Pairs trading on different portfolios based on machine learning

Victor Chang[1], Xiaowen Man[2], Qianwen Xu[1, 3] and Robert Hsu[4, 5, 6]

1. Artificial Intelligence and Information Systems Research Group, School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK
2. IBSS, Xi'an Jiaotong-Liverpool University, Suzhou, China
3. Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan
4. Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan
5. School of Mathematics and Big Data, Foshan University, Foshana 528000, China

Email: ic.victor.chang@gmail.com/V.Chang@tees.ac.uk; mxw960619@gmail.com; iamarielxu@163.com; robertchh@gmail.com

ABSTRACT

This paper presents an advanced visualization and analytics approach for financial research. Statistical arbitrage, particularly pairs trading strategy, has gained ground in the financial market and machine learning techniques are applied to the finance field. The cointegration approach and Long short-term memory (LSTM) were utilized to achieve stock pairs identification and price prediction purposes, respectively, in this project. This article focused on the US stock market, investigating the performance of pairs trading on different types of portfolios (aggressive and defensive portfolio) and compare the accuracy of price prediction based on LSTM. It can be briefly concluded that LSTM offers higher prediction precision on aggressive stocks and implementing pairs trading on the defensive portfolio would gain higher profitability during a specific period between 2016 and 2017. However, predicting tools like LSTM only offer limited advice on stock movement and should be cautiously utilized. We conclude that analytics and visualization can be effective for financial analysis, forecasting and investment strategy.

Keywords: pairs trading; cointegration; long short-term memory; stock price prediction

1. Introduction

1.1 Analytics and visualization Statistical arbitrage and pairs trading

Analytics are referred to as the techniques or software tools that allow complex information processing to be analyzed and presented in ways that users can understand meanings more easily (Keim et al., 2008). They are often in graphical formats to make user interfaces friendly. Visualization is referred to advanced techniques or software packages to comprehend complex data, make sense of them, and present the outputs in graphical forms (Heer et al., 2005). The final outputs can be harder to be interpreted and may support real-time research. Examples may include financial markets to display changes every second, or weather studies to show the live changes due to weather conditions. Analytics and visualization play essential roles for financial analysis, investment strategies and forecasting. In this paper, we demonstrate our approach using the long short-term memory (LSTM) and blend with analytics and visualization to illustrate how to achieve financial analysis, investment strategies and forecasting. We also explain how to maintain the high quality of our work.

1.2 Statistical arbitrage and pairs trading

Statistical arbitrage refers to the trading strategy using quantitative or statistical models to seek trading signals and this type of strategy almost generates certain profits (Göncü and Akyildirim, 2016a). It is widely exercised by various financial institutions to exploit the profits which stem from mispriced assets (Rad, Low and Faff, 2016). In this report, pairs trading, one of the most commonly used statistical arbitrage techniques, will be investigated with a machine learning algorithm (long short-term memory).

Pairs trading, the first-generation of statistical arbitrage, exploits profits of the financial markets, which are non-equilibrium (Göncü and Akyildirim, 2016a). Under the theory of pairs trading, it is perceived that the markets will move to a rational equilibrium gradually if the markets are out of equilibrium (Göncü and Akyildirim, 2016a). In other words, any abnormal price deviation is transitory, and the prices will move back to a reasonable level over time (Liew and Wu, 2013).

Pairs trading strategy aims to exploit short-term deviations under the premise that two assets share a long-term equilibrium pricing relationship. Due to a successful

implementation in the 1980s, pairs trading became a prevalent arbitrage strategy and has been widely acknowledged (Gatev, Goetzmann and Rouwenhorst, 2006; Zeng and Lee, 2014). It is a 'market neutral' strategy because the investors can obtain returns regardless of the market conditions (Huang et al., 2018). Theoretically, identifying potential pairs is not difficult. For instance, the stocks of Coca Cola and Pepsi showed historically high correlation for a long period as they confronted similar business activities, market conditions and risks. For various reasons, the special relationship between the two stocks may be weakened during a short period. The stock price spread may deviate from the historically long-term equilibrium level (Göncü and Akyildirim, 2016b).

Figure 1 presents a simplified process of pairs trading. In this example, the first task is to identify two stocks (a pair) whose prices can move together. We can then calculate the spread of two stock prices and the spread will be normalized (the blue curve in the figure). The black line in the middle is the mean of the normalized spread for the period (here, 250 the days is the length of the period). If the spread departures from the normal range (mean $\pm$ standard deviation, namely the space between the red and the green lines), the trading period will start. During the trading period, the overvalued stock will be short and the undervalued one will belong at the same time. The trading period ends when the spread reverts to the normal range (long-term equilibrium level) and the profits are available at the end of the trading period.

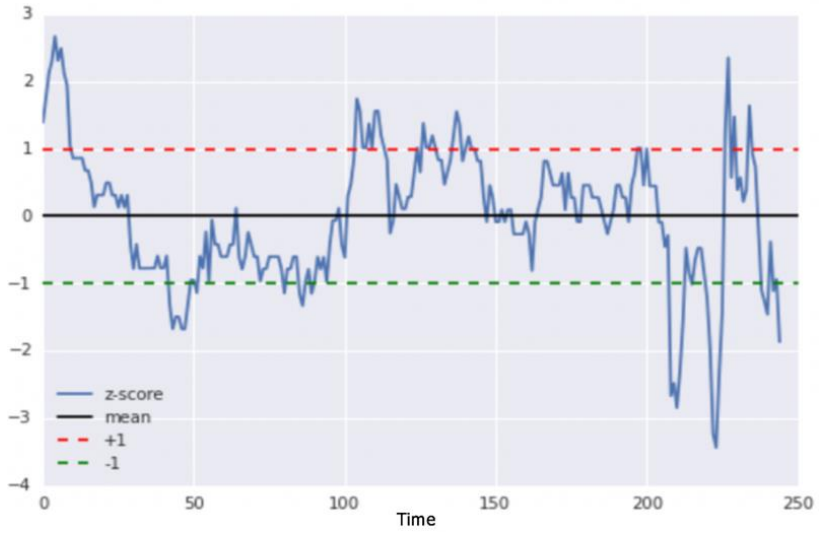

Figure 1. Price spread movement

1.3 Investigation on pairs trading from the literature-based perspective

Since the emergence of pairs trading, Gatev, Goetzmann and Rouwenhorst (2006) are the pioneers who comprehensively investigated the application of pairs trading into the US equity market. Based on this seminal paper, extensive kinds of literature concerning the pairs trading domain emerged and offered insights about this field from diverse perspectives. To illustrate, scholars created various approaches to identify potential pairs, including the Distance method, the Cointegration method, the stochastic control method and other approaches (Krauss, 2017). The frequency of trading opportunities varies while using different methods to find pairs (Göncü and Akyildirim, 2016b), and some methods present declining profitability because more attention is paid to this domain. Computer-based approaches like machine learning also gain ground and afford new insights into the pairs trading field.

1.4 Research objectives

Based on the previous papers, two research aims are proposed: when employing a machine learning technique (LSTM) to forecast price movement for the aggressive and defensive portfolio, respectively, the prediction accuracy on which portfolio presents a better performance?

In terms of cumulative return, which portfolio can outperform the other by trading predicted prices to trade?

1.5 Methodology and description of the procedure for this research

Five main stages are incorporated in this project: (1) categorizing stocks into different types (aggressive or defensive types), (2) identifying pairs with the cointegration test, (3) constructing portfolio, (4) forecasting stock prices by machine learning algorithm (LSTM) and (5) calculating trading profits. Figure 2 shows the general process of this research. They can be explained as follows.

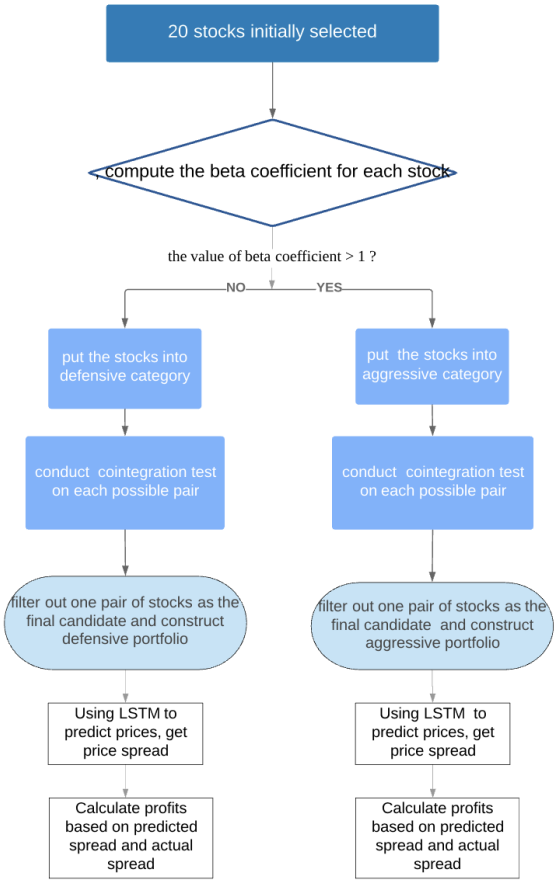

Figure 2. The general process of this research

Twenty stocks across diverse industries were selected initially to filter out the stock candidates for further steps. Among these twenty stocks, ten stocks with a beta coefficient greater than 1 are from the industries which are perceived as potential 'cyclical' sectors. The other ten stocks with a beta coefficient of less than 1 are from the potential 'non-cyclical' sectors.

The next task is to conduct a stationary test then to find the cointegrated pairs from the aggressive and defensive stock list, respectively. The one pair (ORCL and CEA) from the aggressive type constitutes the aggressive portfolio and another pair (JNJ and NEE) containing two defensive stocks is regarded as the aggressive portfolio. The essential method employed in this step to identify the cointegration relationship is the Augmented Engle-Granger two-step cointegration test.

