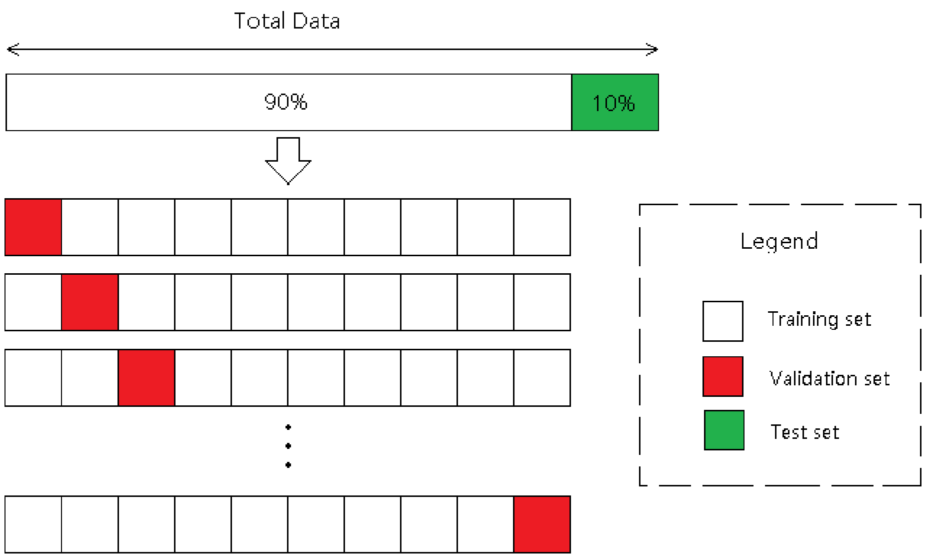


Figure 3.23 10-fold cross validation

Usually the methods used to tune the hyperparameters are classified into two types: manual selection and automatic selection. In this research, to ensure the best settings of the hyperparameters, a large amount of time has been spent on tuning hyperparameter (including timesteps) manually.

The initial values for the hyperparameters are set by experiences: 100 neurons are set in one hidden LSTM layer, the mini-batch size is 512, and the default learning rate is used. The hyperparameters are then tuned one-by-one: different values of a hyperparameter are tested while the other hyperparameters remain the same.

- Number of hidden layers

Table 3.18. The result from number of layers tuning experiment

Cross-validation			
Number of layers	1 layer	2 layers	3 layers
	81.53%	79.52%	76.71%
	80.65%	80.24%	79.84%

Accuracy in cross validation	79.44%	79.84%	75.40%
	82.26%	84.68%	78.63%
	83.47%	81.05%	78.23%
	82.19%	81.38%	78.95%
	81.38%	80.97%	74.09%
	82.19%	83.40%	75.71%
	82.19%	80.57%	82.19%
	81.78%	84.21%	80.16%
Average	81.70%	81.59%	77.99%

- Number of neurons

Table 3.19 Result from number of neurons tuning experiment

Cross-validation						
Number of Neurons in the hidden layer	10	50	100	150	200	250
Accuracy in cross validation	64.94%	81.67%	83.27%	83.27%	82.87%	82.07%
	65.06%	82.33%	81.53%	79.92%	81.93%	83.53%
	63.05%	78.71%	77.11%	75.90%	78.71%	75.90%
	74.60%	83.47%	84.27%	83.87%	82.26%	82.66%
	67.34%	77.42%	81.45%	81.45%	79.44%	82.66%
	66.53%	79.03%	80.65%	82.26%	80.24%	80.24%
	65.32%	76.21%	78.63%	78.63%	77.82%	79.03%
	64.52%	81.45%	84.68%	84.27%	84.68%	83.47%
	68.15%	81.45%	82.66%	81.05%	81.85%	80.24%
	74.19%	81.85%	84.27%	81.85%	82.66%	81.45%

Average	67.37%	80.36%	81.85%	81.25%	81.25%	81.13%

- Batch size

Table 3.20 Result from batch size tuning experiment

Cross-validation					
batch size	32	64	128	256	512
Accuracy	82.07%	82.47%	82.87%	83.27%	83.27%
	78.31%	78.71%	80.72%	82.73%	81.53%
	76.31%	77.51%	76.71%	78.31%	77.11%
	79.44%	81.85%	82.66%	83.47%	84.27%
	79.84%	81.85%	80.24%	79.84%	81.45%
	81.05%	80.65%	81.05%	81.85%	80.65%
	77.02%	80.24%	80.24%	77.82%	78.63%
	79.03%	81.85%	85.08%	85.48%	84.68%
	80.65%	79.03%	81.45%	79.84%	82.66%
	81.85%	81.85%	81.85%	82.26%	84.27%
Average	79.56%	80.60%	81.29%	81.49%	81.85%

- Learning rate: learning rate suggested being set like a decay sequence, decrease with the epochs

Table 3.21. The result from the learning rate tuning experiment

Cross-validation		
	Decay Learning Rate	Fixed Learning Rate (default)
Accuracy	81.67%	83.27%
	81.53%	81.53%
	77.51%	77.11%
	83.87%	84.27%

	80.65%	81.45%
	82.26%	80.65%
	78.23%	78.63%
	83.06%	84.68%
	81.45%	82.66%
	81.45%	84.27%
Average	81.17%	81.85%

The best hyperparameters are selected with the highest average accuracy in cross-validation: the network consists of 100 neurons in one hidden layer with the tanh activation function, and three neurons as the outputs stand for three different trends; the batch size is set to 512.

3.3.4 Neural Network Structure

In conclusion, the neural network used for forex prediction is composed of three layers (Figure 3.24): an input layer with 39 inputs, a hidden layer with 100 LSTM neurons and an output layer with three outputs (increase, decrease and no-change). The standard hyperbolic tangent activation function is used in the 100 LSTM neurons, and the batch size set to 512. The default 0.01 learning rate of the ‘adam’ optimiser has a better performance than the decay learning rate by epochs. The early stopping technique has been applied to avoid overfitting and save training time. One thousand maximum epochs and 100 epochs toleration have been set for early stopping.

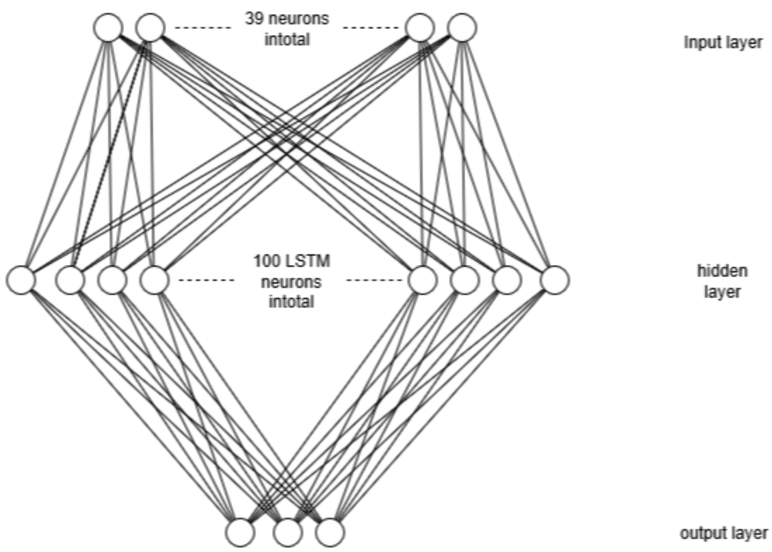


Figure 3.24 The network structure of the LSTM used for forex prediction

3.3.5 Performance Evaluation

Figure 3.25 shows the accuracies and loss values on the training and validation datasets during the LSTM training process. Such a process has been repeated 50 times and the average results are shown below.

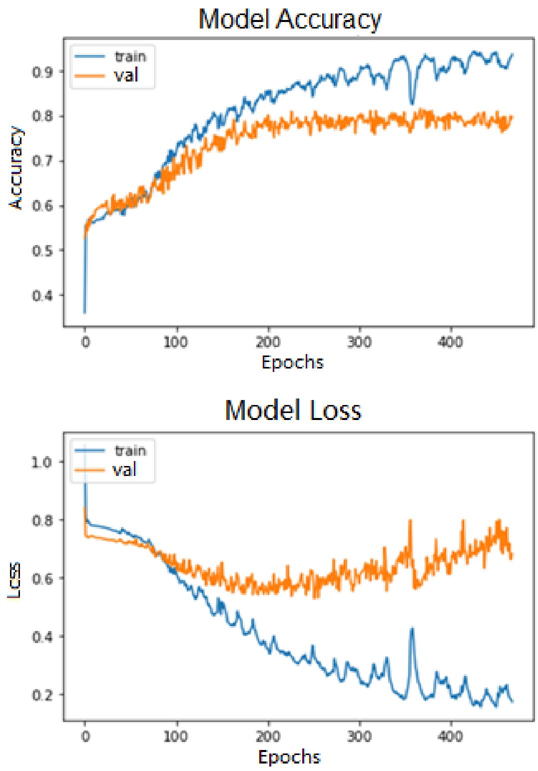


Figure 3.25 Accuracy & loss on the training and validation datasets in LSTM learning

The average of the prediction accuracy on the test data is 78% by using the standard LSTM. This 78% accuracy is evaluated by the new validation method rather than the traditional walk-forward validation method. The new validation method can more comprehensively reflect the prediction accuracy of the whole period time series.

Worth to note that, predicting the trend of forex is a classification problem. Particularly it is a multivariate time-series classification problem because there are multiple features used for predictions. For time series classifications, the two most widely used algorithms are artificial neural networks (ANNs) and support vector machines (SVMs) (Alweshah et al., 2017). With regards to artificial neural networks, deep learning neural networks such as Multi-layer perceptrons, RNNs (recurrent neural networks) and Convolutional neural networks are typical example, each of which has their own advantages and functionalities for time series forecasting.

Multilayer Perceptrons, or MLPs for short, are a classical type of neural networks. They are comprised of one or more layers of neurons. Data is fed to the input layer and then transferred through one or more hidden layers. The final predictions are made on the output layer, which is also called the visible layer. MLPs are suitable for classification problems, where the outputs are assigned a class or label. They are also suitable for regression problems, where a real-value is predicted given a set of inputs. Multi-layer perceptrons can handle missing values, model complex relationships (such as non-linear relationships) and support multiple inputs. But Multi-layer perceptrons (MLP) have a disadvantage of having to provide a fixed number of inputs for producing a fixed number of outputs, that is, to specify the temporal dependences between inputs and outputs before the model design. The fixed mapping from the inputs to the outputs in these feedforward neural networks pose a problem when a sequence of inputs is provided to the model.

Convolutional Neural Networks, or CNNs, were designed to map image data to an output variable. They have proven so effective that they are now almost a must method for any type of prediction problem involving image data (Brownlee, 2018). A benefit of using CNNs is to develop an internal representation of a two-dimensional image. This allows a CNN model to learn the positions and scales in the variant structures of the data, which is important when working with images (Brownlee, 2018). Another benefit of convolutional neural networks (CNNs) is automatic feature extraction from the raw inputs. This helps CNNs be capable of processing multi-variate inputs and complex non-linear relationships and be robust to noises (missing values). Although CNNs are powerful for images data, their applications to non-image data need to be further exploited (Brownlee, 2018).

Recurrent Neural Networks (RNNs), were designed to cope with sequence prediction problems. Specifically, LSTMs, as a type of RNNS, have the following features:

allowing multi-variate input, robustness to noises, allowing multi-variate output, automatic feature extraction, and complex relationship modelling. LSTMs are perhaps the most successful RNNs because they overcome the gradient explosion and gradient vanishing problems in recurrent neural network training and hence have been used for a wide range of applications. LSTMs can also process sequential input data, particularly for multi-variate time series classification problems, outperforming MLPs and CNNs (Brownlee, 2018). For this reason, CNNs and MLPs are not chosen as benchmark algorithms for forex trend predictions.

Instead, standard RNNs and SVMs are selected for benchmark testing, when applied to the same USD/CNY trading data for a comparison with the new validation method based LSTMs proposed in this thesis.

Hyperparameters are fine tuned for the standard RNNs too (details in Section 4.3.1). The tuned RNNs have 60 time steps instead and the resulting average accuracy over 50 runs is 71.6%.

When support vector machines (SVM) (Huang & Wu, 2008; Kazem et al., 2013) are trained for forex predictions, the SVM function from the sci-kit-learn machine-learning library (Pedregosa et al., 2011) is used for implementation. Below is the code:

```
clf=svm.SVC(kernel='rbf')
clf.fit(train_X, train_Y)
result = []
result.append(np.mean(test_Y == clf.predict(test_X)))
print(result)
```

The prediction result is only 61.23% by using SVMs. This might be because SVMs is not suitable for non-linear problems, especially the financial time series data.

The comparison results show that, even though LSTMs have higher computational complexity, they have achieved much better predictions, with an average prediction accuracy 78%, which is higher than those achieved by the benchmark algorithms (Figure 3.26).

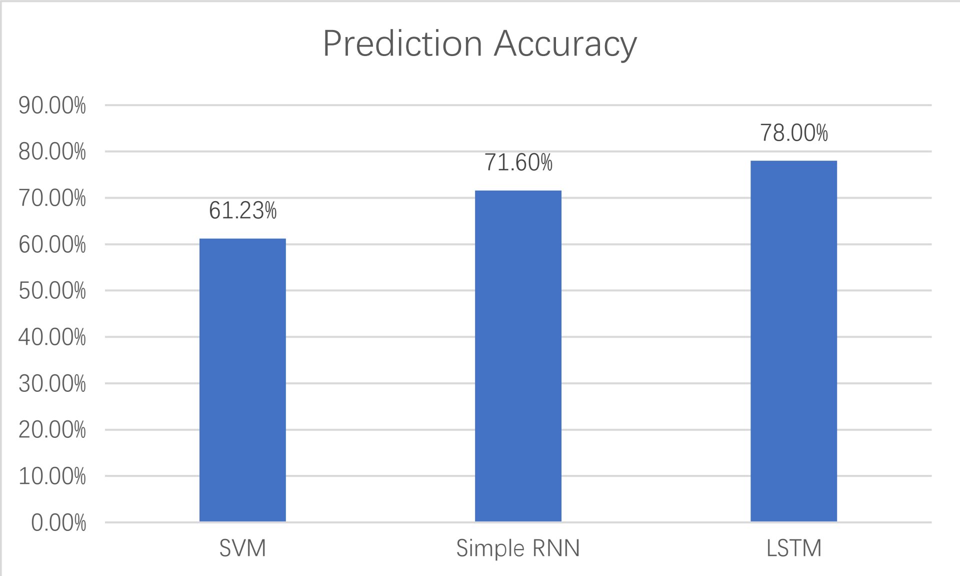


Figure 3.26 Prediction Accuracies of SVMs, RNNs and LSTMs

The high prediction accuracy of 78% is good enough for profitable automatic trading. This point-of-view has been confirmed in (Kuperin et al., 2001, as cited in Kondratenko & Kuperin, 2003), where a profitable trade strategy was built for GBP trading with a similar prediction accuracy and a longer prediction horizon actually.

