A Nonparametric Approach to Option Pricing via Machine Learning Algorithms Based on Visual Environment of Web sites

September 2016

MSc Computer Science
Abstract

Aiming at financial derivatives pricing issues, I propose several machine learning approaches such as Support Vector Machine (SVM), Back-propagation Neural Network (BP), Radial Basis Function Neural Network (RBF) and Genetic Algorithm (GA) can be adopted to implement optimization problems about financial issues. All of them are nonparametric representation to estimate the derivative assets pricing. In consideration of the much better accuracy and more computational efficiency by using the artificial intelligence pricing methods, especially when the price dynamics of underlying assets are not known, and when under the non-arbitrage condition, the common pricing formulas cannot search an analytical solution. With the purpose of making this issue materialization and pertinence, I will focus on one of the derivatives pricing -- option pricing and choose to calculate and simulate the Black-Scholes (BS) model in order to evaluate the potential value of the above three pricing algorithms. For instance, a fixed length of period daily option prices can be acted as training sets and the mentioned algorithms on the basis of the Black-Scholes model produce the results which can be used on option pricing successfully. For comparison, I intend to employ these three prevailing and practical methods to simulate the BS model and estimate the results respectively. The three methods -- support vector machine (SVM), as well as two kinds of neural networks -- back-propagation networks (BP) and radial basis function networks (RBF) will be employed in this project.

Currently, the web technology has been spring up drastically as a significant means so as to make the financial issues visualized to emerge and easy to operate by ordinary users. With the assistance of machine learning, data mining and artificial intelligence, the acquisition of financial information and the prediction of dynamic market can be visually presented on the screen. As web based technology continue to be favorably chased after, a lot of investors choose to complete their investments and transactions through the internet. Probably, observing stocks movements and dealing derivatives trade may not be difficult, however, how to extract effective information, analyze and optimize it and then offer some suggestion automatically by our machines, which still requires some complicated procedures to accomplish. It is impossible for every investor to understand the optimized theory and processes, therefore, providing users with a visual interface like a website to observe the results supplied by machine and select their operations easily will be a crucial solution. While the data straightly obtained via the internet is almost disorderly and unsystematically. As a result, these unstructured data it is necessary to introduce other techniques such as machine learning or data mining to explore the correlations between data and solve this challenging problem.
Table of Contents

1. Introduction ....................................................................................................................... 1
   1.1 Project Aims .................................................................................................................. 1
   1.2 Project Objectives ....................................................................................................... 2
   1.3 Report Structure ........................................................................................................ 2
2. Background Research ....................................................................................................... 4
   2.1 Literature Review ......................................................................................................... 4
      2.1.1 Conventional option pricing modelling and forecasting: ..................................... 4
      2.1.2 Nonparametric option pricing approaches and forecasting: ............................... 6
   2.2 Technologies and Tools .............................................................................................. 11
3. Project Management and Analysis .................................................................................. 12
   3.1 Schedule and Methodology ....................................................................................... 12
   3.2 Development Methodology ....................................................................................... 13
      3.2.1 Improving option pricing forecasting by refining conventional methods: .......... 13
      3.2.2 Improving forecasting precision based on nonparametric methods: ................. 13
   3.3 Requirements Analysis ............................................................................................... 15
4. Design .............................................................................................................................. 17
   4.1 Architectural Design .................................................................................................... 17
   4.2 Detail Design ............................................................................................................... 17
   4.3 User Interface Design ............................................................................................... 22
   4.4 Web Application ......................................................................................................... 26
5. Implementation .................................................................................................................. 28
   5.1 Environment Setup ...................................................................................................... 28
   5.2 Basic Code Structure ................................................................................................. 28
6. Testing and Evaluation ..................................................................................................... 31
   6.1 Experimental Results and Evaluations ...................................................................... 31
   6.2 Comparative Results ................................................................................................. 39
6.4 Critical Evaluation ........................................................................................................................................... 43

7. Conclusion .......................................................................................................................................................... 45

7.1 Future Work .................................................................................................................................................... 45

7.2 Critical Reflection ........................................................................................................................................... 46

8. References ......................................................................................................................................................... 47
List of Figures

Figure 1: ..................................................................................................................6
Figure 2: ..................................................................................................................8
Figure 3: .................................................................................................................11
Figure 4: ................................................................................................................12
Figure 5: .................................................................................................................14
Figure 6: .................................................................................................................19
Figure 7: .................................................................................................................19
Figure 8: ................................................................................................................31
Figure 9: ................................................................................................................32
Figure 10: ..............................................................................................................33
Figure 11: ..............................................................................................................33
Figure 12: ..............................................................................................................34
Figure 13: ..............................................................................................................37
Figure 14: ..............................................................................................................37
Figure 15: ..............................................................................................................38
Figure 16: ..............................................................................................................38
Figure 17: ..............................................................................................................39
Figure 18: ..............................................................................................................39
Figure 19: ..............................................................................................................40
Figure 20: ..............................................................................................................40
Figure 21: ..............................................................................................................40
Figure 22: ..............................................................................................................41
Figure 23: ..............................................................................................................41
Figure 24: ..............................................................................................................42
Figure 25: ..............................................................................................................42
Figure 26: ..............................................................................................................42
Figure 27: ..............................................................................................................43
Figure 28: ..............................................................................................................43
Figure 29: ..............................................................................................................43
Figure 30: ..............................................................................................................43
List of tables