Based on the above preparations, a machine learning algorithm will act the main role during the rest process: predicting stock prices. In this step, sub-tasks, including raw data normalization and transformation, have to be completed first. The processed data can

then be fed to the LSTM algorithm, which can identify the underlying patterns existing in data and make predictions. More specifically, the adjusted close prices of 2008.1.2 ~ 2015.12.31 (eight years, non-trading days are excluded) will be used as a training dataset to forecast the prices for 2016.1.4 ~ 2017.12.29.

After obtaining the series of predicted stock prices of the four stocks, two series of price spread (ORCL-CEA, JNJ-NEE) can be derived based on the predicted prices. Then the trading and profit calculation sections commence. During the trading period, the normal range of the price spread is determined by investors' preference. In this report, we perceive that the trading will open when the price spread walks beyond the normal range (mean $\pm$ 1.5 standard deviations), and when the spread comes back to space between mean $\pm$ 1.5 standard deviations, the trading period ends. Here, the computation of the mean and standard deviations is based on the adjusted close price spanning 2008.1.2 to 2015.12.31.

The article is organized as follows. Related papers that shed light on pairs trading and the research combining machine learning techniques or other approaches will be reviewed systematically in the next section. Section 3 elaborates on the methods adopted in this research and section 4 presents the numerical findings. Discussion of the results, limitations, suggestions for future work is positioned in section 5. Section 6 concludes the outcomes of research outputs.

2. Literature review

Extensive literature investigated pairs trading fields from financial, statistical, computer-based and other different perspectives. In this part, the classical and conventional approaches on pairs trading are provided firstly; next, the paper which applied machine learning algorithms into pairs trading is also reviewed. Finally, the long short-term memory algorithm is introduced and the reasons this research selected this method are explained.

2.1 Classical and conventional methods concerning pairs trading

Presently, four mainstream approaches are well-recognized in pairs trading, including the cointegration theory, the stochastic spread, the minimum sum of the distance squared and the copula strategy.

Cointegration theory was operated by Engle and Granger (1987) and it is based on the error correction model. Based on Engle and Granger's theory, Vidyamurthy (2004) attempts to use parameterized trading rules for pairs trading by using cointegration relationships between assets. The distance approach, provided by Gatev et al. (2006), is used to construct pairs by identifying a matching partner who can minimize the sum of Euclidean squared distance (SSD) between the two normalized price series. The stochastic spread strategy proposed by Elliott et al. (2005) assumes that if the spread between two stocks is mean-reverting, the spread is expected to stay mean-reverting for some time in the future. Liew and Wu (2013) offered insights about the application of copula into pairs trading, and they observed that this approach afforded more trading opportunities and no rigid assumptions were needed.

The cointegration theory is widely utilized for stock selection purposes because it demonstrates the trend of price and precisely measures the extent to which the price spread departs from the long-term equilibrium (Wen et al., 2018). Several studies have confirmed the importance of the cointegration theory. Huck and Afawubo (2015) evaluated the distance approach and cointegration by studying the S&P 500 index components. Results showed that after considering the risk and transaction costs, the return was insignificant when using the distance approach while it was stable and significant when using the cointegration theory. The performance of the cointegration theory was also proved by Rad et al. (2016). In order to make their research more robust, they took the daily data for all US stocks spanning from July 1962 to December 2014. An empirical analysis was conducted to compare the performances (with time-varying trading costs) among the three strategies: the distance, the cointegration and copula approaches. They uncovered that all methods presented better performance during significant volatility periods, and the cointegration method outperformed the other two methods when the market was exposed to turbulent circumstances.

2.2 Machine learning techniques for stock prediction

With artificial intelligence development, more and more machine learning algorithms have been introduced to many industries, including the financial market. The algorithms improve the efficiency of trading and are helpful in monitoring the market trend. Many

machine learning algorithms have recently been used to learn the finance data, including the pairs trading.

2.2.1 Machine learning approaches for pairs trading

Chaudhuri et al. (2017) narrowed their investigative scope to the Indian stock market and focused on predicting the ratio of the prices rather than the spread of stock prices. In their pairs trading research, a period between 2012 and 2015, they tracked three pairs from the same sector, and three different machine learning algorithms, namely SVR, random forest (RF), Adaptive-Neuro Fuzzy Inference System (ANFIS) were used for predictive modeling. They extracted nine independent variables (features) as indicators to forecast the price ratio. Their framework combined technical indicators with the mean-reverting characteristic of the movement of pairs price. Instead of using the distance method, they chose to utilize Mean Squared Error and Mean Absolute Percentage Error to evaluate the predictive ability of different algorithms. As a result, all the algorithms they adopted effectively predicted the ratio of the share price of pairs.

Chen et al. (2018) focused on the application of Convention Neural Network (CNN) into pairs trading in Taiwan Stock Index Futures. They improved CNN by combining financial knowledge and filterbank mechanisms and then proposed a Filterbank CNN framework, which enhanced prediction accuracy and profitability since this framework could successfully capture arbitrage signals. A superior pairs trading system could be generated by combining integrated information technology and financial domain knowledge based on their research.

Huck (2009) and Huck (2010) combined the machine learning and forecast approach altogether. The methodology comprised three steps, forecasting, outranking and trading. Huck (2009) utilized Elman neural networks to predict one-week returns for each security in the first stage. A multi-criteria decision method (MCDM) called ELECTRE III was applied to create an "outranking" system, where the undervalued stocks were located at the top and the overvalued stocks were at the bottom. In the trading stage, stocks at the top of the ranking were bought while the ones at the bottom were sold short. The positions were closed after the trading period opened one week, then a new rank would be generated, and the above process repeated (Krauss, 2017).

In Nóbrega and Oliveira (2014) research, Kalman Filter Regression was deployed to combine Extreme Learning Machine and SVR, which aimed to forecast the price spread deviation. They firstly filtered a set of pairs having the cointegration and stationary properties after conducting ADF and the Phillips-Ouliaris tests. Then the Ornstein-Uhlenbeck process was used for estimating mean reversion's speed. Before starting the machine learning process, they applied the feature selection method to find the best subsets fed in the machine learning algorithm.

2.2.2 Machine learning techniques for other areas of the finance market

Several studies have been conducted on the stock index prediction. Support Vector Machines (SVM) was applied in the research of Najafabadi (2009) to the problem of prediction in the Canadian stock market and of Huang et al. (2018) to predict the NIKKEI 225 index's weekly direction. The result illustrated that SVM showed better performance than Linear Discriminant Analysis (LDA) and Elman's Backpropagation Neural Network (EBPNN). Usmani et al. (2016)'s study on the Karachi Stock Exchange showed that one machine learning technique, the Multi-Layer Perceptron algorithm, provided 77% prediction accuracy of the market performance. Dunis (2012) examined the performance of SVM when it was employed to predict the weekly change in the Madrid IBEX-35 stock index during the period of 10/18/1990 ~ 10/29/2010. SVM was used for data analysis and pattern recognition (prediction) purposes. According to the paper, SVM performed well if the training period was shorter.

Garg (2012) integrated GARCH and machine learning models (Regression Frees, Random Forests (RF), Support Vector Regression (SVR), Least Absolute Shrinkage and Selection Operator (LASSO)) and created an original framework to forecast exchange rates (EUR/SEK, EUR/USD and USD/SEK). It proved that SVM showed superior prediction performance than recurrent neuron network (RNN).

Santoso et al. (2018) make predictions on the price of Astra International (stock code: ASII.JK) stock data. Stock prices by introducing stock price features into SVM. In their study, the Gaussian Mixture Model (GMM) was employed to decompose the stock price

series, which made predictions more precise. It turned out that this integrated framework comprising GMM and SVM offered significant cumulative returns.

Referred to Novak and Veluscek (2016), daily high prices of stocks presented lower volatility compared with the popular daily stock closed prices. Based on this premise, they used SVM and Linear discriminant analysis to forecast whether the daily high price's movement direction (rise or fall). The combined methodology's successful classification rate reached 60%.

Tay and Cao (2003) compared the suitability of SVM and RNN separately and concluded that SVM surpassed the ANN when forecasting the change of bonds and the prices of stock index futures in a prediction horizon of five days, but SVM is prone to be sensitive when the parameters changed.

2.3 Long short-term memory (LSTM)

2.3.1 Background

2.3.1.1 Neural network and artificial neuron

Machine learning tools, neuron networks, in particular, are introduced into the financial area. This is likely because neural networks present decent approximation to almost all nonlinear functions. Due to the nonlinearity of time series data, it is better to use neural networks to predict stock prices, rather than using regular linear frameworks. The merits of neural networks of dealing with time series data: 1) without given any nonlinear relationship, neural networks can detect nonlinearities existing in data; 2) when dealing with new data, neural networks can absorb the new data with no need to process the previously old information again, which remarkably reduces the workload (Have, 2017).

The neural network, a computational structure that performs tasks in a similar way of biological neurons, has gained ground in diverse domains (Wen et al., 2018). A neural network incorporates an input layer, one or several hidden layers and an output layer. Generally, the input layers hold the values and the hidden layers process the input layer's value through certain nonlinear functions then deliver them to the next layers, the output layer. An artificial neuron is referred to as the fundamental processing unit, present in the hidden and output layers. Its simplified structure is depicted below:

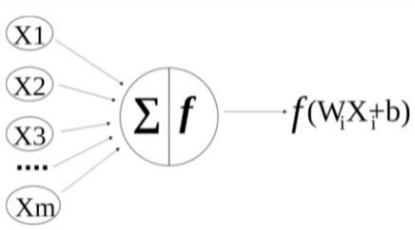

Figure 3. Artificial neuron

The neuron above has m inputs (x_i) connected to neuron by weighted link (w_i), and this neuron derives the output by using equitation:

$$A = \sum x_i * w_i + b$$

b is a threshold value or bias, contingent upon different circumstances; the inputs (x_i) and weights (w_i) are real numbers; an activation function F(A) is responsible for mapping the neuron's output to get the final output.

$$output=F(A)$$

2.3.1.2 Recurrent Neuron Network (RNN)

RNN is a specific neuron network consisting of three layers: the input layer, the hidden layer, and the output layer (Wang et al., 2018). Artificial neurons introduced above are present in the hidden and output layers, and each neuron receives input from a previous layer (Wen et al., 2018).