Chapter Summary

This chapter has firstly provided a comprehensive analysis of the indicators involved in fundamental and technical analysis for forex forecasting, by using the USD/CNY forex pair as the test forex pair. There are 75 indicators in total, including 46 from fundamental analysis and 29 from technical analysis, respectively. 39 influential indicators have then been identified through feature selection, which have more impact on forex predictions. The prediction target has also been set as the trend predictions for the next fifth-day after experimental analysis.

Secondly, a new validation method for time series training is proposed in this chapter after reviewing the current validation methods for non time- series and time-series data. The problems and limitations of the current validation methods, e.g., walk-forward validation, for time series data are presented accordingly. Experimental results have proved that the new validation method for time series data performed better than the walk-forward validation method.

Thirdly, the detailed process of how to build LSTMs with the new validation method for forex forecasting is explained. Hyperparameter tuning for the LSTM learning is also introduced.

Finally, standard RNNs and SVMs are implemented as benchmark methods to test the performance of LSTMs when they are used to predict the USD/CNY price trends. The experimental results have shown that LSTMs have outperformed both RNNs and SVMs.

Chapter 4 Hybrid Activation Functions based LSTMs

This chapter proposes a new algorithm for LSTM training, which is based on Hybrid Activation Functions (HAFs), inspired by biological neural networks. The first section of this chapter introduces the basic role of activation functions in artificial neural networks and explains the inspirations of HAFs by comparing information processing in both artificial neural networks and biological neural networks. In the next section, a novel type of LSTM neural network using HAFs is designed and explained in detail, and plenty of experiments have been done to prove its performance. Furthermore, the HAF has also been successfully applied to standard recurrent neural networks, and their performances have been evaluated. Finally, a conclusion is stated for this chapter.

4.1 Biological Inspirations

4.1.1 The Role of the Activation Function

The activation function is essential for ANNs to learning from complex data. This is because the associations between the input and output variables are usually non-linear. Activation functions work by introducing non-linear properties to an ANN network with a specific purpose of converting input signals to proper output signals (Mishra et al., 2017; López-Rubio et al., 2019). Precisely, in an ANN, an activation function calculates the sum of the weights (W) and their corresponding inputs (X) by following the application of a function f((WX) (Sodhi & Chandra, 2014; Marreiros et al., 2008).

When the activation function is absent in a neural network, the output signal will be degraded to a simple linear function of the inputs. A linear function also refers to a simple polynomial of a single degree. Although a linear equation is easy to solve, it lacks the capability to learn complex associations were often happening in real-world problems. A neural network that has no activation functions can only imply a linear regression model that has limited power and is unable to perform effectively in most forecast analysis. The activation functions in ANNs, therefore, play an important role in helping learn the complex, non-linear, and highly dimensional information from big data. Particularly those ANNs with multiple hidden layers are especially suitable for analysing intricate data sets (Ma & Khorasani, 2003; Marreiros et al., 2008).

Therefore, the performance of an activation function has a substantial impact on an artificial neural network's operation. The common activation functions that have been used to enhance the learning performance (and especially the LSTM) include the hyperbolic tangent and sigmoid (log-sigmoid) functions (Chen & Chang, 1996). Some other activation functions have also been reported to be successful such as the log-log functions, complementary log, probit functions (Gomes & Ludermir, 2008), non-polynomial functions and Hermite polynomials (Ma & Khorasani, 2001, Rigatos & Tzafestas, 2006).

4.1.2 Artificial Neural Networks vs Biological Neural Networks

It is well known that the idea behind Artificial Neural Networks (ANNs) is based on the perspective that biological neurons and their connections are the fundamental basis for a biological brain to function, such as the human brain (Ridzuan et al., 2019). During brain processing, neurons, which are the nerve cells (Figure 4.1), remain attached to axon cells (Swarnkar, 2018). Stimuli from the external environment are received by dendrites, and it is the stimuli that provoke electric impulses which quickly travel through neural networks (Swarnkar, 2018). As a result, one neuron can relay information to another neuron to implement information transfer. However, it may also fail to forward a message on some occasions, i.e., when the stimuli from the external environment are too weak (Swarnkar, 2018).

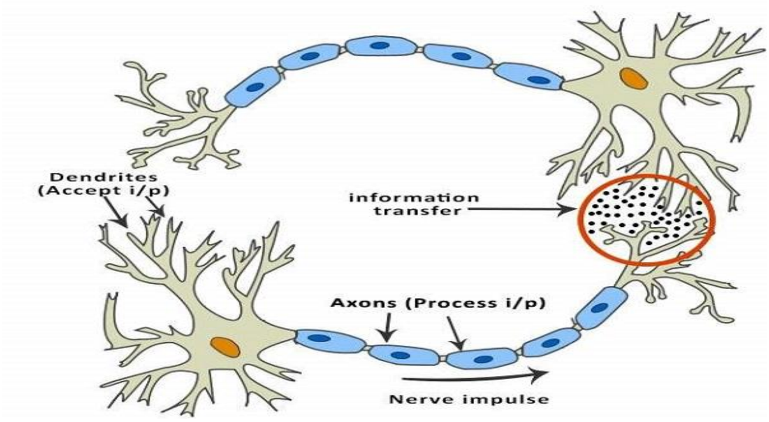


Figure 4.1 A neuron cell (Swarnkar, 2018)

An ANN contains multiple nodes that imitate biological neurons of the brain to reflect their generalisation and learning capabilities (Swarnkar, 2018). Artificial neurons have connections to and interact with one another. The nodes (neurons) take input information and perform some basic operations on the received data. Subsequently, the results obtained from the operations are passed on to other neurons, in which case, the results or output of a node is called the node value or activation (Swarnkar, 2018). Every link (connection) in an ANN is associated with a weight, so the ANN learns by altering the values of the weights. Figure 4.2 illustrates a simple ANN.

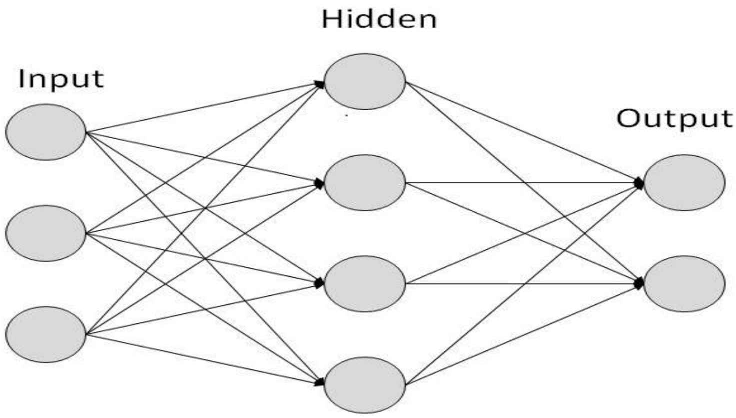


Figure 4.2 An illustration of an ANN

In Figure 4.2, each arrow shows a connection between two neurons and the pathway for information flow. Every connection has a weight that regulates the signal between two neurons at each end of the connection. If the generated results are desired or proper, then there is no need to adjust the weights (Swarnkar, 2018). However, if the output is undesired or poor, then the system has to alter the weights so as to improve the subsequent outcomes in order to solve a problem.

4.1.3 ‘Activation Functions’ In Biological Neural Networks

In a biological neural network, the way neurons process signals are similar to the activation functions in an ANN. Generally, the physiological signals are received in the dendrites of the neuron and transmitted via axon to the axon tips (axon terminals) (Figure 4.3). A synapse is formed when an axon terminal reaches a dendrite of another neuron. The electric nerve impulse is transmitted by a synapse between neurons (Hollander, 2018).

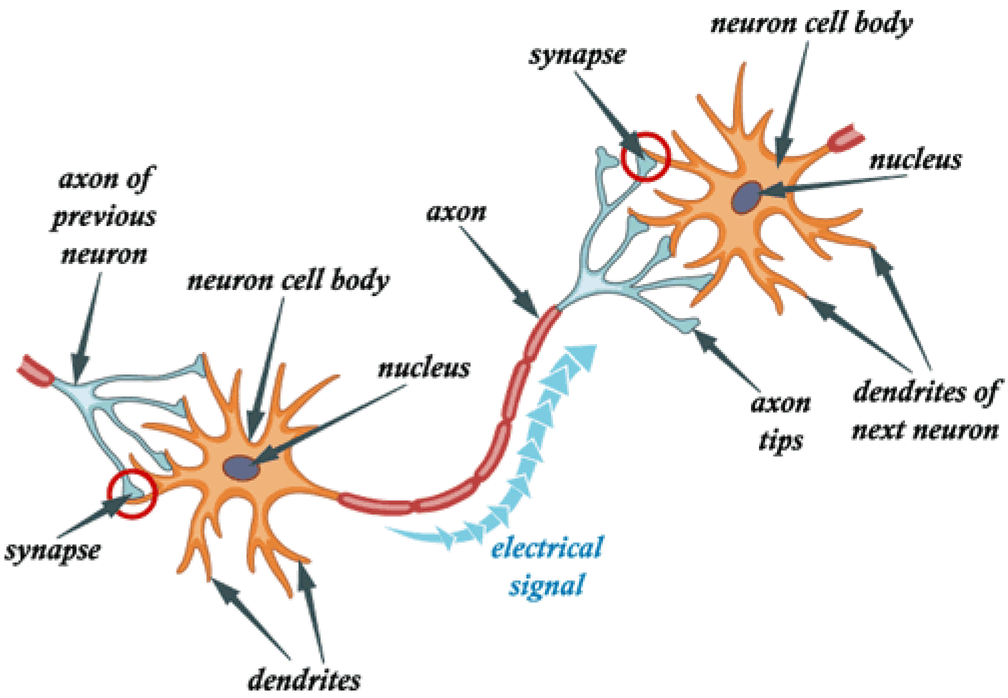


Figure 4.3 How electrical signal transfer between neurons.

Figure 4.4 How neurons process signals received by the synapses.

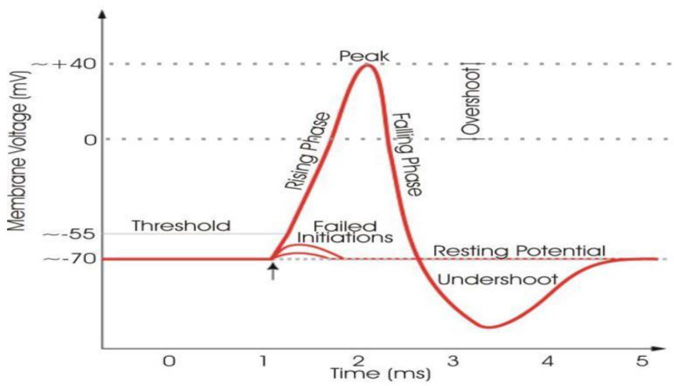


*Figure 4.4 An illustration of a neuron response to stimuli (*Hollander, 2018*)*

As shown in Figure 4.4, there exists a resting potential at -70 mV (millivolts) for a neuron, called membrane potential. At this resting state, the cell does not have any signal. Therefore, in order to transmit a signal, the cell needs to be stimulated by the synapses. The electric nerve impulse transmitted by synapses can be used to increase the voltage. In the illustration, the required threshold voltage to trigger response is -55 mV. This implies that the total sum of inputs must amount to +15 mV (Hollander, 2018).

So, if every input neuron contributes about +5 mV, the initial activation will require three input neurons to initiate a neuron response at the same time (Hollander, 2018). The activation is similar to the ANN in that it calculates the sum of the weighted inputs, and the activation function decides how this value can be transmitted to the next neuron. As a result, the way for a neuron to process signals is similar to the activation functions in the ANN (Hollander, 2018).

4.1.4 Inspirations of Hybrid Activation Functions

The motivation for using hybrid activation functions in LSTMs comes from the biological neural network systems. Usually, in an artificial neural network, there is only one type of activation function in the same layer for all neurons. This is in contrast to biological neurons, wherever different threshold voltages are employed for different neurons to fulfil a task, and the threshold voltage is between -50 and -55 mV (Seifter et al., 2005). Inspired by this, this study takes the hypothesis that, it is possible to improve the decision processes if different activation functions are used for the neurons in the same layer of an artificial neural network, instead of a uniform activation function for all neurons as used in the same layer of traditional artificial neural networks. When applied to LSTMs, the approach entails using different neurons with different activation functions, even in the same layer. The assumption, in this case, is that this approach will help improve the prediction accuracy of the LSTM neural network.

4.2 Hybrid Activation Functions based LSTMs

4.2.1 Activation Functions

There are three widely used non-linear activation functions. They are sigmoid, hyperbolic tangent and Rectified Linear Unit (ReLu) (Walia, 2017; Sharma, 2017; Sharma, 2017).

4.2.1.1 Sigmoid Activation Function

The equation for sigmoid is:

$$\mathrm{y} = \frac{1}{1 + e^{-x}} \qquad (4.1)$$

The reason why a sigmoid function can be widely used as the activation function in an artificial neural network is that sigmoid is a smooth, nonlinear and non-binary function (Sharma, 2017). According to the definition of sigmoid, the range of it is from 0 to 1 (Elliot, 1993) (shown in Figure 4.5). It is obvious that the value changes quickly between x= -2 and x=2, and it helps to make a clear distinction. As a result, it is mainly used to calculate for the probability problems. The softmax function as a transformation of the sigmoid function is specifically used for multiclass classification problem (Sharma, 2017). However, the sigmoid function suffers from the vanishing gradient problem (Sharma, 2017).

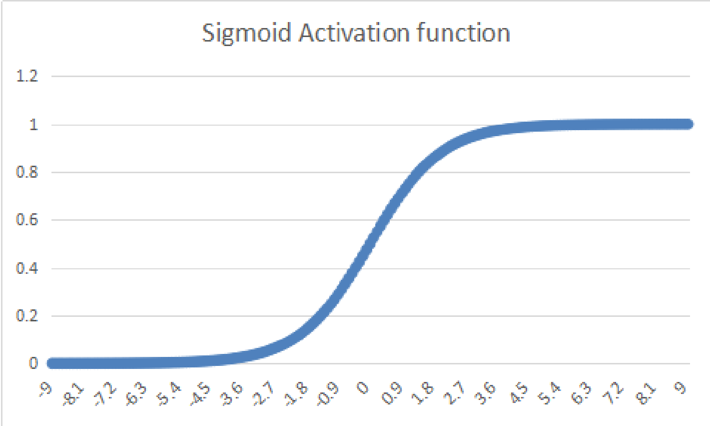


Figure 4.5 Sigmoid activation function in LSTM

4.2.1.2 The Hyperbolic Tangent Activation Function

The hyperbolic tangent function (Eq. 4.2) is a scaled sigmoid function, so it is also shaped as ‘s’. It has a stronger gradient than sigmoid. Hyperbolic tangent can be calculated by Eq. 4.2, and its range is from -1 to 1 (shown in Figure 4.6). As a result of its range, the hyperbolic tangent function is normally used for two-classes classification problems (Sharma, 2017). Same as sigmoid, the hyperbolic tangent also suffers from the vanishing gradient problem (Sharma, 2017).