Table 1: ..........................................................................................................................32
Table 2: ..........................................................................................................................32
Table 3: ..........................................................................................................................35
Table 4: ..........................................................................................................................36
Table 5: ..........................................................................................................................36
Table 6: ..........................................................................................................................37
Table 7: ..........................................................................................................................38
Table 8: ..........................................................................................................................38
1. Introduction

1.1 Project Aims

Option is a nonlinear financial instrument and it has a high transaction risk. Owing to its characteristics of nonlinear prices and high risk, it is significant for investors to have an accurate forecasting on stock prices as much as possible to hedge their risks. One aim of using several algorithms such as support vector machine, back-propagation neural networks and radial basis function is to ascertain whether these approaches can be used for option pricing forecasting. The above statement has been proven in a good deal of papers and experiments that it is feasible. The other aim is whether we can come up with some measures to improve the performance of Black-Scholes model and how can we make improvements to overcome the shortcomings of BS and in what aspects to accomplish them.

Ideally, after that, another aim is publishing the optimized results on a web tool. I partitioned my project into two portions: the first segment was using four mentioned approaches above (e.g. BS, SVM, BP, RBF) to complete simulations and predictions; the second segment was establishing a website to illustrate the optimized results, analyze performance and giving advice. This project not only demonstrated the effect (the forestage part), but also explained how it worked (the backstage part) and why it happened.

On the other hand, HTML-based interface technology enables the users throughout the world to easily utilize web, however, it is difficult for them to comprise integrated tools of both visiting the web sites and optimize their financial derivatives. In this project, I desire a framework where users can functionally combine their operation of financial products with visual components on the Web. The web interface allows users to create a visual component for web sites by employing the techniques of HTML and JavaScript. Users can directly manipulate both web sites and financial products’ tools on their own desktops to define linkages among them. There are no requirements of programming expertise to combine them together functionally, users can connect the two software and build this integrated tool all by themselves.
1.2 Project Objectives

Since the algorithms such as SVM, BP and RBF are powerful and flexible to approximate complicated nonlinear relations, an effective application of which is to implement option pricing whose formulas are extremely nonlinear. Accordingly, whether I can estimate the Black-Scholes formula nonparametrically and achieve a sufficient accuracy to make it used practically, which could be a challenge. I incline to solve this problem for performing Black-Scholes model simulation experiments by using the three approaches referred to above respectively, where different optimizing algorithms are trained to simulate the option prices of BS model and produce the results intelligently, and then they are compared to the Black-Scholes formula to see how close they simulate. After that, by means of comparing with the results of different algorithms depending on various types of stocks or options, I hope I can get some interesting or useful conclusions, moreover supply some advice to users.

The main information resources in the World Wide Web are available on handmade HTML pages, such as those using HTML (Hyper Text Markup Language), JS (JavaScript) and PHP (PHP: Hypertext Preprocessor). Web sites are the application programs that have an HTML-based tool for users to employ some services provided by a remote network service. Many companies provide web sites, such as search engines, financial analysis services, calculation tools, and various other kinds of services of interface and database. Actually, every Internet user can access to these web applications and use many services supplied by internet nowadays, therefore it is a prevailing trend to execute his transaction online. My objective is to support users to be involved in intellectual activities using the web tool and let target users to be specialists in their adept domain such as finance but beginners in computer and web programming. As a consequence, an integrated tool connecting financial products’ tools to web tools make it easier for users to employ the different software to accomplish their goals.

1.3 Report Structure

For reasons of clarity, the following chapters will provide further information of how the aims and objectives of the project will be fulfilled by correlative implemental results, performance analysis and visualized web tools. It starts with the chapter 2 background research that comprises a literature review about the algorithmic theory and optimization which focuses on financial model and enables web technology to build a web tool to demonstrate relevant results. The chapter 3 project management and analysis incorporates the whole project management activities such as schedule and development methodology as well as requirements and risk analysis. Subsequently, an elaborate account of project design which contains the architecture
and detail design will be presented. Furthermore, the thoughts and procedures of the user interface design will be expounded and corresponding web application will be formed as well in chapter 4 design. Afterwards, the chapter 5 describes implementation mainly based on environment setup as well as general code structure and function libraries in two aspects. The chapter 6 explains how to use the experimental results and strategies to test effects and evaluates it from comparative and critical views. Eventually, according to the research and experiment above, there are several conclusions can be drawn, moreover, the future work including existent limitation and possible improvements may be given in the last chapter.
2. Background Research