The hidden layers use a series of nonlinear functions to process the values. Input layers are connected to hidden layers and hidden layers are connected to output layers, and the weights, which decide the importance of the information from a certain node to the receiving node, represent these connections. And the parameters that RNN has to estimate are these weights.

The input sources of RNN stem from two parts: the previous status and the other is current input. This means that hidden layer's value depends on both the input of the current time point and the value of the hidden layer at the previous time point, alternatively, when the

output of the current hidden layer is calculated, the state of the hidden layer of the input layer and the previous time point can be used simultaneously.

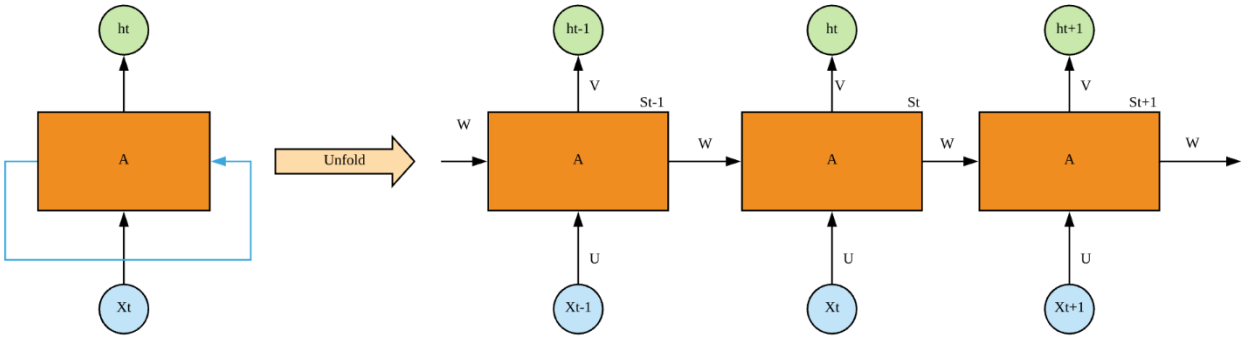

Figure 4. the structure of RNN

More specifically, we can call its output of the hidden layers at time point t-1 when RNN conducts the training process at time point t. This enables RNN to have the ability to remember and recall past information ~~(Have, 2017)~~.

To explicitly explain the mechanism of RNN, we import several variables: x (a vector), s and h represent the values of the input layer and hidden layer, respectively. U and V denote the input layer's weight matrix to the hidden layer and the weight matrix of the hidden layer to the output layer separately; W refers to the weight matrix for the previous time point to the current time point of the hidden layer. The formulas of the RNN is shown as below:

$$h_t = g(Vs_t) \qquad (a)$$

$$s_t = f(UX_t + Ws_{t-1}) \qquad (b)$$

Where g and f are the activation functions.

The above two equations are the calculation formulas for the output layer and the hidden layer separately.

The output layer is a fully connected layer. Each node belonging to the output layers is interconnected to the hidden layer and the hidden layer is a recurrent layer. Recurrent

layers will put the matrix W to the equitation. We can derive the following result by repeatedly introducing equitation (a) into equitation (b):

$$
\begin{aligned}
\mathrm{h}_t &= g(V\mathrm{s}_t) \\
&= gVf(U\mathrm{x}_t + W\mathrm{s}_{t-1}) \\
&= gVf(U\mathrm{x}_t + Wf(U\mathrm{x}_{t-1} + W\mathrm{s}_{t-2})) \\
&= gVf(U\mathrm{x}_t + Wf(U\mathrm{x}_{t-1} + Wf(U\mathrm{x}_{t-2} + W\mathrm{s}_{t-3}))) \\
&= gVf(U\mathrm{x}_t + Wf(U\mathrm{x}_{t-1} \\
&\quad + Wf(U\mathrm{x}_{t-2} + Wf(U\mathrm{x}_{t-3} + \ldots))))
\end{aligned}
$$

2.3.1.3 Forecasting stock prices using long short-term memory network (LSTM)

A special variant of RNN proposed by Hochreiter and Schmidhuber (1997), LSTM, is proficient at overcoming the problems (exploding and gradients vanishing) occurring in the RNN model during the training process. In other words, it means that RNN cannot remember or capture the information of the long-distance and is unable to learn the long-term decencies. LSTM has become a widespread approach in predicting time series. It gets its exceptional predictive ability from the existence of the cell state that allows it to understand and learn longer-term trends in the data. This is especially important for stock price data.

The structure of LSTM differs from that of other neural networks. The essential difference in LSTM is that a cell state exists, C_t, which can store and process the information. Traditional RNN only has a simple feedback loop for the neuron network. However, LSTM possesses a memory block or cell, and three 'gate' (input gate, forget gate and output gate) are designed in each block (or cell) to regulate the data flows, which can make it possible for the network to achieve the goal: evading the long-term dependency problem.

When the information passes through the gates, the information will be selectively added or removed. Sigmoid function, whose value ranges from 0 to 1, is utilized to implement the gate structure, which determines how much information is allowed to pass the gate.

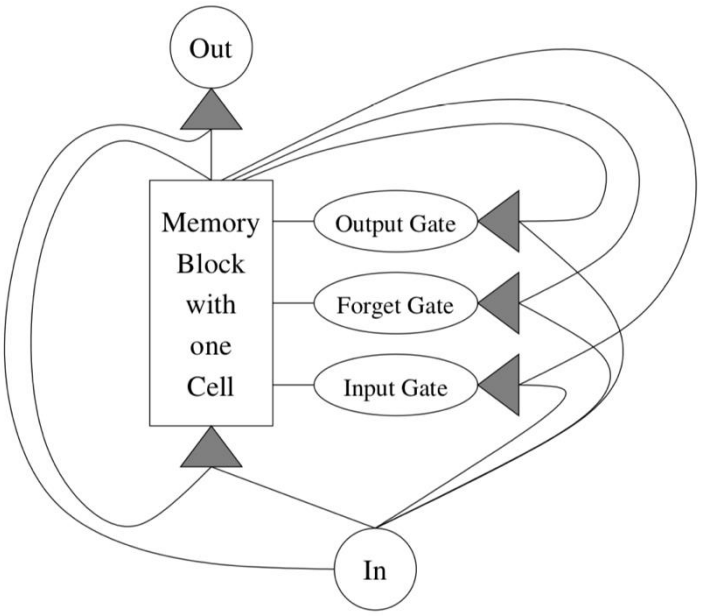

Figure 5.A simplified process of LSTM. In this picture, we can see each gate has two sources of input.

2.3.2 Reasons to choose LSTM

Among the algorithms applied in the above literature, this paper will use long short-term memory algorithms. With artificial intelligence or machine learning development, the smart algorithms shorten the trading time into microseconds, and high-frequency trading was born (HFQ) (Rundo, 2019). Pairs trading in the HFQ provides traders great opportunities to profit. Among the algorithms used in the stock market, LSTM has been proven to be one of the most advanced and successful algorithms.

Fischer and Krauss (2018) utilized Long short-term memory (LSTM) to predict S&P500 index data ranging from December 1992 to October 2015. When conducted financial time series, LSTM outperformed other classification methods which were memory free (no remembering ability), such as random forest (RF), deep neural net (DNN) and logistic regression classifier (LOG), and the strategy adopted LSTM exhibited daily returns of 0.46% and a Sharpe ratio of 5.8 prior to transaction costs. In Hemanth and Basavaraj (2018)'s research on the volatility trend of INR USD currency pair, they employed the LSTM algorithm in the prediction. By comparing with classical regression neural networks, SVM,

RF, regression algorithms, decision trees, and boosting techniques, LSTM achieved the highest accuracy. Ma and Han (2018) applied seven trading strategies established on deep learning algorithms on the financial data from the Shanghai Composite Index and they found that the strategy based on deep neural network performs best.

This research will select LSTM as the algorithm to predict stock price by comparing it with other machine learning algorithms. For pairs trading, long-term (mean-reverting) behavior of time series is essential, so the model deployed should capture this behavior and take this into considerations. Based on this point, LSTM is superior to conduct the predicting process compared to other neuron networks (such as RNN) and other memory free methods, such as random forest (RF), deep neural net (DNN) and logistic regression classifier (LOG), since LSTM are capable of remembering the information that appeared a long time ago. Although SVM is one of the widespread approaches in the prediction model of a stock price, the SVM tends to be affected by the constantly changing market conditions and the length of the trading period (Dunis,2012; Tay and Cao, 2003).

3. Methodology

3.1 Data description and pre-processing

Data description

This project uses daily stock prices (example of data format shows in Table 1) accessed from Yahoo Finance (URL: https://finance.yahoo.com). 'Date', 'Open', 'Low', 'Close', 'Adj Close' and 'Volume' are the six features which describe data from different perspectives. The report concentrates on the period between 2008 and 2017 (the year 2008 and 2017 are included, but non-trading days are excluded) and 2519 officially trading days are considered.

AAPL

Date	Open	High	Low	Close	Adj Close	Volume
2008-01-02	28.467142	28.608572	27.507143	27.834286	18.639585	269794700
2008-01-03	27.915714	28.198572	27.527143	27.847143	18.648197	210516600
2008-01-04	27.35	27.571428	25.555714	25.721428	17.22468	363958000
...	...	...	....	...	...	...
2017-12-27	170.100006	170.779999	169.710007	170.600006	168.076645	21498200
2017-12-28	171	171.850006	170.479996	171.080002	168.549545	16480200
2017-12-29	170.520004	170.589996	169.220001	169.229996	166.726913	25884400

Table 1. Data format example

Data pre-processing

Software:

Data handling and processing relies on programming language Python (version 3.7), software application PyCharm CE (community edition 2018.1); and programming language R (version 3.5.1), software application R Studio (version 1.1.456);

Feature selection:

All features (the six features in Table 1) will be considered within the LSTM training process;

Data normalization and transformation:

The data which is fed to the LSTM network will be normalized through the following equitation:

$$x = \frac{X_r - \mu}{\sigma}$$

Where X_r represents the raw data items, μ is the mean of each attribute for stock data and σ is the standard deviation for each attribute. Take the attribute 'Open' (the daily opening price) as an instance, to compute μ, the sum of the open prices throughout the whole period S should be calculated first, then μ can be obtained by computing S/n (n is the total number of trading days); after we calculate μ, we can derive the standard deviation based on the following formula:

$$\sigma = \frac{\sum_{i=1}^{N}(X_r - \mu)^2}{n}$$

After we get μ and σ, we can derive one x for each X, and we call x is the normalized expression of X.