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.2}$$

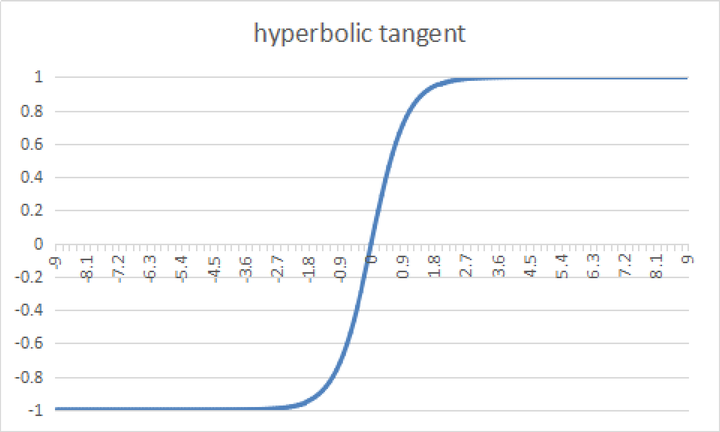


Figure 4.6 Hyperbolic tangent

4.2.1.3 Rectified Linear Unit (ReLu) Activation Function

Due to the vanishing gradient problem in the sigmoid and hyperbolic tangent function, ReLu function has been most widely used right now (Sharma, 2017). It is particularly designed to overcome the vanishing gradient problem (Walia, 2017). It has been proved that the ReLu function has six times improvement in convergence than the hyperbolic tangent function. The equation of ReLu function is (Eq. 4.3):

$$\mathrm{ReLu(x) = max\ (0, x)} \tag{4.3}$$

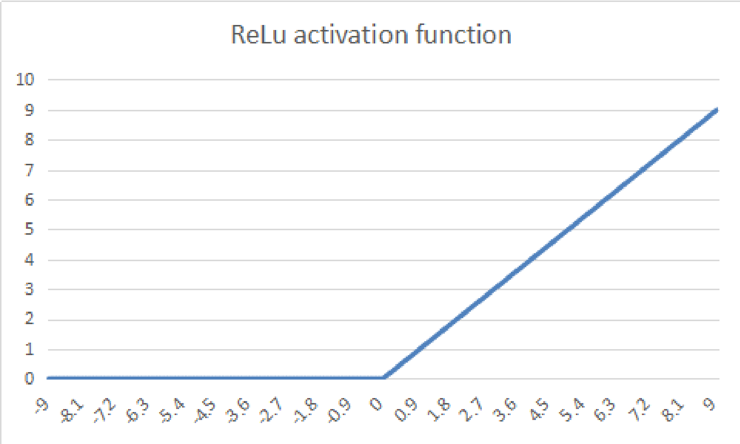


Figure 4.7 ReLu activation function

When x <0, the value is 0 and when x>0, the value is x. It is quite simple but has been proved to be effective (Walia, 2017).

4.2.2 Feasibility Analysis of Using Hybrid Activation Functions in LSTM Neural Networks

4.2.2.1 Activation Functions in LSTMs

In a standard LSTM neuron, there are 5 activation functions. Three sigmoid activation functions are used for the input, forget, and output gates and two tanh (hyperbolic tangent) activation functions are used for the cell state and hidden state respectively (Figure 4.8). The concepts of gates and states have been introduced in Section 3.3.1.

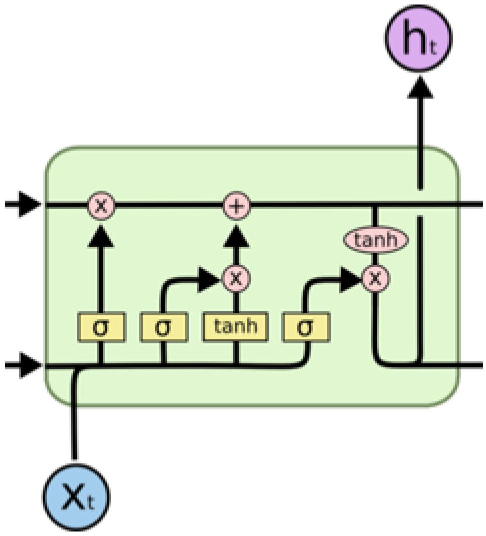


Figure 4.8 5 Activation functions in a LSTM neuron (Olah, 2015)

This is in contrast to the standard RNN which has only one hyperbolic tangent activation function in a RNN neuron. The three sigmoid activation functions playing as a 'Gate' in LSTMs range between [0,1]. When information goes through a 'Gate', all of the information gets through when the 'Gate' is fully open, and none is past through when the 'Gate' is closed. In LSTMs, '0' means 'Gate' closed and '1' means 'Gate' fully open, which is very similar to a sigmoid function. This is why sigmoid is one of the best activation function to use for LSTM gates and this is not going to change in this research.

The other two hyperbolic tangent activation functions for the cell state and hidden state are generally used for calculation and therefore can be substituted by some other functions, such as Keras. Without further explanation, the activation function in this thesis stands for the activation function for the cell state and hidden state. By default, the activation function for the cell state and hidden state in a LSTM neuron is set as a hyperbolic tangent function. For substitutions, sigmoid is actually more suitable than ReLu, Because ReLu is an unbounded activation function which ranges from 0 to ∞ (shown in Figure 4.7). If an unbounded activation function is used in LSTM neurons, the LSTM model will become divergent. For example, if a value passing through the sigmoid gates is between 0 to 1, it will quickly explode when it Hadamard products with an unbounded value. This is further proved by experiments using Relu as the activation function in LSTMs (Figure 4.9). In Figure 4.9, the blue line stands for the forex prediction accuracy on the training data set and the red line stands for the accuracy on the validation data set. It is clear that, the accuracy is not improving during the 500 epoch training process, indicating the network is not learning.

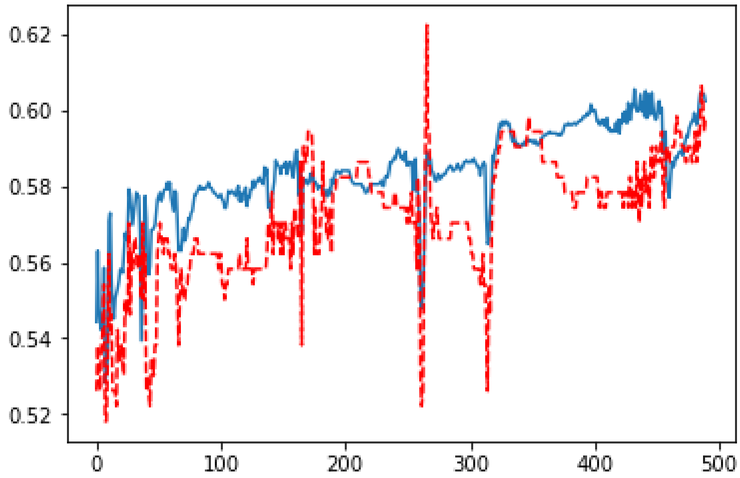
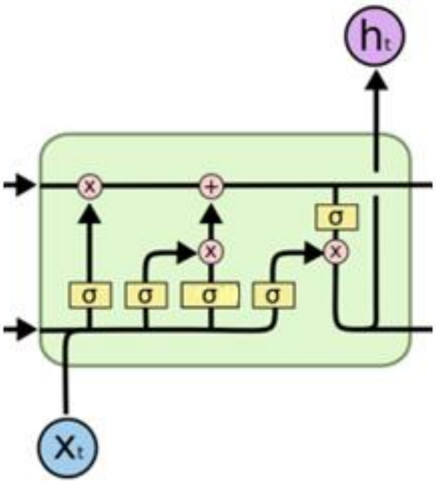


Figure 4.9 Training and validation accuracies using the ReLu activation function

Although the range of sigmoid is half of the range of the hyperbolic tangent function, the sigmoid function can be used as the activation function of LSTM neurons, as illustrated in Figure 4.10.



.

Figure 4.10 LSTM neurons with the sigmoid activation function

4.2.2.2 Hybrid Activation Functions in LSTM Neural Networks

There are three main factors of a neural network, which can be improved to get a better neural network performance, including the neural network architecture and the topology of connections between neurons, the learning algorithm and the activation function (Greff et al., 2016). Most of the research on neural networks have been focused on algorithm improvement, but the improvement of activation functions has not been fully exploited yet (Greff et al., 2016). Using different activation functions in a LSTM neural network can actually lead to different performances (Greff et al., 2016). The activation function used in a LSTM neuron should satisfy the following requirements: continuous and bounded. In addition, it needs to be sigmoidal, that is, the limits for infinity should satisfy the following equations (Greff et al., 2016):

$$\lim_{x \to -\infty} f(x) = \alpha$$

$$\lim_{x \to +\infty} f(x) = \beta$$

$$\text{with } \alpha < \beta$$

Greff et al. pointed out that the two most widely used activation functions in LSTM neurons are sigmoid and hyperbolic tangent. Both sigmoid and hyperbolic tangent activation functions meet the requirements introduced above. However, the ranges and the derivative values of these two activation functions at each point are different. These differences will affect the learning capability of a neural network and hence its learning performance.

This sub-section analyses the features of these two activation functions and their use in LSTM neurons from two aspects. One is the range of values and the other is the derivative value at each point.

As Section 4.2.2.1 explained, it is normally the activation function in the cell state and hidden state that can be changed. So the range of the activation function stands for the value which will be combined with the 'information' coming through a Gate. For the sigmoid activation function, the range is (0,1). If the range is from 0 to 1, this means the negative value will not be added to the 'information'. For the hyperbolic tangent activation function, the range is (-1, + 1), which means any positive or negative value will be added to the 'information'. Under this situation, by using the sigmoid activation function, a neuron can be used more efficiently to pass a positive value; and by using the hyperbolic activation function, a neuron can be used to pass both positive or negative values.

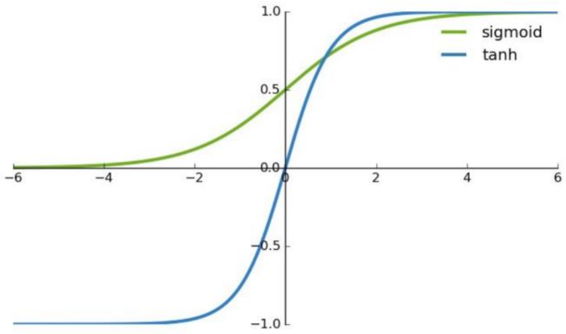


Figure 4.11 Sigmoid vs Tanh activation functions

However, for the sigmoid activation function, as Figure 4.11 shows, the derivative values of the sigmoid activation function changes very slowly. When slow learning happens, the optimization algorithm that minimizes an objective may get stuck in local minimums and won't be able to obtain the maximum performance from an artificial neural network model. Compared with the sigmoid activation function, the advantage of the hyperbolic tangent activation function is that its derivatives are steeper, which means it can reach more values. This suggests that the hyperbolic tangent activation function will be more efficient because it has a wider range of possible solutions for implementing faster learning.

Worth to notice that, the slow learning of the sigmoid activation function may be beneficial than fast learning in some cases. Slow learning may result in a long training process that easily gets stuck. However, fast learning may result in learning a sub-optimal set of weights too fast or an unstable training process (Brownlee, 2019). The effectiveness of an activation function depends on not only its own properties, but also the learning algorithm and training data. The neural network might need more than one kind of activation function to perform different kinds of learning.

Sigmoid activation function and hyperbolic tangent activation function have different value ranges and derivative value changes, which are suitable for different cases respectively. Using them as the activation functions for a neural network at the same time might have complementary advantages and hence improve the performance of the neural network. In other words, using hybrid activation functions in a neural network might have better performance than using only one kind of activation function. At present, different activation functions have been used in different hidden layers of a neural network when it has more than one hidden layer. However, using different activation functions in the same hidden layer has not been researched yet. In this thesis, a series of experiments have been conducted to discover how hybrid activation functions in the same hidden layer can be used to improve the performance of a neural network.

To further understanding the hybrid activation functions in LSTM networks, a detailed explanation is shown in Figure 4.12. For example, there is a LSTM neural network that has in total 100 LSTM neurons in the hidden layer. As a standard LSTM, all the hidden neurons will use the hyperbolic tangent activation function. However, for a hybrid activation functions based LSTM, not all the activation functions used are hyperbolic tangent. Among 100 neurons, X number of neurons utilise the hyperbolic tangent activation function and Y number of neurons use the sigmoid activation function, where X+Y = 100. The ratio of these two types of activation functions is X:Y. The purpose of this research is to investigate whether this kind of hybrid can improve the performance of the LSTM network.

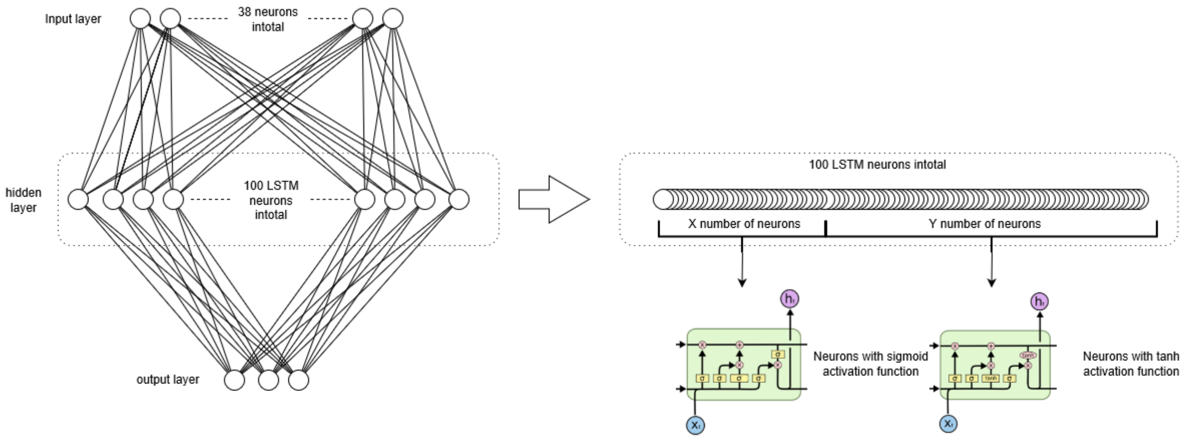


Figure 4.12 Hybrid activation functions in a LSTM network

4.2.3 Performance Evaluation of Hybrid Activation Functions

According to some pilot experiments, the performance of a neural network with hybrid activation functions relates to the ratio of the two different activation functions.