2.1 Literature Review

2.1.1 Conventional option pricing modelling and forecasting:

Stocks and options are both important financial instruments and they highly affect humans’ activity of investment. Unfortunately, in essence, they fill with volatility, uncertainty, nonparametric and nonlinear properties. Therefore, how to forecast the prices precisely so as to fulfil the investors’ expectations according to the movement of time series that is not only extremely challenging but also remarkably significant. Derivatives are important financial instruments whose values are called underlying assets which are descended from the basic asset’s price. underlying assets can be construed as shares, stock market indices, currencies, credits, et cetera (Hull, 1997). Primarily, anything which has an unpredictable influence on any economic activity can be regarded as a certain derivative’s underlying asset. Ascertaining an option’s theoretical price or pricing an option is perceived as one decisive aspect in financial research. There are a large amount of parametric and nonparametric option pricing schemes having been presented and applied to this area. Not only the aim of option pricing is to simulate the past and current price so as to help investors make decisions, but also is the forecasting activity expected to predict the price of future relatively accurately without knowing the situation of underlying asset in advance.

Black-Scholes model:

Black and Scholes (Fischer Black and Myron Scholes, 1973) instituted the option pricing mechanism of European call and put option and established a famous formula which was a second order partial differential equation. The proposed solution and formulation in their studies assumed that the underlying stock price abides by geometric Brownian motion with constant volatility (Shian-Chang Huang, 2007).

\[ \frac{dS}{S} = \mu dt + \sigma dW_t \]

Where \( S \) is the price of stock, \( \mu \) is the expected return, namely the drift rate of \( S \), \( \sigma \) is the standard deviation of the return of stock, and \( W_t \) is the Brownian motion. Black and Scholes figured out the well-known second order partial differential equation which manipulates the prices of a European call or put option.
\[
\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2S^2\frac{\partial^2 f}{\partial S^2} = rf
\]

where \( r \) is the riskless interest rate. The solutions for the call and put option price are:

\[
C = SN(d_1) - Ke^{-r\tau}N(d_2),
\]
\[
P = -SN(-d_1) + Ke^{-r\tau}N(-d_2)
\]

where \( C \) is the call option price, \( P \) is the put option price, \( K \) is the strike price, and \( \tau = T - t \) is the maturity. Parameters \( d_1 \) and \( d_2 \) are as follows:

\[
d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}}
\]
\[
d_2 = d_1 - \sigma\sqrt{\tau}.
\]

These equations above can be viewed as state-space form:

\[
C_t, P_t = F(S_t, K, T - t, r(t), \sigma(t)) + \epsilon_t
\]
\[
r(t) = r(t - 1) + \zeta_t,
\]
\[
\sigma(t) = \sigma(t - 1) + \eta_t,
\]

where \( r, \sigma \) represents the hidden states. The observations of \( C, P \) are the nonlinear function \( S_t, K, T - t, r(t), \sigma(t) \), while \( \tau, K \) are considered as inputs. The key and only unknown parameter is \( \sigma \) in the formulas.

Option price = \( BS(S, K, T, \sigma, r) \)

In the Black-Scholes model, all these variables can be easily obtained apart from the volatility \( \sigma \) which is usually assumed invariable when we make a prediction on option pricing. Because of the property of uncertainty of the volatility \( \sigma \), the forecasting accuracy depends on the calculation of \( \sigma \). Basically, there are two approaches to estimate \( \sigma \): historical and implied volatility. Historical volatility is defined for the next period estimation. Assuming \( N \) is trading days and \( \sigma \) can be estimated as follows:

\[
\sigma = s\sqrt{N}
\]

Where \( s \) is the standard deviation of historical daily returns \( \log (S_t/S_{t-1}) \) and the data of a month and two months are often used to represent historical volatility. A drawback of historical volatility assumes that the future market will be unchangeable, which may not be realistic, as a consequence, this assumption is valid nothing but the historical volatility \( \sigma \) is constant. In contrast, the calculation of implied volatility is much more complicated than historical volatility. The implied volatility can be estimated by using Black-Scholes formula in reverse manner, under certain conditions, it may reflect some traits about future returns. However, it lacks relevant theoretical support and practical proof as well.
2.1.2 Nonparametric option pricing approaches and forecasting:

Recently, nonparametric approaches gain more popularity such as neural networks (NN) and support vector machine (SVM) for option pricing, both of which are used to nonlinear patterns from time series. There is a popular belief that the nonparametric approaches perform better than parametric approaches on the basis of recent researches, especially in financial and economic models, which can efficiently and effectively help us make wise decision. The relevant researchers spent several years and tested various different mechanisms (linear and nonlinear) combining with different methods, in addition, discovered SVM and NN frequently outperformed than other methods (e.g. Regression, Bayes and so on) and often presented preferable experimental results.