3.2. Stock selection: Beta coefficient

As one of the research objectives clearly specifies, this project concentrates on the two portfolios' performance, one is an aggressive portfolio and the other is a defensive portfolio.

Generally, 'aggressive' and 'defensive' resent totally different characteristics.

'Aggressive' means the riskiness of a certain financial asset is higher than the market portfolio, and 'Defensive' signifies the riskiness that an asset face is lower than the market portfolio (Boskovska and Svrtinow, 2016).

A criterion concerning how to define 'aggressive' and 'defensive' portfolios is indispensable to construct portfolios. Therefore, to achieve this goal, the stock's beta coefficient is adopted to divide stock candidates into the appropriate categories.

Beta coefficient (β) measures a certain asset's risk concerning the market portfolio. More specifically, this value implies the variability of the rate of return for or individual security concerning the average return rate variability for the overall market portfolio. It is expressed as follows:

$$X = \frac{S_{t+1} - S_t}{S_t} \qquad (1)$$

$$\beta = \frac{Cov(X,Y)}{Var(Y)} \qquad (2)$$

Where:

X: rates of the return based on the daily price change of each stock

$Cov(X,Y)$: *the covariance of* X *and* Y

$Var(Y)$: *tthe he variance of markethe t portfolio*

Y: the daily return of the S&P 500 index

The daily prices S_t are applied to equitation (1) to calculate the rates of return.

Once we have the computed daily return (X), we can get the beta values from the equitation (2). The value of beta is always positive and can be above or below +1. If the value of beta is lower than +1, this indicates the variability of this asset's return rate is lower than that of the market portfolio. To illustrate, when the beta is 0.5, the asset price will increase (or decrease) by 2% if the price of the market portfolio increases (or decreases) by 4%. On the other hand, if the value of beta is above +1, this indicates the variability of this asset's return rate is higher than that of the market portfolio. For instance, when the beta of a certain asset is 2, the asset price will increase by 8% if the price of the market portfolio increases by 4%. In other words, the risk of this asset is twice higher than the market portfolio (Boskovska and Svrtinow, 2016).

It is time-consuming if we randomly pick stocks and test stocks' beta aimlessly. To make the process easier, we can find stocks within certain sectors, as mentioned before. In this way, we can narrow the scope where we can select stocks. We can focus on several industries that may invest in aggressive or defensive stocks.

To seek aggressive stocks, we can pay more attention to the cyclical industry, since the stocks in this industry face higher volatility of revenue, leading to higher stock returns and sensitivity to the market index. Obviously, high-tech stocks can be regarded as aggressive. Besides, the airline and tourism industries are fairly cyclical (Investopedia, 2017). People have more disposable income in prosperous economic times, so they tend to take vacations and make air travel. Conversely, during sluggish economic times, people are cautious about spending. As a result, air service companies face big differences concerning profits, which contributes to different firms' profitability and volatility of stock return. According to this, ten stocks from air service and high-tech industries were chosen:

Company name	Stock code
Advanced Micro Devices, Inc	AMD
HP Inc.	HPQ
Cisco Systems, Inc.	CSCO
Intel Corporation	INTC
Oracle Corporation	ORCL
Microsoft Corporation	MSFT
Sony Corporation	SNE
The Walt Disney Company	DIS
China Eastern Airlines Corporation Limited	CEA
Telefonaktiebolaget LM Ericsso	ERIC

Table 2. The selection of aggressive stocks

Defensive stocks can be found in the daily necessity sectors, such as the catering industry, pharmaceutical firms (the company Merck & Co., Inc. in the below table belongs to this type), public utilities (including energy companies like NextEra Energy, Inc. and American Electric Power Company, Inc.).

Company name	Stock code
Unilever NV.	UN

Walmart Inc.	WMT
PG&E Corporation	PCG
Johnson & Johnson	JNJ
NextEra Energy, Inc.	NEE
American Electric Power Company, Inc.	AEP
Coca-Cola Consolidated, Inc.	COKE
PepsiCo, Inc.	PEP
McDonald's Corporation	MCD
Merck & Co., Inc.	MRK

Table 3. The selection of defensive stocks

The specific values of the beta coefficient can be found in section 4.1.

3.3. Stationary test and Augmented Engle-Granger two-step cointegration test

3.3.1 Stationary test

First, a stationary test should be conducted on stock price series and price spread separately to confirm the cointegration relationship. Second, developed by Dickey and Fuller in 1979 (Huck, 2015), the Augmented Dickey-Fuller unit root test can be used to test the cointegration relationship between two objects after conducting the stationary test (Kwon and Shin, 1999).

The null hypothesis of the Augmented Dickey-Fuller (ADF) is that a unit root is present in a time series sample (the presence of unit root implies the series is non-stationary), with the alternative that there is no unit root. If the p-value is above a critical size, then we cannot reject that there is a unit root; alternatively, we can reject the statement 'there is a unit root' if the p-value is lower than a critical size. The more negative the test statistic is, the stronger the rejection of the hypothesis that there is a unit root at a certain level of confidence (MacKinnon, 1996)

If the test statistic (p-values) is significant at a 1% rate (or less) in the stationarity test, we can reject the null hypothesis. The p-values are obtained through regression surface approximation from MacKinnon 1994, but using the updated 2010 tables. If the p-value is

close to significant, then the critical values should be used to judge whether to reject the null.

Once the test statistics verify that the one certain pair whose two members' price series are non-stationary but the series of their spread is stationary. This pair can be selected as a pair candidate, which can be tested for the cointegration relationship.

Ten stationary tests on ten stocks will be conducted firstly. We have 10*(10-1)/2=45 pairs, so 45 stationary tests on 45 price spreads will be implemented.

3.3.2 Augmented Engle-Granger two-step cointegration test

Engle and Granger initially introduced the concept and definition of the cointegration. They illustrated that a stationary time series could be generated from a linear combination of two non-stationary time series.

$$S_{t,i,j} = p_{t,i} - \beta * p_{t,j} - \alpha \quad (3)$$

Where p_i and p_j are the closed prices of stock *i* and *j* respectively and both of them are non-stationary. α and β are parameters that can be estimated by using Ordinary Least Squares (OLS).

The data used in both stationary and cointegration tests are the adjusted close price during the eight years (2008.1.2 ~ 2015.12.31).

3.4 Long Short-Term Memory algorithm

LSTM is a special variant of RNN proposed by Hochreiter and Schmidhuber (1997) and has become a widespread approach in predicting time series. It gets its exceptional predictive ability from the cell state's existence that allows it to understand and learn longer-term trends in the data. This is especially important for stock price data. The structure of the LSTM is demonstrated in Figure 6.

Variables specification for Figure 5:

C_{t-1}*: old cell state* $h_{the\,t-1}$*: Output of the previous cell*

C_t: present cell state the h_t: Output of the previous cell

i_t: input gate layer f_t: forget gate layer

o_t: output sigmoid gate layer;

the blue signs (either multiplication or addition) represent the corresponding linear functions.

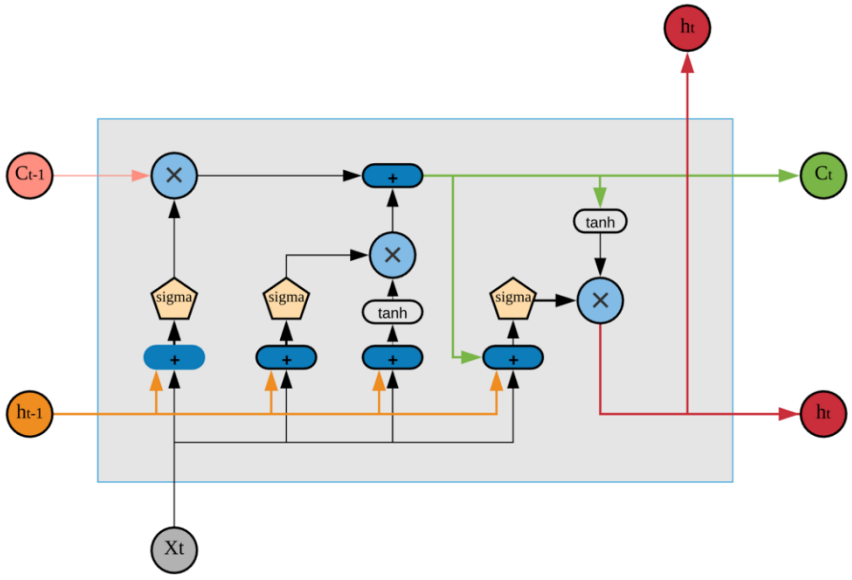

Figure 6. Structure of LSTM network

It can be seen that the cell state receives two input sources: the output (previous cell state, C_{t-1}) and the new input (X_t)

According to the above picture, the procedure of LSTM can be explained as the following: Firstly, the cell of LSTM processes the information from the previous memory state and decides the information be removed or forgotten from the cell state, W_f denotes the weight of the forget gate.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Secondly, how much information will be stored is decided by the LSTM cell; the input gate decides which piece of information to be updated and the tanh layer will update the vector.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

The next step is to update the cell state by combining the two parts we got from the above calculations:

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

Finally, to obtain the output, LSTM utilizes the output gate to control the cell state

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

3.5 Simplified interpretation of machine learning algorithm mechanism

The primary mission of machine learning in this project is to use the input data to train a model used for prediction purposes and then test the accuracy of predictions (Heaton et al., 2017).