Table 4.1 Pilot experiments on ratio vs accuracy

Accuracy vs Ratio									
Ratio	1:9	2:8	3:7	4:6	5:5	6:4	7:3	8:2	9:1
Accuracy	0.674	0.732	0.772	0.725	0.757	0.793	0.797	0.812	0.812

Table 4.1 shows the record of the pilot study which has tested the accuracy with different ratios. The result shows that the larger the ratio (hyperbolic tangent: sigmoid), the higher the accuracy. However, we cannot conclude so hastily, because there are potential problems of over-fitting and randomness of neural networks. The randomness of a neural network is mainly caused by the randomly initialised weights of the network. In general, fixing the initial weights of a neural network might limit the learning ability of the neural network. The randomness of initial weights is a double-edged sword: the advantage is that it can search for more possible solutions and the disadvantage is that it is hard to obtain the real performance. In order to overcome the

randomness of the neural network, generally, an experiment needs to be repeated multiple times to get the average performance. However, for the over-fitting problem, it will not be solved such easily. The over-fitting problem, in this case, means that the results might only fit the current test set. In other words, we cannot apply the trained model or associated hyperparameters to other data sets. The test set is only used for evaluating the performance of the selected model on unseen data. In conclusion, the relationship between the ratio and the neural network performance needs to be further investigated.

To implementing the basic idea of using hybrid activation functions in the same hidden layer and evaluating how it can be used in the real neural network training process, two hypotheses have been raised:

Hypothesis 1. The ratio of the two different activation functions can be treated as a hyperparameter when designing the structure of the neural network. The ratio can be tuned by the validation set before neural network training.

Hypothesis 2. The neurons can be split into two groups. The first group contains most of the neurons (90% of total neurons) in the hidden layer, and the neurons keep the same activation function as usual. This group is responsible for the main training process. Moreover, there is another group that contains a small amount (10%) of neurons, which use another suitable activation function. Such a specific ratio can improve the performance of the neural network. Here, this small group of neurons with sigmoid activation function can be treated as 'seasoning' for the neural network, which will not affect the main training process but help improve the neural network performance to a certain degree.

4.2.4 Verification of Hypotheses

The hypotheses can be verified as correct if the learning performance with hybrid activation functions is better than the standard LSTM neural network. For comparison, the evaluation method is kept same as that introduced in Chapter 3and all the hyperparameters are kept same as well, including the time steps, number of hidden layers, number of hidden neurons, batch size and learning rate. In addition, the early stopping technique has been applied for model selection to avoid overfitting.

4.2.4.1 Experiment for Hypothesis 1

The first hypothesis supposes that the ratio of the hyperbolic tangent function and sigmoid function is a hyperparameter which needs to be tuned by the validation set before neural network training. A suitable choice for ratio will lead to a better performance of the neural network. In order to verify this hypothesis, an experiment has been designed as follow:

Step 1: Use the same hyperparameters as those used in Chapter 3, i.e., one hidden layer with 100 neurons, default fixed learning rate and 512 batch size.

Step 2: Use the same training set, validation set and test set as those used in Chapter 3.

Step 3: Set up the same early stopping technique as that used in Chapter 3.

Step 4: Split the 100 neurons in the LSTM layer into 10 portions, i.e., 10 neurons in each portion

Step 5: Make a possible ratio list for the hyperbolic tangent and the sigmoid activation functions, i.e., 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2, 9:1

Step 6: Do 10-fold cross-validation for each possible ratio in the list, choose a ratio with the highest accuracy.

Step 7: Use this ratio, do a performance evaluation on the test set.

Step 8: Repeat step 7 for 50 times and calculate the average accuracy performance.

There are two reasons for splitting the 100 neurons to 10 portions in Step 4:

1. During the general hyperparameter tuning process for the number of neurons, the numbers to consider are usually a multiplication of 10. There is no obvious difference between the two numbers of neurons whose difference is less than 10.
2. Due to that there are 100 neurons, -testing all the possible ratios for 100 neurons is very time-consuming.

For this reason, a possible ratio list has been generated in Step 5 with 9 different ratios. As a result of this experiment, the average validated accuracy from 10-fold cross-validation for each ratio is shown in Table 4.2.

Table 4.2 10-fold cross validation for different ratios

	10-fold cross validation for Combination Ratio								
tanh:sigmoid	1:9	2:8	3:7	4:6	5:5	6:4	7:3	8:2	9:1
Accuracy	78.09%	72.11%	80.48%	81.27%	82.87%	82.87%	83.27%	83.27%	83.27%
	77.11%	67.87%	79.92%	78.31%	80.32%	79.12%	81.53%	82.33%	79.92%
	65.46%	66.67%	71.08%	75.10%	78.71%	77.11%	79.52%	77.51%	76.71%
	79.03%	69.35%	81.05%	81.45%	81.45%	79.44%	84.27%	83.47%	84.27%
	57.26%	66.94%	75.40%	74.60%	79.84%	79.03%	80.24%	82.26%	78.63%
	77.42%	76.61%	79.84%	66.53%	77.02%	80.65%	75.00%	82.66%	79.84%
	67.34%	74.19%	76.21%	76.21%	78.63%	79.03%	80.24%	77.42%	78.63%
	67.34%	71.77%	70.56%	77.42%	82.66%	81.05%	82.26%	86.69%	83.06%
	81.45%	77.82%	80.24%	81.05%	73.79%	80.24%	82.26%	83.47%	80.24%
	72.98%	81.05%	78.63%	77.42%	79.84%	81.85%	82.26%	81.45%	80.65%
Average	72.35%	72.44%	77.34%	76.94%	79.51%	80.04%	81.08%	82.05%	80.52%

It is clear that the ratio (8:2) has the highest average validation accuracy (82.5%) in the 10-fold cross validation than. For this reason, by using 8:2 as a ratio, 50 repeated experiments of performance evaluation on the test set have been undertaken. The average accuracy over the 50 runs is 78.49% (shown in Table 4.3), which is 0.49% improvement on the LSTMs used in Chapter 3.

Table 4.3 50 Prediction accuracy over 50 runs for ratio 8:2

Accuracy on the test set					
tanh:sigmoid	8:2				
Accuracy	0.783	0.793	0.793	0.786	0.793
	0.797	0.793	0.790	0.772	0.812
	0.779	0.775	0.779	0.797	0.786
	0.801	0.775	0.808	0.783	0.804
	0.768	0.783	0.801	0.768	0.801
	0.772	0.808	0.815	0.804	0.779
	0.783	0.790	0.743	0.761	0.764
	0.746	0.786	0.783	0.801	0.786
	0.783	0.761	0.786	0.768	0.783
	0.793	0.790	0.790	0.783	0.768
Average	0.784928				

T-test has been applied in order to test whether the 0.49% improvement is significant. So the null hypothesis (H_0) and the alternate hypothesis (H_A) are:

H_0 : There is no statistically significant difference between the prediction accuracies of the hybrid activation function based LSTM neural networks with a 8:2 ratio and the new validation method based LSTM neural networks.

H_A: There is a statistically significant difference between the prediction accuracies of the hybrid activation function based LSTM neural networks with a 8:2 ratio and the new validation method based LSTM neural networks.

There is a 95% confidence interval, and the alpha level is 0.05.

The t-test can be realised by the t.test function in Excel. The result of t.test is a number (alpha level). If this number is higher than 0.05, the result shows there is not a significant difference between the two learning algorithms. On the contrary, if this number is lower than 0.05, it generally means that there is a significant difference.

The alpha level in this t-test is 0.138250022, which is bigger than 0.05. The test fails to reject the null hypothesis. So there is no statistically significant difference between the two sets of data. This result shows that using the ratio as a hyperparameter cannot improve the performance of the neural network significantly.

Table 4.4 Test accuracy for ratio 8:2 vs standard LSTM neural network

Accuracy in the test set										
tanh:sigmoid	8:2					10:0				
Accuracy	0.783	0.783	0.793	0.786	0.793	0.765	0.765	0.794	0.805	0.808
	0.797	0.797	0.790	0.772	0.812	0.783	0.783	0.772	0.768	0.805
	0.779	0.779	0.779	0.797	0.786	0.801	0.801	0.786	0.768	0.790
	0.801	0.801	0.808	0.783	0.804	0.794	0.794	0.761	0.761	0.776
	0.768	0.768	0.801	0.768	0.801	0.736	0.736	0.772	0.757	0.790
	0.772	0.772	0.815	0.804	0.779	0.794	0.794	0.754	0.797	0.776
	0.783	0.783	0.743	0.761	0.764	0.794	0.794	0.805	0.783	0.801
	0.746	0.746	0.783	0.801	0.786	0.801	0.801	0.761	0.754	0.797
	0.783	0.783	0.786	0.768	0.783	0.786	0.786	0.768	0.772	0.808

	0.793	0.793	0.790	0.783	0.768	0.786	0.787	0.761	0.776	0.768
Average	0.784928					0.780				

4.2.4.2 Experiment for Hypothesis 2

The second hypothesis supposes that change a small amount (10%) of neurons with sigmoid activation function will improve the performance of the LSTM neural network. The small number of neurons with the sigmoid function will not affect the main training process, but help improve the performance of the neural network in its way.

The success of such a technique could be explained as the minor demand for different activation functions in LSTM neurons. The default hyperbolic tangent activation function ranges from -1 to 1. In 100 LSTM neurons, there must be a few of neurons that do not need the activation function to have a full range from -1 to 1, but a range from 0 to 1. Moreover, different activation functions bring different derivatives to a neural network. Such hybrid activation functions might give positive contributes for a neural network to fulfil its task.

In order to verify this hypothesis, an experiment has been designed as follows:

Step 1: Use the same hyperparameters as those used in Chapter 3, i.e., one hidden layer with 100 neurons, default fixed learning rate and 512 batch size.

Step 2: Use the same training set, validation set and test set as those used in Chapter 3.

Step 3: Set up the same early stopping technique as that used in Chapter 3.

Step 4: Split the 100 neurons in the LSTM layer into 10 portions, i.e., 10 neurons in each portion.

Step 5: Use this ratio 9:1, do a performance evaluation on the test set.

Step 6: Repeat step 5 for 50 times and calculate the average accuracy.

The experimental results of the above test are shown in Table 4.5.

Table 4.5 Accuracy test for ratio 9:1 and 10:0

Accuracy in the test set										
tanh:sigmoid	9:1(HAF)					10:0(standard LSTM)				
Accuracy	0.797	0.797	0.783	0.797	0.786	0.765	0.765	0.794	0.805	0.808
	0.812	0.775	0.779	0.754	0.783	0.783	0.783	0.772	0.768	0.805
	0.772	0.783	0.746	0.786	0.790	0.801	0.801	0.786	0.768	0.790
	0.797	0.804	0.801	0.783	0.804	0.794	0.794	0.761	0.761	0.776
	0.812	0.797	0.783	0.768	0.772	0.736	0.736	0.772	0.757	0.790
	0.764	0.819	0.783	0.783	0.779	0.794	0.794	0.754	0.797	0.776
	0.797	0.779	0.797	0.804	0.815	0.794	0.794	0.805	0.783	0.801
	0.801	0.804	0.783	0.804	0.746	0.801	0.801	0.761	0.754	0.797
	0.797	0.764	0.783	0.793	0.801	0.786	0.786	0.768	0.772	0.808
	0.808	0.761	0.779	0.790	0.779	0.786	0.787	0.761	0.776	0.768
Average	0.787464					0.780				

As we can see from the table, the average accuracy on the test set has been improved to 78.75%, which has a 0.75% improvement on the LSTMs used in Chapter 3.

To check whether this is a significant improvement, t-test has also been applied. In this test, the null hypothesis (H_0) and the alternate hypothesis (H_A) are:

H_0 : There is no statistically significant difference between the prediction accuracies of the hybrid activation function based LSTM neural network with a 9:1 ratio and the new validation method based LSTM neural network.

H_A: There is a statistically significant difference between the prediction accuracies of the hybrid activation function based LSTM neural network with a 9:1 ratio and the new validation method based LSTM neural network.

There is a 95% confidence interval, and the alpha level is 0.05.

The t-test (shown in Table 4.4) for two ratios 9:1 and the ratio 10:0 (the new validation method based LSTM neural network) gives an alpha level of 0.0316, which is smaller than 0.05. As a result, the null hypothesis is rejected. There is a statistically significant difference between the two populations.

4.2.5 Discussions

The results of the above two experiments have shown that hybrid activation functions can be applied to a small group (10%) of hidden neurons in the LSMT neural network. Comparing with the new validation method based LSTMs in Chapter 3, the performance of the LSTM neural network on forex prediction has been improved by 1.02% when using hybrid activation functions.

Moreover, T-tests have been done to prove that there is a significant difference between standard LSTM and HAF based LSTM neural networks. A box and whisker plot has been drawn to show the difference and improvement between these two kinds of neural networks (Figure 4.13).

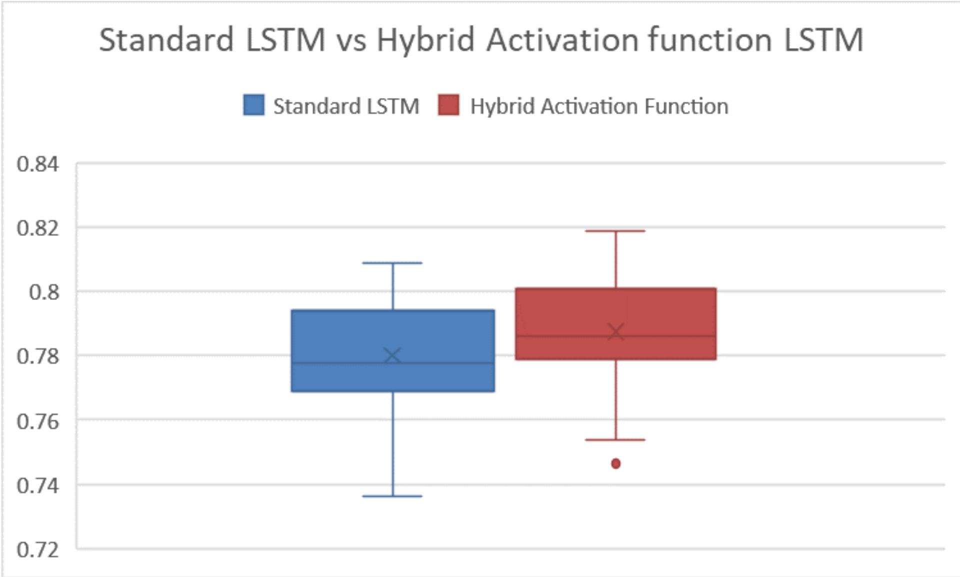


Figure 4.13 standard LSTMs vs LSTMs with hybrid activation functions

Such an improvement is also meaningful. The significance can be explained in three folds:

1. For financial prediction problems, especially for forex trading, 1.02% improvement on predictions can be significantly magnified by large trading volumes and high trading frequencies.
2. For the neural network, the significance lies in the easy implementation of HAF and its easy combination with other NN properties for performance improvement. In the current literature, the activation functions of all the neurons in the same hidden layer are always considered to be the same. This research is the first to study the use of different activation functions in the same hidden layer of a neural network.
3. The existing research on LSTMs have been extensive and it is a challenge to improve it further, particularly for complex problems such as forex forecasting. The use of HAF has superseded the most advanced LSTMs with statistically significant improvements, which is a meaning contribution to the current literature.