Support vector machine (SVM):

Within the recent past, the support vector machine (SVM) method, other than neural networks, which achieves great popularity and is one of the most advanced techniques for regression and classification. Since SVM possesses outstanding performance of generalization on sparse model, it is believed that SVM has the properties of structural risk minimization principle. Furthermore, in the other important aspect, it exhibits essential achievement in practical applications. Support vector machine (SVM) proposed by Vapnik (1995) develops a way it maps input data into high dimensional space in order to reproduce kernel space in which a linear regression and classification can be performed. SVM strives to realize the goal that it will minimize the upper bound of generalization error, which is based on the structural risk minimization principle. Moreover, by mapping the dataset to high dimensional space, SVM can produce the linear regression function:
\[ y = w \phi(x) + b \]

where \( \phi(x) \) is the feature which is nonlinear mapped from input space \( x \) to higher space. The coefficients \( w \) and \( b \) are estimated by minimizing formula:

\[
R(C) = C \frac{1}{N} \sum_{i=1}^{N} L_\varepsilon(d_i, y_i) + \frac{1}{2} ||w||^2
\]

where

\[
L_\varepsilon(d, y) = \begin{cases} 
|d - y| - \varepsilon & |d - y| \geq \varepsilon, \\
0 & \text{otherwise},
\end{cases}
\]

where both \( C \) and \( \varepsilon \) are specified parameters. The term \( L_\varepsilon(d,y) \) is the \( \varepsilon \)-intensive loss function. The \( d_i \) is the actual option price in the \( i \)th period. This function signifies that errors below \( \varepsilon \) need not to be the penalty. The term \( \frac{C}{N} \sum_{i=1}^{N} L_\varepsilon(d_i, y_i) \) is the empirical error and the term \( \frac{1}{2} ||w||^2 \) is the smoothness of the function. \( C \) is the trade-off between empirical risk and the smoothness of model. \( \xi \) and \( \xi^* \) indicates two positive slack variables which signify the distance from actual value to corresponding boundary value of \( \varepsilon \).

\[
\min_{w,b,\xi,\xi^*} R(w,\xi,\xi^*) = \frac{1}{2} w^T w + C \left( \sum_{i=1}^{N} (\xi_i + \xi_i^*) \right)
\]

Subject to:

\[
w \phi(x_i) + b - d_i \leq \varepsilon + \xi_i^* \\
d_i - w \phi(x_i) - b \leq \varepsilon + \xi_i \\
\xi_i, \xi_i^* \geq 0.
\]

After taking the Lagrangian conditions, the following quadratic programming problem can be achieved:

\[
\max_{\alpha,\alpha^*} R_D(x_i, \alpha^*) = \sum_{i=1}^{N} d_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \\
- \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j)
\]

With the constraints:

\[
\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \\
0 \leq \alpha_i \leq C, \\
0 \leq \alpha_i^* \leq C, \\
i = 1, \ldots, N.
\]
\(\alpha_i\) and \(\alpha_i^*\) are the Lagrangian multipliers. The quadratic problem can be represented as follows:

\[
y = f(x, x', x'') = \sum_{i=1}^{N} (x_i - x_i^*)K(x_i, x_i') + b
\]

Where \(K(x, x_i)\) is the kernel function and \(\alpha, \alpha_i^*\) are the solutions to this quadratic problem. The optimization problem is translated to a quadratic problem through the Lagrangian multiplier and the Karush-Kuhn Tucker (KKT) conditions. The quadratic programming problem with constraints which I employ in this model is only appropriate for a linear kernel function, but differs from other kernel functions such as Gaussian kernel function. The kernel function is equal to inner product of this two vectors \(x_i, x_j\):

\[
K(x_i, x_j) = \Phi(x_i) \Phi(x_j).
\]

The linear kernel function is inclined to perform pretty well under general smoothness assumptions.

\(C\) indicates the tolerated derivations from \(\varepsilon\), which can be acted as a measure whether the function is overfitting (it fits too well on its training points). Furthermore, if the function \(f(x)\) fits its training data too well, this implies that it will not forecast much accurately on the future data which cannot be seen by the training function. It is also approved that the data lie outside the given boundary \(\varepsilon\) or the new data are added artificially, which will not be punished sufficiently so that the results are so close to the simulations and deviate the predictions. It can be sketched as the following graph:

![Fig 2](image)

The right graph of Fig 2 is to illustrate the data lie outside the boundary how to be penalized by \(\xi_i\) shown on the left. On the other hand, the data lie out of the boundary either above or below, no matter how many there are, all of them will be penalized and have a less influence over the determination of the function. On the contrary, the data lie inside the boundary will not be punished at all, this means that the positive slack variables (\(\xi_i\) and \(\xi_i^*\)) are set as 0 and will play an important role in the contribution of the function \(f(x)\).