In order to illustrate this step, the whole dataset will be divided into two subsets for training and testing purposes, respectively. The first dataset subset is used to train and find an optimal model with appropriate parameters (the model can find specific patterns in the dataset); the second subset is to test the prediction accuracy of the model. Once a machine learning algorithm learns the underlying patterns of the training dataset, it needs to be tested on fresh data (or test data) that it has never seen before. The model's ability to extract and generalize the pattern from data should be verified in the testing stage.

4. Numerical results

4.1 Stock selection section

Based on the pre-specified rule concerning how to categorize stocks in section 3.2, the stocks can be classified as the following tables:

	Stock code	**Beta coefficient values**
Aggressive	AMD	1.429605
	HPQ	1.073733
	CSCO	1.046079
	INTC	1.023711
	MSFT	1.056504
	ORCL	1.019372

	SNE	1.070834
	DIS	1.062177
	CEA	1.268516
	ERIC	1.242173

	Stock code	**Beta coefficient values**
Defensive	UN	0.755048
	WMT	0.506855
	PCG	0.545610
	JNJ	0.562269
	NEE	0.686814
	AEP	0.633541
	COKE	0.701055
	PEP	0.527201
	MCD	0.572980
	MRK	0.729652

Table 4. Beta coefficient (β) values for different stocks

As the above two tables demonstrate, the stocks with β far lower than one are categorized into the defensive stock group, and the stocks with β above one are classified into aggressive type.

4.2 Stationary and Augmented Engle-Granger two-step cointegration test

4.2.1 Stationary test

After filtering out the 20 stock candidates (shown in Table 4), it is reasonable to conduct two tests, namely the stationary test and the Augmented Engle-Granger two-step cointegration test, to explore whether the price series are qualified to be used for further research purpose. The stationary test should be conducted firstly as the premise of the cointegration test is as follows. The two stock price series should be non-stationary, but the price spread (the difference between the two price series) should be stationary.

The rationale of the Augmented Dickey-Fuller unit root test (ADF) and Augmented Engle-Granger two-step cointegration test are illustrated in section 3.3.

The stationary test is conducted by the Augmented Dickey-Fuller unit root test (ADF) test in the statsmodels package available in python. Statsmodels contains various models and functions specially designed for time series analysis, and the function 'stattools.adfuller' is used to implement a stationary test.
Table 5 demonstrates the stationary test results.

	test statistics	p-value	Critical value for the test statistical at different level		
ORCL	-0.8466	0.8050	1%: -3.4329	5%: -2.8627	10%: -2.5673
CEA	-2.2627	0.1843	1%: -3.4329	5%: -2.8627	10%: -2.5673
ORCL-CEA	-4.9033	0.00003.4348	1%: -3.4330	5%: -2.8627	10%: -2.5674

	test statThe cstics	p-value	The critical value for the test a statistical at a different level		
JNJ	1.1805	0.9958	1%: -3.4329	5%: -2.8627	10%: -2.5673
NEE	1.8435	0.9984	1%: -3.4329	5%: -2.8627	10%: -2.5673
JNJ-NEE	-4.3724	0.008375	1%: -3.4330	5%: -2.8627	10%: -2.5674

Table 5. The statistical results of the stationary test

In each line of the results, five float numbers are present. The first two numbers (test statistic and p-value) of each result are asymptotic p-value based on MacKinnon approximate, the first number is the test statistic of the ADF test. The other three numbers are critical values for the test statistic at 1 %, 5 %, and 10 % based on the regression curve. According to the predefined rule in 3.3: if the test statistic is significant at a 1% rate (or less), we can reject the null hypothesis. We can conclude that the four stocks' price is non-stationary and the two series of the spread are stationary according to the table, and we can use these two pairs to conduct the next stage, cointegration test.

4.2.2 Augmented Engle-Granger two-step cointegration test

There are ten stocks in an aggressive stock set, so $\frac{(10-1)*10}{2}$ = 45 pairs can be generated among different stocks. Among these pairs, the cointegration test will be conducted for each pair and the other pair with the strongest cointegration relationship will be selected

out. Similarly, another pair from a defensive portfolio will be filtered out, and these two pairs comprise the aggressive and defensive portfolio, respectively. The cointegration test belonging to the statsmodels package in python is used to test the cointegration relationship between variables.

	Result in python
ORCL and CEA	sm.tsa.stattools.coint(orcl, cea) >(-3.8930207508983967, 0.010141531127109688, array([-3.90079646, -3.33855861, -3.04613545])) sm.tsa.stattools.coint(cea, orcl) >(-3.5796546047779265, 0.02595770897461987, array([-3.90079646, -3.33855861, -3.04613545]))
JNJ and NEE	sm.tsa.stattools.coint(jnj,nee) >(-4.889060031945324, 0.00026196221864075125, array([-3.90079646, -3.33855861, -3.04613545])) sm.tsa.stattools.coint(nee,jnj) >(-4.821816456834575, 0.0003453336675023922, array([-3.90079646, -3.33855861, -3.04613545]))

Table 6. The result of Augmented Engle-Granger two-step cointegration test

In each result, five float numbers are present. The first two numbers of each result are asymptotic p-value based on MacKinnon approximate, and the three numbers in 'array()' are critical values for the test statistic at the 1 %, 5 %, and 10 % levels based on the regression curve.

The Null hypothesis (H_0) here is: there is no cointegration between the two variables. The alternative hypothesis (H_1) is a cointegrating relationship. Under the statistical meaning, if the p-value is small (below a critical size), then we can reject the hypothesis that there is no cointegrating relationship. Therefore, the alternate can be considered valid (e.g., the two series have the cointegration relationship) if the null hypothesis is rejected.

Based on the second pair (JNJ and NEE) results, the test statistics (-4.889060031945324 and -4.821816456834575) are smaller than the three levels critical sizes; simultaneously, the p-values for the test is smaller than 1%. According to the above two numerical evidence, it is reasonable to reject H_0 (there is no cointegrating relationship), which means we can confidently (with 99% confidence) say there is a cointegration relationship between the two price series.

Likewise, for the first pair (ORCL and CEA), the test statistics (-3.8930207508983967 and -3.579656047779265) are lower than -3.33855861 (5%). We can identify a cointegration relationship between the two price series with 95% confidence.

Based on the above two test results, we can use these two pairs (ORCL & CEA and JNJ & Na EE) to conduct a pairs trading strategy.

4.3 Evaluation of performance for price prediction (Figure 7-12)

Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) have been adopted to evaluate the prediction accuracy. They are computed as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\{Y_{act}(i) - Y_{pred}(i)\}^2$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{Y_{act}(i) - Y_{pred}(i)}{Y_{act}(i)}\right| * 100$$

	ORCL	CEA	spread	NEE	JNJ	spread
MSE	83.73548	8.710255	146.459	1687.807	427.2755	416.6617
MAPE	0.0149134	0.01699028	0.04867455	0.03123961	0.03378202	1.348482

Table 7. Statistics for evaluating the prediction performance

In the above two equitation, $Y_{act}(i)$ and $Y_{pred}(i)$ denote the actual and predicted prices of the corresponding variables respectively, and N is the total number of observations, 500 (recall that we use eight years' data to predict the prices for 2016-2017, and the period of 2016-2017 has 500 trading days). The two 'spread' represent the price spread of ORCL and CEA, NEE and JNJ separately.

The lower values of MSE and MAPE indicate that a better prediction for effectiveness. When assessing the prediction performance from the individual stock perspective, LSTM can effectively and accurately predict the stocks' prices that belong to the aggressive portfolio, especially for CEA, because the value of MSE for CEA prediction is quite low (8.710255).

Nevertheless, LSTM does not excellently perform the forecast for the two stocks of defensive stocks. As the values of MSE for NEE and JNJ indicate, the deviation level is much higher than that of ORCL and CEA, in which two are in the aggressive portfolio. The right part for the defensive portfolio, it can be seen that the value of MSE for NEE is remarkably higher and the value of MAPE for the price spread is greater than 1 (1.348482), which means that the predicted stock prices deviate from the real values significantly compared to the MAPE for aggressive portfolio's spread (0.04867455). It implies the LSTM works more efficiently on an aggressive portfolio than the defensive one.

Figures 9-12 show that the predicted values for all these stocks can forecast the general trend of the real price. More specifically, the line of predicted prices for CEA shows a similar upward or downward movement trend with the real costs. The line of predictions for the other three stocks also presents a similar direction or pattern with the real situation. However, some fluctuation and errors still exist.

Another visual feature stemming from Figures 9-12 is that, although the predictions show similar trends in most cases, the lags are present. For example, although the line for predicted prices shows the same downwards direction with the line for actual prices, it takes more time for the predicted lines to react and show a declining trend.

Additionally, Figures 9 and 12 demonstrate the same conclusion mentioned above since they can be easily observed. In Figures 8, 9 and 10, the line of predicted values is tightly close to the line of real values while the predicted values in those figures are not that precise, the lags can also be observed.

4.4 Trading strategy and profit calculation

4.4.1 Trading strategy framework

The trading will open when the two stock prices diverge abnormally, which means that the price spread between two stocks deviates from the historical mean by more than 1.5

historical standard deviations. And the trading closes when the spread reverts to the normal range (mean $\pm$ 1.5 standard deviations). It means that if the spread stays in the space out of mean $\pm$ 1.5 standard deviations, the trading status will remain open continuously until the spread comes back to the normal range.

4.4.2 Profit calculation based on the approach of Gatev

When the price spread deviates from the normal range, we can say the trading signal emerges. The trading signals may occur at different time points during the two-year period (2016-2017), so the portfolios are likely to have multiple cash flows during the trading interval. For the pair which has opened the trading but does not show a converge trend yet at the end of 2017, the cash flows only occur at the last one day of the period studied (2017.12.29).

Trading interval (when the trading opens) refers to the period when the price spread departs from the mean by more than 1.5 standard deviations; the period for predicting is 2016-2017 (2016.1.4~2017.12.29)

Figure 7. The actual versus predicted price spread of ORCL and CEA

Figure 8. The actual versus predicted price spread of JNJ and NEE

Figure 9. The actual versus predicted price spread of ORCL

Figure 10. The actual versus predicted price spread of CEA

Figure 11. The actual versus predicted price spread of JNJ

Figure 12. The actual versus predicted price spread of NEE

The cumulative profit is calculated based on the approach of Gatev et al. (2006), who is the pioneer in pairs trading domain, as mentioned in the literature review section.