In summary, the success of HAF indicates that the performance of the neural network can be improved by assigning a small group (e.g., 10%) of hidden neurons to a different activation function.

4.3 Generalisation of the HAF Method

In the previous section, the HAF method has been applied to LSTMs. HAF is actually suitable for other types of ANNs that have hidden layers. In order to test the generalisation of this HAF method, two experiments have been done. The first experiment has been undertaken by applying the HAF method to the standard recurrent neural network (RNN). Although the RNN is not the best network for forex prediction, the application of the HAF to the RNN will test the generalisation of the HAF method. The other experiment is to apply the HAF method with LSTMs on a different dataset. This section will begin with the first experiment.

4.3.1 Application of HAFs to RNNs

4.3.1.1 Standard RNNs

To build a standard RNN for forex forecasting, the hyperparameters of the RNN needs to be tuned first by the use of the validation set.

Same as the experiments in Chapter 3, the USD/CNY trading records are split into the training, validation and test sets. The hyperparameter tuning starts from timesteps, followed by the number of hidden layers, number of neurons in the hidden layer, batch size and learning rate. The order of tuning hyperparameters is decided by heuristic experiences. 10-fold cross-validation has been applied for tuning the hyperparameters. The initial values for the hyperparameters are set as: 100 neurons in one hidden layer, 512 for the mini-batch size, and the default value for the learning rate. The hyperparameters are then tested one-by-one, that is, different values of one hyperparameter are tested while the other hyperparameters remain the same. The tuning results are shown below:

Time steps:

Table 4.6 Time steps tuning results

Cross Validation										
Time steps	90	80	70	60	50	40	30	20	10	1
Accuracy in cross validation	77.29%	76.49%	75.20%	76.80%	73.60%	77.20%	74.80%	77.60%	76.40%	59.60%
	76.31%	74.40%	69.60%	77.20%	77.11%	75.60%	77.20%	75.50%	74.40%	58.40%
	78.71%	79.44%	74.80%	76.31%	73.90%	75.50%	73.90%	73.09%	74.80%	54.00%
	75.40%	78.23%	73.49%	75.00%	75.50%	73.90%	76.71%	75.50%	68.67%	64.26%
	73.39%	75.81%	79.12%	77.02%	77.11%	75.50%	76.71%	77.42%	71.37%	62.10%
	78.23%	76.21%	81.05%	71.37%	77.11%	78.23%	80.65%	74.19%	70.97%	58.87%
	70.97%	72.98%	73.39%	77.42%	77.82%	70.16%	75.00%	73.79%	71.37%	63.71%
	75.00%	79.44%	74.49%	75.00%	73.79%	77.82%	78.23%	76.21%	72.58%	57.66%
	74.19%	75.81%	77.73%	77.82%	72.87%	78.54%	75.30%	73.39%	70.85%	64.37%
	72.18%	71.77%	79.76%	80.65%	75.71%	76.52%	74.49%	76.21%	74.90%	61.94%
Average	75.17%	76.06%	75.86%	76.46%	75.45%	75.90%	76.30%	75.29%	72.63%	60.49%

The results show that, when the value of time steps equals to 60, the RNN has the highest average prediction accuracy in 10-fold cross validation. As a result, the number of time steps is kept at 60 when tuning the other hyperparameters.

Number of hidden layers

Table 4.7 Number of hidden layers tuning results

Cross- Validation		
Number of layers	1 layer	2 layers
	76.80%	76.80%

Accuracy in cross validation	77.20%	74.40%
	76.31%	73.49%
	75.00%	75.40%
	77.02%	75.40%
	71.37%	70.16%
	77.42%	75.00%
	75.00%	74.19%
	77.82%	77.42%
	80.65%	73.39%
Average	76.46%	74.57%

The results show that, when there is only one hidden layer, the RNN has the highest average accuracy in 10-fold cross validation. As a result, one is kept as the default value for the number of hidden layers.

<u>Number of neurons in the hidden layer</u>

Table 4.8 Number of hidden neurons tuning results

Cross- Validation				
number of neurons	50	100	150	200
Accuracy in cross validation	75.20%	76.80%	76.80%	76.00%
	75.20%	77.20%	74.00%	79.20%
	71.89%	76.31%	76.31%	76.71%
	71.37%	75.00%	76.61%	72.58%
	71.37%	77.02%	76.21%	76.61%
	69.35%	71.37%	76.61%	70.56%
	70.56%	77.42%	79.44%	77.02%
	71.77%	75.00%	76.61%	72.98%
	74.60%	77.82%	74.60%	76.61%
	76.61%	80.65%	80.65%	73.79%

Average	72.79%	76.46%	76.78%	75.21%

The results show that, when there are 150 neurons in a hidden layer, the RNN has the highest average accuracy in 10-fold cross validation. As a result, the default value for the number of hidden neurons has been updated to 150.

Batch size

Table 4.9 Batch size tuning results

Cross- Validation				
batch size	256	512	1024	full batch
Accuracy in cross validation	74.40%	76.80%	77.60%	79.60%
	75.60%	74.00%	72.00%	76.00%
	70.28%	76.31%	75.50%	74.70%
	69.76%	76.61%	75.81%	76.21%
	74.19%	76.21%	75.00%	77.82%
	72.18%	76.61%	75.00%	73.39%
	75.81%	79.44%	77.42%	80.24%
	72.98%	76.61%	74.19%	75.00%
	72.18%	74.60%	77.42%	78.63%
	71.77%	80.65%	77.02%	78.63%
Average	72.92%	76.78%	75.70%	77.02%

The results show that, when using the full size as the batch size, the RNN has the highest average accuracy in 10-fold cross validation. As a result, the default value of the batch size has been updated by the full size.

Learning rate

Table 4.10 Learning rate tuning results

Cross- Validation		
Learning rate	Decay Learning Rate	Fixed Learning Rate (default)
Accuracy in cross validation	78.80%	79.60%
	71.20%	76.00%
	75.10%	74.70%
	77.42%	76.21%
	74.60%	77.82%
	74.19%	73.39%
	78.63%	80.24%
	74.60%	75.00%
	75.00%	78.63%
	78.23%	78.63%
Average	75.78%	77.02%

The results show that, when using the default fixed learning rate, it has the highest average accuracy in 10-fold cross validation. As a result, the default value of the learning rate keeps the same.

The hyperparameter tuning results for the RNN are that 100 hidden neurons in one hidden layer, 60 for the number of time steps, 512 for the batch size and the default value for the learning rate. In addition, the early stopping technique has been applied to stop the training rather than choosing a specific number for epochs. By using such a RNN to learning forex predictions, the training has been repeated by 50 times, and the prediction accuracies on the test set have been recorded (shown in Table 4.11). The average accuracy over 50 repeated experiments is 71.64%.

Table 4.11 Prediction accuracies of the standard RNN

Accuracy in the test set

Accuracy	0.712	0.696	0.693	0.744	0.750
	0.709	0.747	0.703	0.709	0.753
	0.687	0.709	0.718	0.703	0.690
	0.728	0.722	0.684	0.718	0.728
	0.718	0.715	0.756	0.715	0.731
	0.731	0.677	0.658	0.731	0.687
	0.699	0.731	0.699	0.747	0.706
	0.709	0.759	0.658	0.744	0.715
	0.718	0.759	0.706	0.712	0.741
	0.753	0.668	0.741	0.756	0.677
Average	0.716392405				

4.3.1.2 Applying the HAF to the RNN

In a standard RNN, there is only one activation function used for the neurons. The default activation function for an RNN is the hyperbolic tangent. When HAF is applied to the RNN, 10% of the hidden neurons use a different activation function which is the sigmoid function. So the ratio between the use of the hyperbolic tangent and sigmoid functions is still 9:1. Because there are 150 neurons in the hidden layer of the RNN, these hidden neurons are split into 10 portions, each of which has 15 neurons. Accordingly, 9*15 =135 neurons are equipped with the hyperbolic tangent activation function and the remaining 15 neurons use the sigmoid activation function. so that it will be Moreover, the 9:1 ratio means, there will be

4.3.1.3 Generalisation Evaluation on the RNN

The experimental results of the forex trend predictions for the USD/CNY pair by using the standard RNN and HAF based RNN are shown in Table 4.12 (over 50 runs).

Table 4.12 HAF based RNN vs standard RNN

Accuracy on the test set										
tanh:sigmoid	10:0(standard RNN)					9:1(HAF based)				
Accuracy	0.712	0.696	0.693	0.744	0.750	0.734	0.747	0.722	0.785	0.763
	0.709	0.747	0.703	0.709	0.753	0.772	0.759	0.744	0.763	0.709
	0.687	0.709	0.718	0.703	0.690	0.737	0.744	0.747	0.737	0.722
	0.728	0.722	0.684	0.718	0.728	0.750	0.778	0.756	0.709	0.722
	0.718	0.715	0.756	0.715	0.731	0.728	0.703	0.734	0.753	0.703
	0.731	0.677	0.658	0.731	0.687	0.750	0.747	0.759	0.750	0.756
	0.699	0.731	0.699	0.747	0.706	0.734	0.772	0.737	0.709	0.731
	0.709	0.759	0.658	0.744	0.715	0.763	0.769	0.744	0.737	0.718
	0.718	0.759	0.706	0.712	0.741	0.718	0.728	0.728	0.725	0.747
	0.753	0.668	0.741	0.756	0.677	0.722	0.706	0.731	0.725	0.747
Average	0.71639					0.73943				

As we can see from the table, the average prediction accuracy of the RNN on the test set has been improved to 73.94%, which has 2.3% improvement on the standard RNN. In order to check whether there is a significant improvement, t-test has been applied. In this case, the null hypothesis (H_0) and the alternate hypothesis (H_A) are:

> H_0 : There is no statistically significant difference between the prediction accuracies of the hybrid activation functions based RNN neural network with a 9:1 ratio and the standard RNN neural network.

H_A: There is a statistically significant difference between the prediction accuracies of the hybrid activation functions based RNN neural network with a 9:1 ratio and the standard RNN neural network.

There is a 95% confidence interval, and the alpha level is 0.05.

The t-test result (shown in Table 4.12) generates an alpha level of 0.000004363, which is smaller than 0.05. As a result, the null hypothesis has been rejected. There is a statistically significant difference between the two populations.

In conclusion, the HAF technique has been successfully applied to RNN; there is a 3.2% performance improvement by applying HAF. The effectiveness and generalisation of HAF have been proved by applying to both the LSTM network and RNN. This finding verifies that HAF can be used not only in LSTM but also some other types of ANNs such as the RNN.

4.3.2 Generalisation of HAFs to Different Datasets

In Section 3.2.5, the USD/GBP trading data is used as a different dataset to verify the reliability of the new validation method. Here, this dataset is used again to test how HAF performs on a different dataset too. Section 3.2.5 has introduced the USD/GBP dataset in detail. The USD/CNY and USD/GBP datasets are similar but the numbers of the features chosen for training and the number of samples are different. The hyperparameters of the LSTM ned to re-tuned because of the change of the dataset.

The process for hyperparameter tuning is the same as before. The tuning results are listed below:

Time steps:

Table 4.13 Time steps tuning results

Cross Validation										
Time steps	20	30	40	50	60	70	80	90	100	110
Accuracy in cross validation	78.62%	80.29%	79.25%	82.39%	82.18%	83.65%	80.67%	77.36%	81.30%	82.60%
	78.83%	76.94%	78.83%	84.07%	81.13%	77.78%	83.40%	84.24%	79.20%	83.86%
	77.15%	80.88%	81.76%	80.67%	79.45%	83.86%	83.19%	83.19%	84.03%	84.28%
	77.73%	80.04%	79.83%	80.25%	82.35%	81.13%	78.78%	79.62%	83.61%	79.41%
	77.52%	77.94%	80.88%	80.88%	81.30%	86.34%	83.19%	82.77%	84.87%	81.30%
	76.26%	79.41%	78.36%	80.46%	80.67%	81.47%	81.09%	82.35%	81.09%	81.26%
	80.88%	81.09%	82.77%	80.88%	80.21%	81.26%	84.03%	80.46%	83.82%	82.95%
	77.64%	77.89%	81.65%	77.89%	82.32%	81.26%	82.77%	81.93%	83.61%	81.47%
	76.16%	80.38%	79.75%	82.28%	82.49%	84.81%	82.53%	84.18%	82.53%	82.74%
	76.58%	81.22%	78.69%	80.38%	81.22%	81.65%	83.12%	82.07%	82.07%	82.28%
Average	77.74%	79.61%	80.18%	81.02%	81.33%	82.32%	82.28%	81.82%	82.61%	82.21%

The number of hidden layers:

Table 4.14 The number of hidden layers tuning results

Cross- Validation			
Number of layers	1	2	3
Accuracy in cross validation	81.30%	80.04%	79.62%
	79.20%	79.20%	75.84%
	84.03%	82.98%	81.09%
	83.61%	81.51%	81.72%
	84.87%	83.40%	78.57%
	81.09%	80.88%	77.94%
	83.82%	82.56%	79.83%
	83.61%	81.93%	81.51%
	82.53%	81.26%	80.84%

	82.07%	81.22%	79.75%
Average	82.61%	81.50%	79.67%

The number of hidden neurons:

Table 4.15 The number of neurons tuning results

Cross- Validation				
number of neurons	100	200	300	400
Accuracy in cross validation	81.30%	80.04%	80.30%	81.51%
	79.20%	78.57%	80.46%	78.78%
	84.03%	84.03%	84.24%	83.82%
	83.61%	83.40%	82.35%	82.35%
	84.87%	85.50%	83.19%	81.51%
	81.09%	80.67%	80.04%	80.46%
	83.82%	83.61%	83.19%	82.35%
	83.61%	82.56%	82.35%	83.40%
	82.53%	82.32%	84.00%	82.53%
	82.07%	82.91%	80.38%	81.22%
Average	82.61%	82.36%	82.05%	81.80%

Batch size:

Table 4.16 Batch size tuning results

Cross- Validation							
batch size	64	128	256	512	1024	2048	6000
Accuracy in cross validation	81.09%	81.09%	81.72%	81.30%	80.67%	82.14%	79.20%
	78.78%	78.57%	78.99%	79.20%	78.36%	76.89%	75.63%
	81.09%	83.19%	83.82%	84.03%	85.29%	83.82%	82.35%
	82.14%	83.40%	82.98%	83.61%	83.19%	83.82%	80.88%

	78.78%	83.82%	83.40%	84.87%	83.82%	83.61%	79.41%
	79.20%	80.88%	80.88%	81.09%	80.25%	79.83%	77.94%
	81.30%	82.77%	82.35%	83.82%	81.72%	82.35%	54.41%
	81.30%	82.98%	82.77%	83.61%	83.19%	82.35%	81.93%
	82.53%	81.89%	83.16%	82.53%	81.68%	81.47%	81.89%
	81.22%	81.86%	82.49%	82.07%	80.59%	81.43%	78.69%
Average	80.74%	82.05%	82.26%	82.61%	81.88%	81.77%	77.24%

As a result, the hyperparameters of the LSTM neural network for the USD/GBP dataset are tuned to have 100 time steps, 1 hidden layer, 100 neurons in the hidden layer and 512 mini batch-size.