**Back-propagation neural networks (BP):**
Back-propagation neural networks are prevailing and efficient computational technique and they provide an innovative way to explore dynamic financial products. Neural networks are skilled in regression and simulating trends, sequentially making predictions because they can model a structure and relationship between inputs and outputs even if there are some noisy data inside. The reason why Black-Scholes model is capable of capturing dynamic stock prices is based on grasping the formula assumption and key parameters, while neural networks only reveal the relationship of related information from historical data. Hutchinson et al.’s experiments manifested clearly that neural networks were able to recover the conventional models such as Black-Scholes formula, moreover, they were more computationally accurate and efficient when the prices were fluctuant and uncertain. What is more, neural networks can use the past volatilities and other financial market factors to forecast future volatility, which allow investors to develop their own strategies on the basis of individual demand. Compared with the historical and implied volatility calculated by Black-Scholes model, the volatility derived from neural networks was superior. By comparison, the latter was preferred.

As a popular and famous supervised learning method, back-propagation is used to make predictions and classifications, and it has been emerged as the form of multilayer networks. BP neural network means the network where errors back propagate, which consists of an input layer, one or more hidden layers and an output layer, in addition, each layer is composed of a certain number of neurons. I chose three parameters ($S/K$, $T$, $\sigma$) as inputs which were transmitted via input layer to hidden layers, and the inputs were executed complex calculations in the hidden layer (Matlab library), then obtained the results from output layer. If there were deviations between the expectation of output and actual outcome, the deviations should be back propagated from output layer to input layer via hidden layers. In this process, the deviations were distributed to all the neurons of each layer by gradient descent, by virtue of the error values to correct the weights of each unit, so as to redistribute the weights of the whole network. After this process, the inputs were transmitted from input layer to output layer and the process proceeded back and forth until the deviations were reduced within the rational or acceptable range, or reached the learning times I set in advance. The process of weights adjustment is the process of training the learning network. The deviations between forecasting output values and actual values needed the back propagated process through the whole networks to update the weight of each unit. It was the measure that tried minimizing the errors between actual values and forecasting values, and the output value for a unit $j$ of this procedure can be formed as follows:

$$O_j = G\left(\sum_{i=1}^{m} w_{ij} x_i - \theta_j\right)$$

Where $x_i$ is the predicted value of $i$th unit, $w_{ij}$ is the weight of $i$th unit, $\theta_j$ is the threshold, and $m$ is the amount of units in preceding layer. The function $G()$ is a widely
used sigmoid function in backpropagation neural networks.

\[
G(z) = \tanh(z) = \frac{1 - e^{-z}}{1 + e^{-z}}
\]

Theoretically, BP can simulate almost all types of data as long as the sufficient training has been supplied. Especially, it is good at dealing with nonlinear patterns by learning from historical data. When we employed BP as a simulator of BS and then as an option pricing forecasting tool, one of the key factors was to estimate the volatility, which would determine the effectiveness of algorithmic performance.

**Radial Basis Function neural networks (RBF):**

Because the financial market is becoming more and more complicated, promptly updated information about financial derivatives and precise computer simulation and prediction upon them become a huge demand. Radial basis function is used for solving and optimizing Black-Scholes equation and I choose the European option as an indicator of RBF algorithm which works well and usefully, since it has the ability to perform a more complicated task of approximation, which has been found recently. A typical RBF formula can be formed:

\[
u(\bar{x}) = \sum_{j=1}^{N} \lambda_j \phi(\varepsilon \| \bar{x} - \bar{x}_j \|)
\]

\(\phi(x)\) is the radial basis function and \(\bar{x}_j, j = 1, \ldots, N\) are center points. \(\varepsilon\) is a significant shape parameter and it has an important effect on the accuracy of RBF approach. RBF neural network is a three-layer forward propagation network with the capability of approximating arbitrary nonlinear functions. That means there are three sets of parameters can be adjusted: the basis function center of hidden layers, variance and the weights from hidden units to output units. These parameters can be determined in two forms: on the one hand, we choose center and variance based on clustering and empiricism. After that, since the output is the linear unit, of which the weight can be calculated by iterative least square method. On the other hand, we can use supervised learning to train samples and correct errors and update the three parameters step by step, in other words, calculate the total output error gradient for each parameter and update the pending learning parameters by using gradient descend method.
2.2 Technologies and Tools

A significant part of my project is to identify and choose suitable programming tools and technologies to implement my experiments thus obtain the correct results. In addition, since different technologies or tools have different strengths and weakness, good choices will lead to decrease difficulties or complexities, and improve performances accordingly. With a view to the scope of my project, the following technologies and tools I will adopt:

Development tool
- Matlab
- Sublime Text
- Excel databases

Web development framework
- Apache network environment

Web interface
- HTML: Hypertext Markup Language
- JavaScript: object-oriented client-side programming language for web application
3. Project Management and Analysis

3.1 Schedule and Methodology

Fig 4 illustrates the schedule and duration about my project and the gantt chart generally introduces the activities I will be involved in.