The trading strategy can be seen as self-financed as we the short and long the two stocks for the same amount. To illustrate, we will short $1 the relatively overpriced stock, which means that we will short $\frac{\$1}{the\ stock\ price\ of\ the\ overpriced\ one}$ shares of the overvalued stock. Similarly, we long $1 the relatively underpriced stock ($\frac{\$1}{the\ stock\ price\ of\ undervalued\ one}$share). The stock prices here are adjusted close prices - for instance, the trading opens on the first day and still opens if the spread walks in the space beyond mean $\pm$ 1.5 standard deviations. At the end of the trading interval, all positions will be closed at the prices of the last day of the trading interval.

Take the pair, ORCL and CEA as an example, three scenarios can explain why the spread (ORCL - CEA) between these two stocks departures from the mean by more than 1.5 historical standard deviations:

① ORCL's stock price is overvalued, while the price for CEA remains at a normal level;
② ORCL's stock price remains in a normal range, while the price for CEA is undervalued;
③ ORCL's stock price is overvalued, and the price for CEA is undervalued at the same time.

In the above three situations, one common point is that one stock is overvalued or undervalued relative to the other stock.

Price spread (JNJ-NEE) based on predicted prices

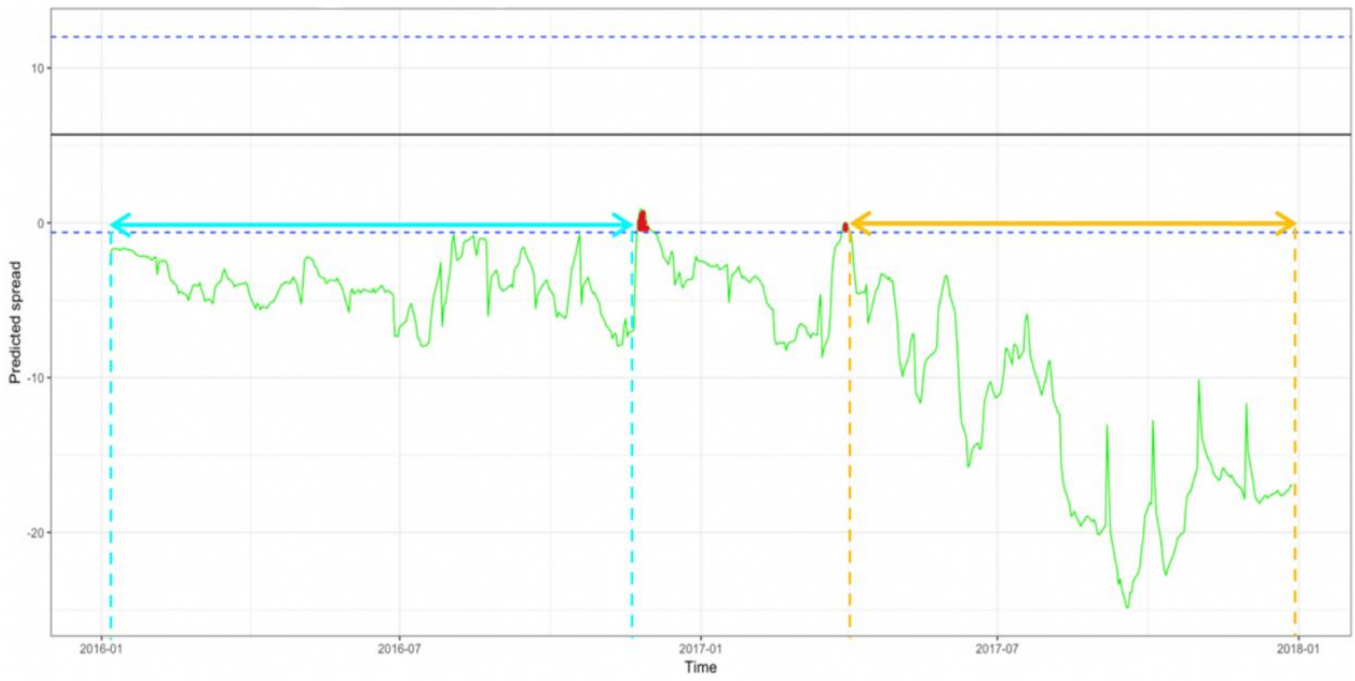

Figure 13. The explanation for profit calculation

Once the trading is triggered, we buy the undervalued stock and short the overvalued stock simultaneously. When the spread comes back to the normal range, we close out the positions by buying back the overvalued one and selling the undervalued one.

During the predicting period (2016.1.4~2017.12.29), if the price spread does not diverge and stays in the normal range (this period is the small space colored in red), there will be no profits to exploit.

When calculating the profits of the strategy three occasions for a trading interval, two situations may occur (please refer to Figure 13 to fully understand):

I. The whole trading interval is covered in the trading period (both the spread divergence and convergence occur during the period), in Figure 12, this period is the interval colored in bright blue:

$$s_1 = \sum_{i}^{n} \frac{\$1}{\text{the stock price of the overpriced one in the day } i}$$

$$s_2 = \sum_{i}^{n} \frac{\$1}{\text{the stock price of the underpriced one in the day } i}$$

$$p_i = s_1 * price_1 - s_2 * price_2$$

$$profit = \sum_{i=1}^{n} p_i$$

s_1: The number of shares for the undervalued stock;

s_2: The number of shares for the overvalued stock;

$price_1$: The adjusted close price for the undervalued stock on the last day of the trading interval;

$price_2$: The adjusted close price for the overvalued stock on the last day of trading interval.

p_i: The profits in trading interval i;

Part of the trading interval is in the period (the price spread diverges but converge does not occur during the period studied). In Figure 13, this period is the interval colored in bright yellow.

Profit calculation is similar to the scenario I:

$$s_1 = \sum_{i}^{n} \frac{\$1}{\text{the stock price of the overpriced one in the day } i}$$

$$s_2 = \sum_{i}^{n} \frac{\$1}{\text{the stock price of the underpriced one in the day } i}$$

$$profit = \sum_{i=1}^{n} p_i$$

$$p_i = s_1 * price_3 - s_2 * price_4$$

$price_3$: the adjusted close price for the undervalued stock on the last day of the period we studied (2017.12.29);

$price_4$: The adjusted close price for the overvalued stock on the last day of the period we studied (2017.12.29);

III. As shown in Figure 13, no trading interval emerges. This situation is highlighted in red, and this period does not generate any profits because there are no trading opportunities. Results and analysis between Figures 7 and 12 support the effectiveness and successful demonstration for analytics and visualization. Real-time checks can be used to help improve the quality of our analysis at all times.

4.5 Performance Evaluation

The hardware environment is as follows. Two machines were used for performance evaluation with the identical hardware infrastructure. Each machine had Intel® Core™ i7-10700K Processor with 8 cores, 16 threads, and 5.1 GHz max, 32 GB RAM, 480 SSD hard-disk, 6 GB GTX 1660 graphics card and 10 Gbps network speed. This allowed the maximum utilization of CPU and GPU powers, as well as the fast network speed, for the optimum performance. Two types of experiments were conducted. The first experiment focused on the execution time for auto-trading between 1,000 and 10,000 times between the aggressive and defensive portfolios. The reasons for performing auto-trading between 1,000 and 10,000 times were that each auto-trading simulation took place quickly and up to 10,000 times could provide reasonable execution time records. The second type of experiment was focused on long-term profitability by using aggressive and defensive portfolios. The objective was to allow auto-trade and measure the expected profitability, with both predicted and actual prices calculated and compared. Each experiment took three times of measurements to get the mean values.

4.5.1 Performance evaluation on execution time

This section shows the results of our performance evaluation. Each auto-trading can be considered as a simulation. Our work can allow up to 10,000 simulations. Figure 14 shows the execution time of completing between 1,000 and 10,000 simulations, starting from 4.85 to 67.14 seconds. There is a directly proportional relationship between the execution time and the number of simulations equivalent to linear regression. In financial computing, it is common to run over-night tests to identify any errors and ensure the trading activities the following day can be smooth. Our work can run overnight tests of up to 10 hours. The default is to run 10,000 simulations that our program can continuously run for 10 hours. At the end of each hour, the mean execution time was recorded. Figure 15 shows the execution time of completing 10,000 simulations in 10 hours. The curve looks like a gentle parabola and it is not a linear regression. After every hour, slightly more time is added on top of the previous mean time taken. Errors and uncertainties can be maintained within a 3% difference. It shows that our performance has been acceptable for overnight tests.