When applying the HAF method to the LSTM, the 100 neurons in the hidden layer are split into 90 neurons equipped with the hyperbolic tangent activation function and 10 neurons equipped with the sigmoid activation function. The training and evaluation of the HAF based LSTM and the standard LSTM neural network have been repeated by 50 times. The results are listed below:

Table 4.17 Standard LSTM vs HAF LSTM (USD/GBP dataset)

Accuracy on the test set										
tanh:sigmoid	10:0(standard LSTM)					9:1(HAF based)				
Accuracy	0.7713	0.7864	0.8053	0.7732	0.7921	0.7788	0.7864	0.7902	0.7996	0.7826
	0.7921	0.7675	0.7637	0.7902	0.7921	0.7769	0.7769	0.7883	0.7902	0.7656
	0.7845	0.7769	0.7958	0.7732	0.7694	0.7845	0.7524	0.7769	0.7713	0.7958
	0.7807	0.7958	0.7656	0.7845	0.7656	0.7883	0.7958	0.7826	0.7826	0.7807
	0.7750	0.7713	0.7807	0.8053	0.7977	0.7902	0.7732	0.8015	0.7769	0.7845
	0.7750	0.7807	0.7845	0.7940	0.7864	0.7883	0.7883	0.8015	0.7826	0.7977

	0.7807	0.7750	0.7732	0.7883	0.5690	0.7864	0.7826	0.8015	0.7921	0.7675
	0.7656	0.7713	0.7845	0.7864	0.7732	0.7656	0.7921	0.7902	0.7940	0.7826
	0.7977	0.7921	0.7675	0.7750	0.7618	0.7826	0.7713	0.7732	0.7769	0.7845
	0.7769	0.7750	0.7599	0.7769	0.7694	0.7845	0.7845	0.7883	0.7996	0.7845
Average	0.775916825					0.784310019233				

The significance of the better performance of HAF based LSTM is checked by t-test. In this case, the null hypothesis (H_0) and the alternate hypothesis (H_A) are:

H_0 : There is no statistically significant difference between the prediction accuracies of the hybrid activation functions based LSTM neural network with a 9:1 ratio and the standard LSTM neural network.

H_A: The prediction accuracy of the hybrid activation functions based LSTM neural network with a 9:1 ratio is higher than the standard LSTM neural network.

There is a 95% confidence interval, and the alpha level is 0.05.

The alpha level of the t-test is 0.042644666, which is smaller than 0.05. As a result, the null hypothesis has been rejected. The accuracy by using HAF based LSTM neural network is higher than the standard LSTM neural network.

In conclusion, the HAF technique has been successfully applied to a different dataset; there is a 0.84% accuracy improvement on the standard LSTM. This improvement has been proved to be statistically significant by T-test.

4.3.3 Discussions

By applying the hybrid activation function (HAF) to different types of neural networks and different datasets, the generalisation of the HAF method has been successfully proved.

For HAF’s application on the RNN, the prediction accuracy is improved from 71.6% to 73.9%. Compared with the application of HAF to the LSTM, the improvement on the RNN by the use of HAF is more than that on the LSTM (78% to 78.74%). It is mainly because LSTMs have strong learning capabilities than RNNs, particularly when solving complex time series problems. Forex trading is often affected by special events such as news, natural disasters, wars and disease, so it is very difficult to achieve a high prediction accuracy. But the use of the HAF in LSTMs and RNNs have helped both performed better, when their standard algorithms have achieved the limits in predictions. Even the weaker RNN has shown more improvements. This indicates that hybrid activation functions may be a new promising approach to enhance neural networks’ learning in an inherent way.

For HAF’s application to a different dataset, the improvements achieved by the HAF based LSTMs are similar for both forex pairs, i.e., USD/CNY and USD/GBP. This similarity shows that the HAF method provides an improvement on the inherent learning capability of neural networks, which is not affected by the training data. Therefore, HAF can be generalised to any time series data when combined with an ANN such as the RNN or LSTM.

Chapter Summary

This chapter has firstly introduced the activation function and compared the artificial neural network with the human neural network. The inspirations of using hybrid activation functions in an artificial neural network have also been explained. The idea has been applied to the LSTM neural network after investigating the activation functions in LSTM neurons. In a pilot experiment, the performances of the hybrid activation functions when they are used in different ratios are initially investigated.

For further investigation, two hypotheses are made about hybrid activation functions and two experiments have been designed to verify the hypotheses accordingly. According to the experimental results, there is no significant improvement if the ratio of the different activation functions used is tuned as a hyperparameter before training. However, there is a significant improvement if a fixed small amount (10%) of neurons are equipped with a different activation function.

As a result, the hybrid activation functions with a 9:1 ratio have been used to improve the performance of the LSTM neural network. It has achieved a 78.75% average accuracy in forex predictions, which is a 1.02% improvement on the standard LSTM. HAF has been tested on different types of ANNs (e.g., RNNs) and different datasets (e.g., a different forex pair in forex forecasting). The experimental results have shown improvements in all the scenarios, compared with the standard ANNs which have only employed one kind of activation function. The successful applications of HAF to different ANNs and datasets have demonstrated its generalisability.

The proposed HAF technique is a meaningful contribution to ANN learning; the significance lies in its application to different ANNs and its robust capabilities to improve the learning performance of ANNs. Particularly when deep learning algorithms such as RNNs and LSTMs have reached their limitations on complex problems such as forex time series predictions, the use of HAF have shown statistically significant improvements on the standard learning algorithms. Hybrid activation functions have brought a new way to improve ANN learning, which is worth further research.

Chapter 5 Conclusions and Future Work

5.1 Conclusions

Forex prediction continues to be a popular research topic because there are huge profits and great economic benefits in it. Academically, forex prediction has also attracted great attention because this is a complex time series problem. Extensive studies have been developed for forex forecasting, including statistical approaches and machine learning methods.

In this thesis, a comprehensive literature review on forex forecasting has been conducted first. Data from both fundamental and technical analysis has been investigated and it is proposed that it will be beneficial to use indicators of both analysis methods to make forex predictions. For this reason, this research has collected all the data related to fundamental analysis and technical analysis and studied the data through comprehensive feature analyse. By the use of Decision-Tree-based feature selection, 39 indicators have been identified to have a strong impact on the forex trading for the USD/CNY pair. These indicators should be used as the inputs for machine learning, e.g., the Long Short Term Memory (LSTM) which is especially suitable for time series predictions.

The prediction targets for forex forecasting have also been studied. The benefits of different prediction targets, the real demands of forex traders and the practical use of intelligent predictions for forex trading are discussed in detail. Predicting the forex price trend for a future day (e.g., the next 5^{th} day) is suggested to be a proper prediction target, which will give forex traders reliable and realistic suggestions when they make their trading strategies.

In order to overcome the overfitting problem, the limitations of the current validation methods used in ANN training have been reviewed and consequently a new validation method for time series data has been designed. This new validation method converts time series data to a set of independent samples and then applies the holdout and cross validation methods to validate the training of a learning algorithm. Compared with the traditional walk-forward validation method, this new validation method is time-efficient and able to find more generalised solutions which are not overtrained by particular samples.

With the new validation method, a LSTM neural network based prediction method has been successfully developed to predict forex price trends. By using information from both fundamental and technical analysis, an excellent average prediction accuracy (78%) has been achieved for the USD/CNY pair, which is better than using the technical data only (76%). The performance of the new validation method based LSTMs have also superseded the LSTMs with the traditional walk-forward validation method. To further improve learning, the idea of deploying hybrid activation functions (HAF) in the same hidden layer of the LSTM has been proposed, inspired by biological neural networks. When the HAF is applied to LSTMs, it shows a better performance than standard LSTMs. Through detailed investigation and proofs, it is found out that the ratio of the two hybrid activation functions used needs to be kept at 9:1, which means only a small group (10%) of the hidden neurons is required to operate with a different activation function. By using hybrid activation functions, LSTMs have demonstrated a further improvement of 1.02% for USD/CNY predictions. Furthermore, even RNNs using HAFs has shown a 3.2% performance improvement. Similar improvements have been obtained when the same HAF based LSTM is used to predict for a different forex pair USD/GBP. The successful applications of HAFs to different types of ANNs and different datasets have proved their effectiveness and generalisability.

5.2 Limitations and Future Works

Due to the limited time, there are some limitations in this research, which can be addressed in further studies:

- This research has only used the USD/CNY and USD/GBP forex pairs as examples for forex forecasting. It would be better to test the new validation method based LSTMs and hybrid activation functions based LSTMs on some other popular forex pairs as well so as to validate the performances of these two new algorithms even more.
- The new validation method proposed in this research needs to be further verified, by testing on other kinds of time series data. Particularly, the time complexity of this new validation method and traditional walk-forward validation should be analysed and compared so as to fully confirm the feasibility of the new validation method for time series problems.
- Because the research topic of this thesis is forex forecasting, the proposed hybrid activation functions are only applied to forex time series. This method, however, should be generic for all kinds of ANNs that have hidden layers. As HAFs have shown its generalisability on LSTMs and RNNs and different datasets as proved by this research, further investigation is required to test HAFs' use on other kinds of ANNs and applications. This research is the first time that hybrid activation functions are used in the same hidden layer of an ANN; this new idea inspired by biological neural networks may provide a novel, promising way to enhance ANN learning.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ...& Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org.

Abardanell, Jeffery S., and Brian J. Bushee. 1997. Fundamental Analysis, Future Earnings, and Stock Prices.Journal of Accounting Research 35(1): 1–24.

Abednego, L., Nugraheni, C. E., & Rinaldy, I. (2018). Forex Trading Robot with Technical and Fundamental Analysis. Journal of Computers, 13(9), 1089-1098.

Abraham, A., & Chowdhury, M. U. (2001). An intelligent forex monitoring system. In 2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479) (Vol. 3, pp. 523-528). IEEE.

AbuHamad, M., Mohd, M., & Salim, J. (2013). Event-driven business intelligence approach for real-time integration of technical and fundamental analysis in forex market. Journal of Computer Science, 9(4), 488.

Admiral markets. (2019) “Forex predictions: how to predict the Forex market” admiralmarkets.com. Admiral Markets UK Ltd. Retrieved from https://admiralmarkets.com/education/articles/forex-basics/forex-predictions-how-to-predict-the-forex-market On 25 February 2019.

Admiral markets. (2019) “Understanding Forex Market Analysis” admiralmarkets.com. Admiral Markets UK Ltd. Retrieved from https://admiralmarkets.com/education/articles/forex-analysis/understanding-forex-market-analysis On 25 February 2019.

Admiral markets. (2019) “Forex vs. Stocks: Should You Trade Forex or Stocks?” admiralmarkets.com. Admiral Markets UK Ltd. Retrieved from https://admiralmarkets.com/education/articles/forex-basics/forex-vs-stocks-should-you-trade-forex-or-stocks On 25 February 2019.

Admiral markets. (2019) “Introduction to Forex technical analysis” admiralmarkets.com. Admiral Markets UK Ltd. Retrieved from https://admiralmarkets.com/education/articles/forex-analysis/introduction-to-forex-technical-analysis. On 25 February 2019.

Admiral markets. (2019) “Trading With the Cloud: Using the Ichimoku Kinko Hyo Indicator in MetaTrader 4” admiralmarkets.com. Admiral Markets UK Ltd. Retrieved from https://admiralmarkets.com/education/articles/forex-indicators/ichimoku-kinko-hyo-indicator. On 25 February 2019

Aliber, R. Z. (2000). Capital flows, exchange rates, and the new international financial architecture: Six financial crises in search of a generic explanation. Open economies review, 11(1), 43-61.

Altavilla, C., & De Grauwe, P. (2010). Forecasting and combining competing models of exchange rate determination. Applied Economics, 42(27), 3455-3480.

Alweshah, M., Rashaideh, H., Hammouri, A. I., Tayyeb, H., & Ababneh, M. (2017). Solving time series classification problems using support vector machine and neural network. International journal of data analysis techniques and strategies, 9(3), 237-247.

An, S., Ling, Z., & Dai, L. (2017, December). Emotional statistical parametric speech synthesis using LSTM-RNNs. In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1613-1616). IEEE.

Angadi, M.C. & Kulkarni, A.P. (2015). Time Series Data Analysis for Stock Market Prediction using Data Mining Techniques with R. Int. J. Adv. Res. Comp Science. 6(6), pp. 104-109.

Arévalo, A., Niño, J., Hernández, G., & Sandoval, J. (2016). High-frequency trading strategy based on deep neural networks. In International conference on intelligent computing (pp. 424-436). Springer International Publishing.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics surveys, 4, 40-79.

Armano, G.; Marchesi, M.; Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. Inf. Sci., 170, 3–33.

Asaithambi, S. (2018). Why, How and When to apply Feature Selection. Towards Data Science. Retrieved from https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2 on 15th March 2019

Aungiers, J. (2016). LSTM Neural Network for Time Series Prediction. Retrieved from http://www.jakob-aungiers.com/articles/a/LSTM-Neural-Network-for-Time-Series-Prediction.

Azzouni, A., & Pujolle, G. (2017). A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690.

Bakhshi, Z., & Ebrahimi, M. (2016). The effect of real exchange rate on unemployment. Marketing and Branding Research, 3(1), 4.