![Fig 4 schedule gantt chart](image)

Phase 1. Background analysis in two weeks’ duration
The project starts with nearly 14 days’ duration of introduction and background research, which concentrates on the topic, aims and objectives that contain machine learning algorithms I will adopt and the web application I want to design.

Phase 2. Implementations of three algorithms (SVM, BP network and RBF network) in three weeks’ duration
The first part of implementation is to develop the algorithms mentioned above and optimize the results of Black-Scholes model.

Phase 3. Comparison and evaluation in 5 days’ duration
According to the results generated by the different approaches, compare them each other and evaluate the advantages and disadvantages respectively, in addition, try to give some interesting advice.
Phase 4. Web application design and implementation in three weeks’ duration
The second part of implementation is to design the interface and functions of web tool, for instance, the requirements of how to connect to the results that Matlab produced and how to make users easier to operate the website and check the outcomes distinctly. Moreover, focus on the technologies, tools and methodologies, such as the language I will use and the network environment will be configured.

Phase 5. Performance measures and results analysis in three weeks’ duration
Make use of different measures to test and evaluate the performance of simulation, prediction and web tool, clarify the functions and purposes of the integrated tool, meanwhile analyze the merits and drawbacks.

Phase 6. Conclusion in a week’s duration
In the end, I will summary the outcomes which I achieved and draw a conclusion about my project. Furthermore, I will illustrate the unaccomplished functions or parts and the future work need to be done, reflect the gains and deficiencies.

3.2 Development Methodology

3.2.1 Improving option pricing forecasting by refining conventional methods:

The Black-Scholes method was aimed at demonstrating the theoretical price of option under the condition of given the current underlying asset price and relevant parameters, because of this, in order to conform with market discipline, it is necessary for enhancing the precision of option pricing to do some modifications. The following notations are used: $S$ is the underlying asset price, $K$ is the option’s exercise price, $\sigma$ is the asset price volatility, $r$ is the current risk-free interest rate, $\Delta t$ is the time interval (length of a time period), $T$ is the expiration time, $C$ is the option price, and $\hat{C}$ is the forecast option price.

$$\hat{C} = e^{r(T)\Delta t} C.$$ 

3.2.2 Improving forecasting precision based on nonparametric methods:

I primarily study the issue of option pricing and forecasting on the strength of three nonparametric approaches: support vector machine (SVM) and two neural networks: radial basis function (RBF) and Back-propagation (BP).
Traditional option pricing methods chiefly depend on the relevant variables of formulas which only follow relatively the ideal and stationary option market; in other words, it is valid when economic rules are unchangeable and similar or the period is very short. Once the option market is dynamically changing, the idealized models will not generate rational and practical results. On the other hand, the selection of span of time is an essential issue in the financial forecasting market, since option pricing is probably influenced by the historical data; what is more, the recent historical data will reflect the correct trend much more accurately. As a result, the empirical financial studies indicate that the sliding window of time length $L$ should not be very large when we make predictions. If the sliding window is too large, it means the longer data will produce larger errors sequentially result in inaccurate prediction due to capturing incorrect trend of market. For this reason, I employ the SVM and NNs preferably resolving the two problems: firstly, they can learn the market rules dynamically; secondly, an appropriate window size $L$ can present a periodic feedback in SVM and NNs architectures, as a consequence, achieve the price forecasting’s aim by making use of option’s historical prices.

The steps are as follows:

Step 1 (training): utilizing SVM or NNs to train the extracted data within the sliding window length $L$.

Step 2 (forecasting): owing to the economic market is dynamic, when the new data (the real price of current day) come in, adopt them and discard the data of earlier day, then shift the window forward and train the algorithm on these updated data again.

![Fig 5 training and forecasting window](image-url)
3.3 Requirements Analysis

(a) Generally, a direct way was to train a SVM or NNs to estimate the option pricing formula (BS) using the historical data, in comparison, using the same methodology with adjusting related parameters on the same data of stock market, the researchers drew the conclusions that the parametric approaches with SVM and NNs were superior to the pure BS in a variety of experimental cases.

(b) The model-reference architecture used SVM or NNs framework and simultaneously derived the strength of conventional parametric approaches to forecast the differences between the predictable results and real observed option prices. The first step of this architecture employed the traditional parametric models for simulation track, next used SVM or NNs for volatility inference. The experimental results manifested the similar consequence that the algorithmic mechanisms usually generated better outcomes than conventional models likewise. Meanwhile, the nonlinear residuals between actual observed option prices and the predictions of parametric methods with SVM or NNs can be captured.

(c) The cascade architecture took the parametric models as the preprocessing procedure and merely implemented SVM or NNs on the obtained results of traditional models. The purpose is that in accordance with known outcomes, it can sharply alleviate the burdens of training process and complexity of algorithms. As a consequence, the algorithms can only concentrate on capturing the residuals and take advantage of their own merits to diminish the pricing errors, furthermore, to reach the goal of the improvement of pricing precision.