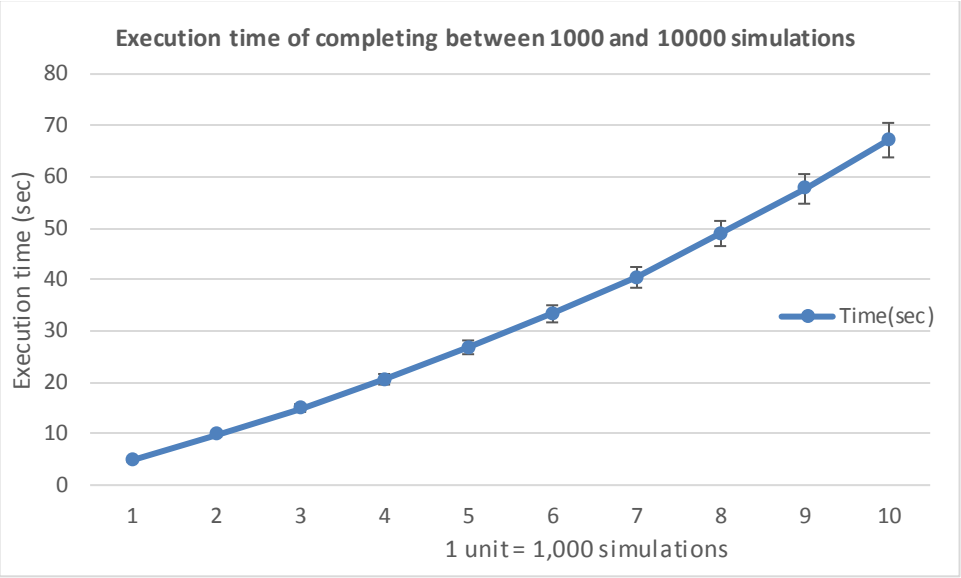

Figure 14: Execution time of completing between 1,000 and 10,000 simulations

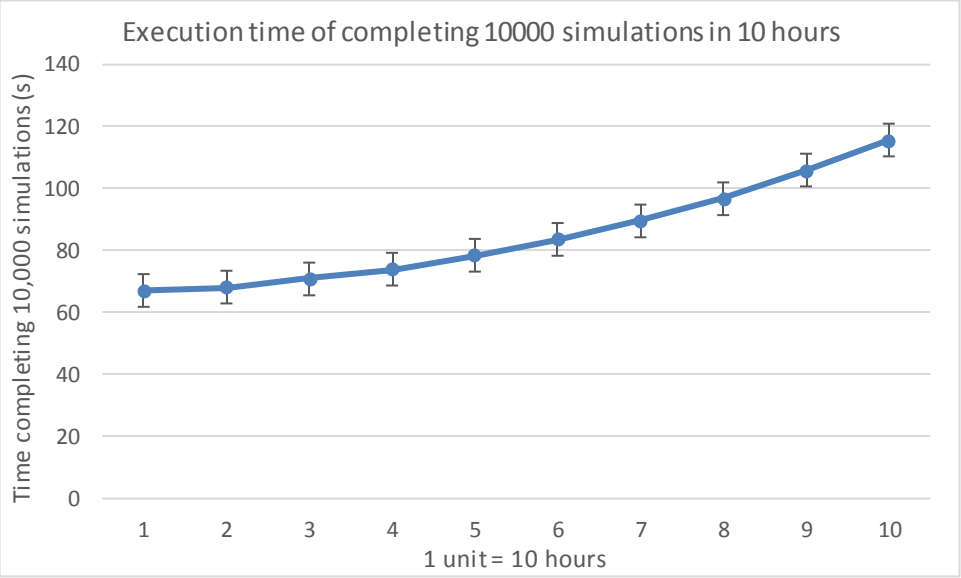

Figure 15: Execution time of completing 10,000 simulations in 10 hours

4.5.2 Predictive tests

In financial computing, it is important to compare the actual and predicted prices. For example, our previous work (Chang et al., 2019) has demonstrated an accurate comparison between the actual and predicted prices and risks management. Our program can simulate long-term trading for both aggressive and defensive strategies. This allows potential investors to understand the possible outcomes and any risks involved during the investment period. For predictive tests, we simulate in identified trading intervals to ensure that aggressive and defensive portfolios can be simulated at those periods. Definitions of profits were explained in Figure 13. Table 8 summarizes all the results for comparison.

Results show that for the long-term investment, a defensive portfolio can acquire higher profitable values.

		Trading intervals	**Total trading days**	**Profits**
Aggressive Portfolio	Predicted price	Six trading intervals are present, covering 371-378, 383-400, 402-420,422-460, 462-476, 486-490th days	114 days in total	1.648179
	Actual price	Five trading intervals (covering 369, 372, 387-401, 403-428, 430-473th days	87 days in total	5.066765
Defensive portfolio	Predicted price	Three trading intervals (covering 1-224，232-308，313-500th days	489 days in total	6.747507
	Actual price	13 trading intervals, (covering 1-72, 78-134, 137-140, 142-144, 146-147, 151-162, 168-171, 173-187, 193-214, 227-229, 231-238, 241-300, 302-500th days.	461 days in total	29.94679

Table 8. Comparison between Profits based on predicted stock prices and actual profits based on the real price

5 Discussion of the results

The discussion concerning stock price prediction

Based on the numerical results generated from section 4.3 and 4.4.2, two conclusions can

be derived:

i) LSTM performs better on the price prediction for aggressive stocks rather than defensive ones;

ii) Using LSTM to make price predictions can afford arbitrage opportunities and make profits, but this approach earns fewer profits compared to the real situations for both types of portfolios.

Some factors exert effects when we derive the conclusions and the following will discuss what possible reasons contribute to such results.

5.1. The selection of a threshold for pairs trading

During the trading process, 1.5 standard deviations are regarded as the tolerable scope; alternatively, 1.5 standard deviations are chosen as the line of demarcation, which means that once the spread is greater or lower than the mean by 1.5 standard deviations, the trading will open.

However, the number of the standard deviation is decided by the investors' risk tolerance. For example, the people who want to make frequent transactions may choose to open the trading when the spread 1 deviates one standard deviation or less from the historical mean. In contrast, others who prefer the low-frequency trading pattern may open trading when the spread is out of the space within mean $\pm$ 2 standard deviations. Once the tolerance level varies, the profits concerning the two strategies will also change according to the different thresholds.

5.2. Problems stemming from LSTM

Each of the parameters we set (learning rate, the number of layers, the neuron units, etc.) in the LSTM algorithm will profoundly impact the learning performance, so predicting accuracy can vary by adjusting these parameters.

5.2.1 Overfitting problem of LSTM

The overfitting problem in the machine learning process is common. This problem refers to the LSTM model shows fairly precise prediction on the training set, but the model may be unable to predict accurately when dealing with the test dataset, and the model does

not have the capability of generalization, so the model fails to make predictions precisely when it meets new dataset.

To mitigate this problem, dropout, a select method, has already been used in the training process. This approach comes to effect by strengthening the model's generalization ability. However, the overfitting problem may still exist.

5.3. Analysis and LSTM during the period, the characteristics of the stock market

5.3.1 LSTM used for analyzing macroeconomic conditions

This project focuses on the period spanning from 2008 to 2018 (including 2008, but 2018 is excluded). During this particular period, the financial world witnessed a massive revolution, various types of economic activities and even some financial regulations. For instance, this decade incorporates several intervals with intricate financial contexts and significant events, such as the financial crisis, the recovery of the global financial market, etc. These events make the overall market turbulent and unpredictable (Grout and Zalewska, 2016). However, the tool (LSTM) we employ to make predictions does not consider these external factors, which contributes to some inaccuracy of prediction.

5.3.2 LSTM for the stock market

Share prices often reflect the overall economy and can be volatile as investors react to financial news and current events, but not all traders can react to these events correctly. This means that noises will arise since some irrational investors may take on reverse actions to reality and make the market more complicated. Hence, it is challenging for LSTM to capture the exact pattern of the price movement.

5.4 Evaluation, limitation and future work

The two research objectives stated in the introduction:

i) When employing a machine learning technique (LSTM) to forecast price movement for the aggressive and defensive portfolio, which portfolio presents a better performance?

ii) In terms of cumulative return, which portfolio can outperform the other by trading predicted prices to trade?

They can be critically answered as the following:
Under the circumstance where using the adjusted close prices of 2008.1.2 ~ 2015.12.31 to forecast the prices for 2016.1.4 ~ 2017.12.29, LSTM shows higher prediction accuracy on aggressive stocks and their spread as well.

The profits generated from a defensive portfolio exceed that of an aggressive portfolio, whether using the predicted prices or using the actual prices to calculate the return.

Limitations and deficiencies exist throughout the whole process. First, the period studied is unique, as it contained several intricate financial intervals. During different ranges, the beta coefficient may vary significantly, so the indicator, the beta coefficient value, may not be considered the optimal parameter to determine whether the stock is aggressive or defensive so that it may misclass the stocks into the wrong stock types.

Second, transaction costs are not taken into account throughout the whole trading process. In the real financial market, transaction costs could be vast and non-negligible, especially when the trading frequency is high and trading volume is large.

Next, the profit calculation is based on the adjusted close prices, while it may be challenging to take trading positions exactly at the adjusted close prices. Therefore, it might be unrealistic to trade at the adjusted close prices. Additionally, the prices for the same stock also vary during a different period of a single day. If the machine learning techniques can forecast the exact time, such as during which hour even the minute when the most optimal price for conducting trading will emerge, the return can be fully exploited.

Furthermore, more machine learning tools can be deployed to improve the predicting precision. In this project, only one type of machine learning tool, LSTM, is applied to

achieve predicting purpose. So, the accuracy of stock price prediction is merely investigated by using LSTM and no other tools are involved in the prediction process. Nevertheless, accuracy may be improved if we attempt to use other types of machine learning algorithms. In future work, various algorithms can predict price series, and comparison concerning the prediction precision may help identify the most effective predicting tool.

Even though the cointegration relationship between the two assets is verified based on historical data, the cointegration relationship may be destroyed by certain factors of the volatile financial market. Such a relationship may be absent for several periods. In principle, the cointegration relation between two assets will last for an extended period. However, as the financial market experienced unpredictable changes every day, the following scenario is possible: the two objects historically share a cointegration relationship that may not still follow this relation during certain days due to the unusual market conditions. If the above situation emerges, it is hard to predict the price movement according to the historical cointegration relationship. Moreover, the prediction accuracy will be thus adversely affected when the two objects do not follow cointegration. However, we still assume that there is a cointegration relationship just based on historical data.

5.5 Research contributions

To sum up, our research contributions are as follows:

1. Design and implementation of LSTM to achieve a good accurate analysis and forecasting – we demonstrated clearly how the LSTM approach could enhance the accuracy and quality of our analysis and prediction to a large extent.
2. The demonstration of modern analytics and visualization in Finance – our analysis can provide in-depth comparisons of our focused stocks and make both real-time and off-time analysis. This matches the demands in current financial research and the market. Our analysis and prediction can also be verified through analytics and visualization.

How our paper is relevant to this special issue is explained as follows. Our research is suitable to "Models, Theory, and Methods for Interactive Computational Visual Analytics", since we have explained how LSTM can be used effectively for computational analysis in analytics and visualization. Our work is related to "Real-World

Applications Using Interactive Computational Visual Analytics", as we have analyzed large scale, high dimensional, streaming and real-time data with the step by step approach. Our analysis contributes to "Evaluation of Interactive Computational Visual Analytics", because we have demonstrated a novel evaluation technique for financial investment, analysis and prediction.