Bakker, B. (2007, April). Reinforcement learning by backpropagation through an LSTM model/critic. In 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (pp. 127-134). IEEE.

BANK OF CHINA. (2008). “个人外汇买卖常识—— 汇率分析篇”. www.bankofchina.com. BANK OF CHINA. Retrieved from http://www.bankofchina.com/custserv/cs7/200808/t20080814_1279.html on 25 February 2019.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long short-term memory. PloS one, 12(7), e0180944.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157-166.

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). An overview and comparative analysis of recurrent neural networks for short term load forecasting. arXiv preprint arXiv:1705.04378.

Bleikh, H. Y., & Young, W. (2014). Time series analysis and adjustment. Farnham: Gower.

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

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M., & Ljung, G. M. (2015). Time series analysis: Forecasting and control (Fifth edition ed.). New York: John Wiley & Sons, Incorporated.

Brockwell, P. J., Davis, R. A., & Calder, M. V. (2002). Introduction to time series and forecasting (Vol. 2). New York: springer.

Brown, N., Mundkowsky, R. & Shiu, Sam. (2013). Predicting Intraday Price Movements in the Foreign Exchange Market. Retrieved from http://cs229.stanford.edu/proj2011/BrownMundkowskyShiu%20-PredictingIntradayPriceMovementsInTheForeignExchangeMarket.pdf

Brownlee, J. (2017). How to Reshape Input Data for Long Short-Term Memory Networks in Keras. Machinelearningmastery. Retieved from https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/

Brownlee, J. (2018). A gentle introduction to k-fold cross-validation. Machinelearningmastery. Retieved from https://machinelearningmastery.com/k-fold-cross-validation/

Brownlee, J. (2018). When to Use MLP, CNN, and RNN Neural Networks. Machinelearningmastery. Retieved from https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/

Brownlee, J. (2019). Understand the Impact of Learning Rate on Neural Network Performance. Machinelearningmastery. Retieved from https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

Cai, Z., Chen, L., & Fang, Y. (2012). A new forecasting model for USD/CNY exchange rate. Studies in Nonlinear Dynamics & Econometrics, 16(3).

Cao, L. J., & Tay, F. E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks, 14(6), 1506-1518.

Chan, K. C., & Teong, F. K. (1995, November). Enhancing technical analysis in the Forex market using neural networks. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 2, pp. 1023-1027). IEEE

Chandra, S. R., & Al-Deek, H. (2009). Predictions of freeway traffic speeds and volumes using vector autoregressive models. Journal of Intelligent Transportation Systems, 13(2), 53-72.

Chatfield, C. (2016). The Analysis of Time Series: An Introduction, Sixth Edition. CRC Press, London.

Chatrath, A., Miao, H., Ramchander, S., & Villupuram, S. (2014). Currency jumps, cojumps and the role of macro news. Journal of International Money and Finance, 40, 42–62.

Chen, A. S., & Leung, M. T. (2004). Regression neural network for error correction in foreign exchange forecasting and trading. Computers & Operations Research, 31(7), 1049-1068.

Chen, C. T., & Chang, W. D. (1996). A feedforward neural network with function shape autotuning. Neural networks, 9(4), 627-641.

Chen, G.; Chen, Y. & Fushimi T. (2016). Application of Deep Learning to Algorithmic Trading. Retrieved from http://cs229.stanford.edu/proj2017/final-reports/5241098.pdf

Chen, J. (2019). Relative Strength Index – RSI Definition. Investopedia. Retrieved from https://www.investopedia.com/terms/r/rsi.asp.

Chen, S. H., Hwang, S. H., & Wang, Y. R. (1998). An RNN-based prosodic information synthesizer for Mandarin text-to-speech. IEEE transactions on speech and audio processing, 6(3), 226-239.

Chollet, F. (2015). Keras. https://keras.io

Chun, S. H., & Kim, S. H. (2004). Automated generation of new knowledge to support managerial decision-making: case study in forecasting a stock market. Expert Systems, 21(4), 192-207.

Chung, H. and Shin, K. (2018). Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction. Sustainability, 10(10), p.3765.

Cummins, F., Gers, F., and Schmidhuber, J. (1999). Automatic discrimination among languages based on prosody alone. Technical Report IDSIA-03-99, IDSIA, Lugano, CH.

De Matos, G. (1994). Neural networks for forecasting foreign exchange rates.

De Oliveira, J. F. L., & Ludermir, T. B. (2016). A hybrid evolutionary decomposition system for time series forecasting. Neurocomputing, 180, 27-34.

De Zwart, G., Markwat, T., Swinkels, L., & van Dijk, D. (2009). The economic value of fundamental and technical information in emerging currency markets. Journal of International Money and Finance, 28(4), 581-604.

Dinarelli, M., & Tellier, I. (2016, April). New recurrent neural network variants for sequence labeling. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 155-173). Springer, Cham.

Doetsch, P., Kozielski, M., & Ney, H. (2014, September). Fast and robust training of recurrent neural networks for offline handwriting recognition. In 2014 14th International Conference on Frontiers in Handwriting Recognition (pp. 279-284). IEEE.

Dunis, C. L., Laws, J., & Sermpinis, G. (2011). Higher order and recurrent neural architectures for trading the EUR/USD exchange rate. Quantitative Finance, 11(4), 615-629.

El Shazly, M. R., & El Shazly, H. E. (1999). Forecasting currency prices using a genetically evolved neural network architecture. International review of financial analysis, 8(1), 67-82.

Elliott, D. L. (1993). A better activation function for artificial neural networks.

ElSaid, A., Jamiy, F. E., Higgins, J., Wild, B., & Desell, T. (2018, July). Using ant colony optimization to optimize long short-term memory recurrent neural networks. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 13-20). ACM.

ElSaid, A., Jamiy, F.E., Higgins, J., Wild, B. and Desell, T. (2018). Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration. Applied Soft Computing, 73, pp.969-991.

Eng, M. H., Li, Y., Wang, Q. G., & Lee, T. H. (2008, December). Forecast forex with ANN using fundamental data. In 2008 International Conference on Information Management, Innovation Management and Industrial Engineering (Vol. 1, pp. 279-282). IEEE.

Fan, Y., Lu, X., Li, D., & Liu, Y. (2016, October). Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In Proceedings of the 18th ACM International Conference on Multimodal Interaction (pp. 445-450). ACM.

Findley, D. F., Martin, D. E., & Wills, K. C. (2002). Generalizations of the Box-Jenkins airline model. Census Bureau and Howard University, manuscrito.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Floares, A. G., Calin, G. A., & Manolache, F. B. (2016, June). Bigger Data Is Better for Molecular Diagnosis Tests Based on Decision Trees. In International Conference on Data Mining and Big Data (pp. 288-295). Springer, Cham.

Fuller W.A. (2009). Introduction to Statistical Time Series. Volume 428 of Wiley Series in Probability and Statistics. John Wiley & Sons.

Gahirwal, M. (2013). Inter time series sales forecasting. arXiv preprint arXiv:1303.0117.

Galeshchuk, S. and Mukherjee, S. (2017). Deep networks for predicting direction of change in foreign exchange rates. Intelligent Systems in Accounting, Finance and Management, 24(4), pp.100-110.

Ganegedara, T. (2018). Stock Market Predictions with LSTM in Python. Data Camp. Retrieved from https://www.datacamp.com/community/tutorials/lstm-python-stock-market.

Gao, Q. (2016). Stock market forecasting using recurrent neural network (Doctoral dissertation, University of Missouri--Columbia).

Gelenbe, E., Bakircioglu, H., & Kocak, T. (1997, July). Image processing with the random neural network (RNN). In Proceedings of 13th International Conference on Digital Signal Processing (Vol. 1, pp. 243-248). IEEE.

Gers, F. (2001). Long short-term memory in recurrent neural networks (Doctoral dissertation, Verlag nicht ermittelbar).

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10), 2451–2471.

Ghazali, R., Hussain, A. J., Liatsis, P., & Tawfik, H. (2008). The application of ridge polynomial neural network to multi-step ahead financial time series prediction. Neural Computing and Applications, 17(3), 311-323.

Gomes, G. S. D. S., & Ludermir, T. B. (2008, September). Complementary log-log and probit: activation functions implemented in artificial neural networks. In 2008 Eighth International Conference on Hybrid Intelligent Systems (pp. 939-942). IEEE.

Goodwin, P. (2014). Using naïve forecasts to assess limits to forecast accuracy and the quality of fit of forecasts to time series data. Available at SSRN 2515072.

Grabczewski, K., & Jankowski, N. (2005, November). Feature selection with decision tree criterion. In Fifth International Conference on Hybrid Intelligent Systems (HIS'05) (pp. 6-pp). IEEE.

Graves, A., Fernández, S., & Schmidhuber, J. (2005, September). Bidirectional LSTM networks for improved phoneme classification and recognition. In International Conference on Artificial Neural Networks (pp. 799-804). Springer, Berlin, Heidelberg.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Habibullah, M. (2017). A study on currencies – gold - crude and trade balance. International Journal of Management and Applied Science, 3(7), pp. 2394-7926

Hall-Smith, W. (2018). The top 10 most traded currencies in the world. IG. Retieved from https://www.ig.com/au/trading-strategies/the-top-ten-most-traded-currencies-in-the-world-180904

Hassan, K., & Salim, R. A. (2011). Is there any link between commodity price and monetary policy? Evidence from Australia. Economic Analysis and Policy, 41(3), 205-215.

Hayes, A. (2019). Stochastic Oscillator Definition. Investopedia. Retrieved from https://www.investopedia.com/terms/s/stochasticoscillator.asp.

Hayward, R. (2018). Foreign Exchange Speculation: An Event Study. International Journal of Financial Studies, 6(1), p.22.

He, K., Xie, C., & Lai, K. K. (2008, October). Multi scale nonlinear ensemble model for foreign exchange rate prediction. In 2008 Fourth International Conference on Natural Computation (Vol. 7, pp. 43-47). IEEE.

Heryadi, Y., & Warnars, H. L. H. S. (2017, November). Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM. In 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom) (pp. 84-89). IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), pp.1735-1780.

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Oxford, England: U Michigan Press.

Hollander, B. (2018). Natural vs Artificial Neural Networks. Retrieved from https://becominghuman.ai/natural-vs-artificial-neural-networks-9f3be2d45fdb

Hsu, W., Hsu, L. S., & Tenorio, M. F. (1995). A neural network procedure for selecting predictive indicators in currency trading. In Neural networks in the capital markets (pp. 245-257). John Wiley & Sons Redwood City.

Huang, S. C., & Wu, T. K. (2008). Integrating GA-based time-scale feature extractions with SVMs for stock index forecasting. Expert Systems with Applications, 35(4), 2080-2088.

Huang, Y. (2016). Forecasting the USD/CNY Exchange Rate under Different Policy Regimes (No. 2016-001).

Hyrina, Y., & Serletis, A. (2010). Purchasing power parity over a century. Journal of Economic Studies, 37(1), 117-144.

IMF. (2018). "Special drawing right (SDR) - factsheet". www.imf.org. International monetary fund. Retrieved 25 February 2019.

Jiang, F., & Wu, W. (2016, August). Hybrid Genetic Algorithm and Support Vector Regression Performance in CNY Exchange Rate Prediction. In 2016 International Conference on Engineering Science and Management. Atlantis Press.

Johnston, F. R., Boyland, J. E., Meadows, M., & Shale, E. (1999). Some properties of a simple moving average when applied to forecasting a time series. Journal of the Operational Research Society, 50(12), 1267-1271.

Kaltwasser, P. R. (2010). Uncertainty about fundamentals and herding behavior in the FOREX market. Physica A: Statistical Mechanics and its Applications, 389(6), 1215-1222.

Kamruzzaman, J., & Sarker, R. A. (2003, December). Forecasting of currency exchange rates using ANN: A case study. In International Conference on Neural

Networks and Signal Processing, 2003. Proceedings of the 2003 (Vol. 1, pp. 793-797). IEEE.

Kamruzzaman, J., Sarker, R. A., & Ahmad, I. (2003, November). SVM based models for predicting foreign currency exchange rates. In Third IEEE International Conference on Data Mining (pp. 557-560). IEEE.

Kazem, A., Sharifi, E., Hussain, F. K., Hussain, O. K., & Saberi, M. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Applied Soft Computing Journal, 13(2), 947-958.

Khan, K., Su, C., Tao, R. and Chu, C. (2018). Is there any relationship between producer price index and consumer price index in the Czech Republic? Economic Research-Ekonomska Istraživanja, 31(1), pp.1788-1806.

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Systems with Applications, 103, 25-37.

Kim, K.J.; Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst. Appl., 19, 125–132.

King, M. R., Osler, C. L., & Rime, D. (2011). Foreign exchange market structure, players and evolution.

Koizumi, T., Mori, M., Taniguchi, S., & Maruya, M. (1996, October). Recurrent neural networks for phoneme recognition. In Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96 (Vol. 1, pp. 326-329). IEEE.

Kondratenko, V. V., & Kuperin, Y. A. (2003). Using recurrent neural networks to forecasting of forex. arXiv preprint cond-mat/0304469.

Korczak, J., Hernes, M., & Bac, M. (2016, September). Fundamental analysis in the multi-agent trading system. In 2016 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 1169-1174). IEEE.

Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014). Neural network ensemble operators for time series forecasting. Expert Systems with Applications, 41(9), 4235-4244.

Krušković, B. D., & Maričić, T. (2015). Empirical Analysis of the impact of foreign exchange reserves to economic growth in emerging economics. Applied economics and finance, 2(1), 102-109.

Kuan, C. M., & Liu, T. (1995). Forecasting exchange rates using feedforward and recurrent neural networks. Journal of applied econometrics, 10(4), 347-364.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling (Vol. 26). New York: Springer. Page 70

Kumar, J., Goomer, R., & Singh, A. K. (2018). Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters. Procedia Computer Science, 125, 676-682.

Lai, K. K., & Nakamori, Y. (2003). An empirical analysis of sampling interval for exchange rate forecasting with neural networks. Journal of Systems Science and Complexity, (2), 2.

Lee, Y., Tiong, L. C. O., & Ngo, D. C. L. (2014, May). Hidden markov models for forex trends prediction. In 2014 International Conference on Information Science & Applications (ICISA) (pp. 1-4). IEEE.

Leung, M. T., Chen, A. S., & Daouk, H. (2000). Forecasting exchange rates using general regression neural networks. Computers & Operations Research, 27(11-12), 1093-1110.

Levich, R. M. (2011). Evidence on financial globalization and crises: Interest rate parity.

Li, W. (1991). Absence of 1/f spectra in Dow Jones daily average. International Journal of Bifurcation and Chaos, 01(03), pp.583-597.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

Liu, F. Y. (2010, March). The hybrid prediction model of CNY/USD exchange rate based on wavelet and support vector regression. In 2010 2nd International Conference on Advanced Computer Control (Vol. 4, pp. 561-565). IEEE.