Traditional models are only for the purpose of pricing options, in order to improve the performance of conventional methods, I plan to incorporate the function of prediction to the pricing structure, in addition, I intend to extend the parametric approaches to nonparametric approaches including SVM and NNs algorithms used in. apparently, option pricing mechanism equipped with forecasting function of artificial intelligence, which accords with the demand of real market more efficiently. Since the phase of pricing is primarily designed to provide reasonable prices in accordance with the given variables and formulas at present, the most intractable part is anticipant activity. Because of which, the anticipant errors should be larger than the calculations’. Accordingly, the predictions for future would generate much more inescapable indeterminacies.

For the sake of the shrinkage of forecasting errors, several artificial intelligent methods are recommended and focus on reducing these errors, further refine the conventional methods. In the simulation stage, the pricing formulas can explicitly instruct the algorithms trace a similar trend of price movements and consequently release the
pressure of algorithms and let them simply concentrate on reducing errors and improving precisions. It has been proved that this methodology is really worked in practice.
4. Design

4.1 Architectural Design

An effective web tool can extract the data from target databases which can be visible on Hypertext Markup Language (HTML) pages, and this is a process that transfers databases to visualized graphs. HTML forms, JavaScript client side, compatible and portable databases constitute a comprehensive data extraction process. Extracting organized and structured data from databases and mapping them to the web sites, which can be regarded as a useful and efficient way to apply to stock market and business intelligence. A majority of information about them is viewed with a browser in the form of HTML, moreover, in consideration of the function of HTML format for presentation purpose, I try to design an interface to publish the existing outcomes on the website or extract data and draw them as visible graphs automatically. The function of recognizing and extracting different types of documents and then producing and publishing the visible outcomes on the web can be implemented by JavaScript Language. In my project, Apache web server is used to construct the network environment.

4.2 Detail Design

The traditional measurement methods of forecasting accuracy are mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percent error (MAPE) and their expression are referred as follows:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (d_i - \hat{d}_i)^2,$$

$$\text{RMSE} = \left( \frac{1}{N} \sum_{i=1}^{N} (d_i - \hat{d}_i)^2 \right)^{1/2},$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |d_i - \hat{d}_i|,$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{d_i - \hat{d}_i}{d_i} \right|.$$

Where $N$ indicates the period of forecasting, $d_i$ is the normalized actual option price at period $t$ ($d_i = C_i/K$ or $d_i = P_i/K$; $C_i$ and $P_i$ express the call option price and put option price
respectively), \( \hat{d}_i \) is the predicted price at period \( t \). In this study, I primarily focused on two of the measurements: mean square error (MSE) and mean absolute error (MAE).

a). Firstly, I stimulated the standard Black-Scholes model. Secondly, when I implemented the Support Vector Machine, Radial Basis Function and Back-Propagation are adopted as the algorithms to complete training, testing and forecasting by using the Matlab software.

b). The SVM algorithm whose training process aims at minimizing the quadratic error of objective function, thereby to find the optimal solution. It is worth noting that in the following implementation I will use the constrained optimization problem is suitable for linear kernel function, but different from other kernel functions like Gaussian. Because of convex programming problem, there exists global optimal solution; relatively fast training speed and sparse representation, which are three distinguished advantages of SVM.

In order to improve the performance of SVM, how to find the best possible values for capacity and e-insensitivity \( \epsilon \) in an optimized way. In consideration of the extremely similar training results in the training process, this implied that I can spend less computational time on training points to find the best possible values for capacity and e-insensitivity \( \epsilon \). And the outcomes testified that it was feasible to find the best values of them using less training points under the condition of very similar training results. That means even though I adopted the less training points to gain the best values, I could also use SVM with more training data. Nevertheless, there is a problem in actual operation. If you add or delete some data from the already trained SVM, the existing best possible values of capacity and e-insensitivity \( \epsilon \) may change, especially deleting the data. If you want add or delete some data, you had better retrain the SVM and observe the results. In general, add more data will not influence the optimized value of capacity and e-insensitivity \( \epsilon \), whereas, delete excessive data will lead to the changes of both optimized values. Using less training data to obtain the approximate optimized value of capacity and e-insensitivity \( \epsilon \) of more training data, which does not mean this can be done all the time. The more inputs in the training process, the better results will be produced. The reason why this strategy can be employed, because we consider the computational complexity and it also depends on the extremely similar training results.