6. Conclusion

Pairs trading became a prevalent arbitrage strategy in the current financial industry. Three main methods were introduced to implement this project, namely, the beta coefficient, Augmented Engle-Granger two-step cointegration test and machine learning technique (LSTM), and they were deployed for three purposes respectively: i) initially selecting stocks; ii) identifying potential pairs and iii) predicting adjusted close prices. By selecting two pairs (ORCL, CEA and JNJ, NEE) to construct the aggressive and defensive portfolio based on the first two methods, the mean and standard deviation of spread (ORCL-CEA and JNJ-NEE) would be calculated based on the previous 8-year price data. During the forecasting stage, the first eight years' data of the four stocks were used to train the algorithm to find underlying patterns existing in data, and the model would predict the price for a period of 2016.1.4~2017.12.29 (2 years). After obtaining four price series, the two spread series were easily accessible, and they would be used throughout the profit computation process.

In our research, the threshold was set as 1.5 standard deviations. In other words, when the spread walked in the space beyond mean $\pm$ 1.5 standard deviations, the trading algorithms by LSTM would open by taking a long position of \$1 for overvalued assets and taking a short position of \$1 for the overvalued asset simultaneously. Once the spread reverted to the normal level, all positions were closed, and the profits would be available.

According to the numerical results, it can be concluded as follows. i) LSTM performs better when predicting price movement on aggressive stocks and ii) for two portfolios, the profits generated by using predicted prices are lower than the profit calculated by using the actual price. However, the defensive portfolio presents higher profits both in the cases of utilizing the predicted prices and actual prices (the profits for defensive portfolio: 6.75 predicted VS 29.95 actual; the profits for aggressive portfolio: 1.65 predicted VS 5.07 actual).

Deficiencies are present in the research process. First, the length, the start, and endpoint of the study period should be cautiously selected because the market conditions have profound impacts on stock performance. These macroeconomy factors may influence the cointegration relationship between two objects. Thus, the pairs identification step will be affected. The cointegration relationship may be 'spurious', which means that the cointegration relationship between the pair may be absent due to the unpredictable market conditions. However, the cointegration relationship is already confirmed based on historical performance. The accuracy of prediction also can be enhanced by investigating more predicting models, while LSTM is the only tool used in this project.

The threshold of pairs trading is regarded as one key determinant of high profitability. The specific value of the threshold deserves more investigation for the investors who desire to obtain a higher return and fully exploit the arbitrage opportunities.

In conclusion, long short-memory can be cautiously applied to make stock price predictions and the profitability of pairs trading on different types of portfolios (aggressive or defensive) will vary depending on a series of market factors. We demonstrated the effective use of combining analytics and visualization with LSTM to achieve more accurate analysis, prediction and investment strategies.

Acknowledgment

This work is supported by VC Research (VCR 0000052) and the National Natural Science Foundation of China (Grant No. 61872084).

.

References

Boskovska, D. and Svrtinov, V. (2016). Risk of shares on the Macedonian capital market measured by the beta coefficient. *Economic Development.* 1(2), pp.79-90.

Chang, V., Li, T., & Zeng, Z. (2019). Towards an improved Adaboost algorithmic method for computational financial analysis. *Journal of Parallel and Distributed Computing*, *134*, 219-232.

Chaudhuri, T., Ghosh, I. and Singh, P. (2017). Application of Machine Learning Tools in Predictive Modeling of Pairs Trade in Indian Stock Market. *IUP Journal of Applied Finance,* 23(1), pp.5-25.

Chen, Y., Chen, W. and Huang, S. (2018). Developing Arbitrage Strategy in High-frequency Pairs Trading with Filterbank CNN Algorithm. In: IEEE, *2018 IEEE International Conference on Agents (ICA).* Singapore, 28-31 July 2018. IEEE.

Dunis, C., Rosillo, R., Fuente, D. and Pino, R. (2013). Forecasting IBEX-35 moves using support vector machines. *Neural Comput & Applic,* 23, pp.229-236.

Fisher, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research,* 270, pp.654–669.

Gatev, E., Goetzmann, W. and Rouwenhorst, K. (2006). Pairs trading: performance of a relative value Arbitrage rule. *The Review of Financial Studies,* 06(19), pp.797-827.

Grout, P. and Zalewska, A. (2016). Stock market risk in the financial crisis. *International Review of Financial Analysis,* 46, pp.326-345.

Göncü, A. and Akyildirim, E., (2016 a). Statistical Arbitrage with Pairs Trading. *International Review of Finance,* 16(2), pp.307–319.

Göncü, A. and Akyildirim, E. (2016 b). A stochastic model for commodity pairs trading. *Quantitative Finance,* 16(12), pp.1843–1857.

Huang, B., Huan, Y., Xu, L., Zheng, L. and Zou, Zhou., 2018. Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*, [e-journal] Available at: <https://www.tandfonline.com/doi/full/ 10.1080/17517575.2018.1493145 > [Accessed 24 April 2019]

Huck, N., Afawubo, K., 2015. Pairs trading and selection methods: is cointegration superior? Applied Economics 47 (6), pp.599-613.

Heaton, J., Polson, N. and Witte, J. (2017). Deep learning foF finance: deep portfolios. Applied Stochastic Models in Business and Industry. 33, pp.3-12.

Heer, J., Card, S. K., & Landay, J. A. (2005, April). Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 421-430). ACM.

Hemanth Kumar, P., & Basavaraj Patil, S. (2018). Forecasting volatility trend of INR USD currency pair with deep learning LSTM techniques. 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). doi:10.1109/csitss.2018.8768767

Hochreiter, S and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8), pp.1735–1780

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research.* 207(3), pp.1702–1716.

Huck, N. And Afawubo, K. (2015). Pairs trading and selection methods: is cointegration superior? *Applied Economics*. 47(6), pp.599–613.

Investopedia (2017). *Cyclical Industry.* [online] Available at: <https://www.investopedia.com/terms/c/cyclical_industry.asp > [Accessed 24 April 2019].

Kwon, C. S., & Shin, T. S. (1999). Cointegration and causality between macroeconomic variables and stock market returns. *Global Finance Journal*, *10*(1), 71-81.

Keim, D., Andrienko, G., Fekete, J. D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In *Information visualization* (pp. 154-175). Springer, Berlin, Heidelberg.

Krauss, C. (2017). Statistical Arbitrage Pairs Trading Strategies: Review and Outlook. *Journal of Economic Surveys*, 31(2), pp.513–545.

Liew, R. and Wu, Yu. (2013). Pairs trading: A copula approach. *Journal of Derivatives & Hedge Funds,* 19(1), pp.12–30.

Ma, Y.; Han, R. Research on stock trading strategy based on deep neural network. In Proceedings of the 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, Korea, 17–20 October 2018; pp. 92–96.

MacKinnon, J. G. (1996). Numerical distribution functions for unit root and cointegration tests. Journal of applied econometrics, 11(6), 601-618.

Najafabadi, S. (2009). *Prediction of Stock Market Indices using Machine Learning*. Master thesis. McGill University.

Novak, M. and Veluscek, D. (2016). Prediction of stock price movement based on daily high prices. *Quantitative Finance,* 16(5), pp. 793–826

Nóbrega, J. and Oliveira, A. (2014). A Combination Forecasting Model Using Machine Learning and Kalman Filter for Statistical Arbitrage. In: IEEE, *2014 IEEE International Conference on Systems, Man, and Cybernetics.* San Diego, CA, USA, October 5-8, 2014. IEEE.

Rad, H., Low, R. and Faff, R. (2016). The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance,* 16(10), pp.1541-1558.

Rundo F. (2019) Deep LSTM with Reinforcement Learning Layer for Financial Trend Prediction in FX High Frequency Trading Systems. Applied Sciences. 9(20), pp.4460.

Santoso, M., Sutjiadi, R., & Lim, R. (2018). Indonesian Stock Prediction using Support Vector Machine (SVM). In MATEC Web of Conferences (Vol. 164, p. 01031). EDP Sciences.

Tay, F. and Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega,* 29(4), pp. 309–317.

Usmani, M., Adil, S., Raza, K. and Ali, S. (2016). Stock market prediction using machine learning techniques. In: Department of Computer Science, Iqra University, Karachi,

Pakistan,*2016 3rd International Conference On Computer And Information Sciences (ICCOINS)*. Kuala Lumpur, Malaysia, 15-17 Aug. 2016. IEEE.

Wang, C., Han, D., Liu, Q. and Luo, S. (2019). A Deep Learning Approach for Credit Scoring of Peer-to-Peer Lending Using Attention Mechanism LSTM. *IEEE Access 7,* pp.2161-2168.

Wen, D., Ma, C., Wang, G. and Wang, S. (2018). Investigating the features of pairs trading strategy: A network perspective on the Chinese stock market. *Physica A: Statistical Mechanics and its Applications,* 505, pp. 903–918.

Zeng, Z. and C, Lee. (2014). Pairs trading: optimal thresholds and profitability. *Quantitative Finance,* 14, pp.1881–1893.

~~Zacks Investment Research (2019). *6 Characteristics of Stock Markets.* [online] Available at: <https://finance.zacks.com/6-characteristics-stock-markets-2511.html> [Accessed 24 April 2019].~~

Engle, R. F., & Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. Econometrica: journal of the Econometric Society, 55(2), 251-276.

Vidyamurthy, G. (2004). Pairs Trading: quantitative methods and analysis (Vol. 217). John Wiley & Sons.

Elliott, R. J., Van Der Hoek, J., & Malcolm, W. P. (2005). Pairs trading. Quantitative Finance, 5(3), 271–276. doi:10.1080/14697680500149370.

Garg, A. (2012) 'Forecasting exchange rates using machine learning models with time-varying volatility', Master Thesis, [online] Available at: < https://www.diva-portal.org/smash/get/diva2:538398/FULLTEXT01.pdf> [Accessed 16 Nov 2020].