Liu, Z., Zheng, Z., Liu, X., & Wang, G. (2009, July). Modelling and Prediction of the CNY Exchange Rate Using RBF Neural Network. In 2009 International Conference on Business Intelligence and Financial Engineering (pp. 38-41). IEEE.

López-Rubio, E., Ortega-Zamorano, F., Domínguez, E., & Muñoz-Pérez, J. (2019). Piecewise Polynomial Activation Functions for Feedforward Neural Networks. Neural Processing Letters, 1-27.

Lui, Y. H., & Mole, D. (1998). The use of fundamental and technical analyses by foreign exchange dealers: Hong Kong evidence. Journal of International Money and Finance, 17(3), 535-545.

Ma, L., & Khorasani, K. (2001, November). Constructive Hermite polynomial feedforward neural networks with application to facial expression recognition. In Video Technologies for Multimedia Applications (Vol. 4520, pp. 31-43). International Society for Optics and Photonics.

Ma, L., & Khorasani, K. (2003). A new strategy for adaptively constructing multilayer feedforward neural networks. Neurocomputing, 51, 361-385.

Maknickien, N., & Maknickas, A. (2012). Application of neural network for forecasting of exchange rates and forex trading. In The 7th international scientific conference" Business and Management (pp. 10-11).

Malik, V., & Kumar, A. (2018). Analysis of Twitter Data Using Deep Learning Approach: LSTM. International Journal on Recent and Innovation Trends in Computing and Communication, 6(4), 144-149.

Mańdziuk, J., & Rajkiewicz, P. (2016, July). Neuro-evolutionary system for FOREX trading. In 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 4654-4661). IEEE.

Marreiros, A. C., Daunizeau, J., Kiebel, S. J., & Friston, K. J. (2008). Population dynamics: variance and the sigmoid activation function. Neuroimage, 42(1), 147-157.

Masry, M. (2017). The Impact of Technical Analysis on Stock Returns in an Emerging Capital Markets (ECM's) Country: Theoretical and Empirical Study. International Journal of Economics and Finance, 9(3), p.91.

Mayeda, Andrew. (2015). IMF Approves Reserve-Currency Status for China's Yuan. Bloomberg Retrieved from https://www.bloomberg.com/news/articles/2015-11-30/imf-backs-yuan-in-reserve-currency-club-after-rejection-in-2010 on 25th February, 2019.

McNally, S., Roche, J., & Caton, S. (2018, March). Predicting the price of Bitcoin using Machine Learning. In 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP) (pp. 339-343). IEEE.

Meese, R. A., & Rogoff, K. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? Journal of International Economics, 14(1), 3-24.

Messina, R., & Louradour, J. (2015, August). Segmentation-free handwritten Chinese text recognition with LSTM-RNN. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 171-175). IEEE.

Mishra, A., Chandra, P., Ghose, U., & Sodhi, S. S. (2017). Bi-modal derivative adaptive activation function sigmoidal feedforward artificial neural networks. Applied Soft Computing Journal, 61, 983-994.

Mitchell, C. (2019). Average Directional Index-ADX Definition and Uses. Investopedia. Retrieved from https://www.investopedia.com/terms/a/adx.asp

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.

Moore, M., Schrimpf, A., & Sushko, V. (2016). Downsized FX markets: causes and implications.

Nanayakkara, K. A. D. S. A., Chandrasekara, N. V., & Jayasundara, D. D. M. (2014). Forecasting exchange rates using time series and neural network approaches. European International Journal of Science and Technology, 3.

Narayan, A., & Roe, P. H. N. (2018). Learning Graph Dynamics using Deep Neural Networks. IFAC-PapersOnLine, 51(2), 433-438.

Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. C. L. (2015). Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. Expert Systems with Applications, 42(1), 306-324.

Nassirtoussi, A. K., Wah, T. Y., & Ling, D. N. C. (2011). A novel FOREX prediction methodology based on fundamental data. African Journal of Business Management, 5(20), 8322.

Nau, R. (2014). Forecasting with moving averages. Fuqua School of Business, Duke University, 1-3.

Navarin, N., Vincenzi, B., Polato, M., & Sperduti, A. (2017, November). LSTM networks for data-aware remaining time prediction of business process instances. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-7). IEEE.

Neely, C., Weller, P., & Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. Journal of financial and Quantitative Analysis, 32(4), 405-426.

Nugrahani, T. A., Adi, K., & Suseno, J. E. (2018). Information System Prediction With Weighted Moving Average (WMA) Method And Optimization Distribution Using Vehicles Routing Problem (VRP) Model for Batik Product. In E3S Web of Conferences (Vol. 73, p. 13004). EDP Sciences.

Nwiado, D., & LeneeTorbira, L. (2016). A Panel Data Analysis of the Validity of Uncovered Interest Rate Parity (UIRP) in Selected African Countries. Research in World Economy, 7(2), 15-25.

Olah, C. (2015). Understanding LSTM Networks. colahs' blog. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 24(4), 694-707.

Pang, X. W., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2018). Stock Market Prediction based on Deep Long Short Term Memory Neural Network. In COMPLEXIS (pp. 102-108).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

Philip, A. A., Taofiki, A. A., & Bidemi, A. A. (2011). Artificial neural network model for forecasting foreign exchange rate. World of Computer Science and Information Technology Journal, 1(3), 110-118.

Pincak, R. (2013). The string prediction models as invariants of time series in the forex market. Physica A: Statistical Mechanics and its Applications, 392(24), 6414-6426.

Prechelt, L. (1998). Early stopping-but when?. In Neural Networks: Tricks of the trade (pp. 55-69). Springer, Berlin, Heidelberg.

Pring, M. J. (1991). Technical Analysis Explained, Second Edition, New York: McGraw-Hill (1991).

Pujari, M. V., Sayyed, A. H., Shahani, H., Rupani, D., & Student, B. E. (2018). Forex Trading System. International Journal of Engineering Science, 17116.

Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

Rahman, L., Setiawan, N. A., & Permanasari, A. E. (2017, November). Feature selection methods in improving accuracy of classifying students' academic performance. In 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 267-271). IEEE.

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 42(6), 3234-3241.

Rawal, A., & Miikkulainen, R. (2016, July). Evolving deep lstm-based memory networks using an information maximization objective. In Proceedings of the Genetic and Evolutionary Computation Conference 2016 (pp. 501-508). ACM.

Refenes, A. N., Zapranis, A., & Francis, G. (1994). Stock performance modeling using neural networks: a comparative study with regression models. Neural networks, 7(2), 375-388.

Rehman, M., Khan, G. and Mahmud, S. (2014). Foreign Currency Exchange Rates Prediction Using Recurrent Neural Network. IERI Procedia, 10, pp.239-244.

Ridzuan, H. F., Ja'afar, A., Amali, A. K., Syam, A. J. S. H. F., & Yulni, J. (2019). MLP Based Tan-Sigmoid Activation Function for Cardiac Activity Monitoring. In MATEC Web of Conferences (Vol. 255, p. 03005). EDP Sciences.

Rigatos, G. G., & Tzafestas, S. G. (2006, May). Feed-forward neural networks using Hermite polynomial activation functions. In Hellenic Conference on Artificial Intelligence (pp. 323-333). Springer, Berlin, Heidelberg.

Rime, D. (2003). New electronic trading systems in foreign exchange markets. New Economy Handbook, 469504.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

Satchell, S., & Knight, J. (2011). Forecasting volatility in the financial markets. Elsevier.

Schaefer, A. M., Udluft, S., & Zimmermann, H. G. (2008). Learning long-term dependencies with recurrent neural networks. Neurocomputing, 71(13-15), 2481-2488.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Seifter, J., Sloane, D., & Ratner, A. (2005). Concepts in medical physiology. Lippincott Williams & Wilkins.

Sharma, A. (2017). Understanding Activation Functions in Neural Networks. Towards Data Science. Retieved from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Sharma, S. (2017). Activation Functions in Neural Networks. Towards Data Science. Retieved from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Sidehabi, S. W., & Tandungan, S. (2016). Statistical and Machine Learning approach in forex prediction based on empirical data. In Computational Intelligence and Cybernetics (CYBERNETICSCOM), 2016 International Conference on (pp. 63-68). IEEE.

Sodhi, S. S., & Chandra, P. (2014). Bi-modal derivative activation function for sigmoidal feedforward networks. Neurocomputing, 143, 182-196.

Sønderby, S. K., & Winther, O. (2014). Protein secondary structure prediction with long short term memory networks. arXiv preprint arXiv:1412.7828.

Soulas, E., & Shasha, D. (2013). Online machine learning algorithms for currency exchange prediction. Computer Science Department in New York University, Tech. Rep, 31.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In Thirteenth annual conference of the international speech communication association.

Swarnkar, S. K. (2018). Neural Networks (Introduction & Architecture). Computational Science With Suman. Retrieved from http://computationalsciencewithsuman.blogspot.com/p/neural-networks-introduction.html

Takeuchi, L., & Lee, Y. Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In Technical Report. Stanford University.

Talebi, H., Hoang, W., & Gavrilova, M. L. (2014). Multi-scale foreign exchange rates ensemble for classification of trends in forex market. Procedia Computer Science, 29, 2065-2075.

Tay, D. (2017). Time series analysis of discourse: A case study of metaphor in psychotherapy sessions. Discourse Studies, 19(6), pp.694-710.

Taylor, M. P., & Allen, H. (1992). The use of technical analysis in the foreign exchange market. Journal of international Money and Finance, 11(3), 304-314.

Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. Applied Artificial Intelligence, 10(6), 567-582.

Thomason, M. (1999). The practitioner method and tools. Journal of Computational Intelligence in Finance. 7(3):36–45

Tlegenova, D. (2015). Forecasting Exchange Rates Using Time Series Analysis: The sample of the currency of Kazakhstan. arXiv preprint arXiv:1508.07534.

Triennial Central Bank. (2016). Triennial Central Bank Survey: Foreign exchange turnover in April 2016. Triennial Central Bank Survey. Basel, Switzerland: Bank for International Settlements. Retrieved from https://www.bis.org/publ/rpfx16fx.pdf.

Tseng, F. M., Yu, H. C., & Tzeng, G. H. (2002). Combining neural network model with seasonal time series ARIMA model. Technological forecasting and social change, 69(1), 71-87.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M. and Baik, S. (2018). Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. IEEE Access, 6, pp.1155-1166.

Ursell, M. (2015). How to Calculate the PSAR Using Excel - Revised Version [Video file]. Retrieved from https://www.youtube.com/watch?v=MuEpGBAH7pw

Velicer, W. F., & Fava, J. L. (2003). Time series analysis. Research methods in psychology, 2.

Venkateswarlu, G., & Sarma, A. D. (2017, February). Performance of Holt-Winter and exponential smoothing methods for forecasting ionospheric TEC using IRNSS data. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-5). IEEE.

Wagner, H. (2018). Why Volatility is Important For Investors. Investopedia. Retieved from https://www.investopedia.com/articles/financial-theory/08/volatility.asp.

Walia, A.S. (2017). What are Artificial Neural Networks ?. Towards Data Science. Retieved from https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

Wang, G. J., & Xie, C. (2013). Cross-correlations between Renminbi and four major currencies in the Renminbi currency basket. Physica A: Statistical Mechanics and its Applications, 392(6), 1418-1428.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2285-2294).

Wang, Y., Velswamy, K. and Huang, B. (2017). A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems. Processes, 5(4), p.46.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. Neural computation, 1(4), 425-464.

Wielgosz, M., Skoczeń, A. and Mertik, M. (2017). Using LSTM recurrent neural networks for monitoring the LHC superconducting magnets. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 867, pp.40-50.

Wilcox, E., Levy, R., Morita, T., & Futrell, R. (2018). What do RNN Language Models Learn about Filler-Gap Dependencies?. arXiv preprint arXiv:1809.00042.

Wilder, J. W. (1978). New concepts in technical trading systems. Trend Research.

Williamson, J. (2009). Understanding Special Drawing Rights (SDRs) (No. PB09-11). Washington, DC: Peterson Institute for International Economics.

Wöllmer, M., Metallinou, A., Eyben, F., Schuller, B., & Narayanan, S. (2010). Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In Proc. INTERSPEECH 2010, Makuhari, Japan (pp. 2362-2365).

Wu, Z., & King, S. (2016, March). Investigating gated recurrent networks for speech synthesis. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5140-5144). IEEE.

Wu, Z., Watts, O., & King, S. (2016, September). Merlin: An Open Source Neural Network Speech Synthesis System. In SSW (pp. 202-207).

Xu, G., & Xia, L. (2018, October). Short-Term Prediction of Wind Power Based on Adaptive LSTM. In 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) (pp. 1-5). IEEE.

Yao, J., & Tan, C. L. (2000). A case study on using neural networks to perform technical forecasting of forex. Neurocomputing, 34(1-4), 79-98.

Yazdi, S. H. M., & Lashkari, Z. H. (2013). Technical analysis of Forex by MACD Indicator. International Journal of Humanities and Management Sciences (IJHMS), 1(2), 159-165.

Ye, F., Zhang, L., Zhang, D., Fujita, H., & Gong, Z. (2016). A novel forecasting method based on multi-order fuzzy time series and technical analysis. Information Sciences, 367, 41-57.

Yin, D., & Chen, W. (2016, July). The forecast of USD/CNY exchange rate based on the Elman neural network with volatility updating. In 2016 35th Chinese Control Conference (CCC) (pp. 9562-9566). IEEE.

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923.

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.

Yu, L., Wang, S., & Lai, K. K. (2005). A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. Computers & Operations Research, 32(10), 2523-2541.

Yu, L., Wang, S., & Lai, K. K. (2010). Foreign-exchange-rate forecasting with artificial neural networks (Vol. 107). Springer Science & Business Media.

Zazo, R., Lozano-Diez, A., Gonzalez-Dominguez, J., Toledano, D. T., & Gonzalez-Rodriguez, J. (2016). Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks. PloS one, 11(1), e0146917.

Zen, H., & Sak, H. (2015, April). Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4470-4474). IEEE.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.

Zhang, G. P., & Berardi, V. L. (2001). Time series forecasting with neural network ensembles: an application for exchange rate prediction. Journal of the Operational Research Society, 52(6), 652-664.

Zhang, H; Zhao, B & Yan, F. (2019). Trusted Computing and Information Security: 12th Chinese Conference, CTCIS 2018, Wuhan, China, October 18, 2018, Revised Selected Papers. Springer.

Zhang, J., Lin, Y., Song, Z., & Dhillon, I. S. (2018). Learning long term dependencies via fourier recurrent units. arXiv preprint arXiv:1803.06585.