It is believed that since SVM can reflect the structural risk minimization principle mostly, it combines superb generalization performance with a sparse model representation. The advantages of SVM are:

1). The global minimum solution resulting from the minimization of a convex programming problem (Shian-Chang Huang, 2007).
2). It possesses a relatively fast training speed and excellent convergence.
3). The characteristic of sparseness embodying in solution representation.
c). Option pricing which is based on BP neural network adopts a principle of learning correction. With regard to the options I studied on, the variables of input and output need to be designed on the basis of types of data. Stock prices of time series are usually fluctuant and full of noises since the fact is the financial market is influenced by all kinds of factors. As a consequence, directly predicting stock prices in accordance with noisy data is often accompanied by a variety of errors. Back propagation neural network is perceived as an effective approach to forecast stock prices. In order to demonstrate the advantages of BP, it will be compared with other algorithms (e.g. RBF and SVM) by using real European stock data.
According to the BS formula, I used $S, K, T, \sigma, r$ as the inputs and took the price (e.g. call option) as the output to establish the neural network. The above four of five parameters can be easily achieved but volatility $\sigma$ is the only one which is unknown. Although volatility $\sigma$ may be obtained by the two means introduced in the chapter of Black-Scholes model, unfortunately, there is a complex problem that it cannot precisely reflect the real market so as to result in mispricing.

Thus instead of attempting to estimate the volatility $\sigma$, I ignored it and did not consider it as an input. While I threw it into a “black box” directly, namely the BP neural networks, in other words, it was taken as rather the modelling process of “rebuilding” Black-Scholes function than an input as the ready-made BS formula. Hence, instead of putting the pattern $(S, K, T, \sigma, r)$ in the back-propagation networks, I would establish the network with the pattern $(S, K, T, r)$. Since whether I kept the volatility $\sigma$ constant or adopted the approximated estimation for training, it would carry the same outcomes with BS formula. Because volatility $\sigma$ expresses the uncertainty of future price movements, the value of it is an important measurement to evaluate the precision of approaches. The historical volatility is the standard deviation of returns, and implied volatility can be also calculated by using some procedures of BS function in practice likewise. If both of them gain correct estimators, that means that BS formula is good enough and it is not a necessity to use other approaches to improve the performance, since the volatility $\sigma$ directly reflects the validation of assumption in BS model. However, the volatility $\sigma$ cannot be reflected in BP neural networks, because BP has not a specific function to calculate it and then it is not regarded as the input to establish the networks. In spite of this, it can be achieved eventually by BP.

For instance, a three layers BP neural network, when training the network, the back propagated process determines and adjusts the weight of each node in every layer. The weight of each node is allocated an initial value primarily. The deviations between actual and estimated values from output layer propagate back for updating the weights of each node. Because of this, BP goes round and round and then minimizes least-mean-square error between forecasting and real values. Theoretically, the neural network can generate satisfying outcomes as long as it gets sufficient training times, and it can simulate almost all kinds of patterns as well.

d). RBF network based on matlab library: At first, I chose newrb() to create neural networks which can be self-regulation based on particular case. The function newrb() was designed to approximate radial basis network and it was formed as follows:

```
net = newrb(P, T, GOAL, SPREAD)
```

where $P$ is output vector, $T$ is object vector, GOAL is mean square error and the default is 0.0, SPREAD is the expansion rate of radial basis function and the default is 1. There was a superiority while using newrb() to establish RBF network: newrb can increase the hidden neurons of RBF network automatically till mean square error is in a
reasonable range. Nevertheless, a drawback of RBF networks using matlab library will emerge larger deviations in later period, thereby results in the inaccuracy in prediction stage. That is in relation to inadequate sample data, so if the sample data is insufficient, RBF networks are deprecated.

After that, I employed the algorithm of obscuring k-mean clustering to ascertain the centers of each basis function and corresponding variances. In addition, the corrections of network weights were determined by local gradient descend method. The method processes as follows:

1. Determine the center of basis function \( c_i \) by obscuring k-mean clustering:
   (1). Stochastically choose \( h \) sample values as initialization of \( c_i \), \( i = 1,2,\ldots,h \), other samples are classified to the cluster which had the nearest Euclidean distance to center \( c_i \), thereby formed \( h \) subclass \( a_i \), \( i = 1,2,\ldots,h \);
   (2). Recalculate the value of center of each subclass \( c_i \):
   \[
   c_i = \frac{1}{s_i} \sum_{k=1}^{s_i} x_k
   \]
   \( x_k \in a_i \), \( s_k \) is the sample of subclass \( a_i \); meanwhile calculate to which cluster each sample will be subordinate.
   \[
   u_{ij} = \frac{\min \sum_{j=1}^{s_i} \left\| x_j - c_i \right\|}{\left\| x_k - c_i \right\|}
   \]
   \( x_j, x_k \in a_i \), \( U = \{ u_{ij} \in [0,1]; i = 1,2,\ldots,h; j = 1,2,\ldots,s \} \);
   (3). Inspect whether \( c_i \) is in an allowable error range.

2. Determine the width of basis function (variance \( \sigma \)):
   \[
   \sigma^2 = \frac{\sum_{j=1}^{s} u_{ij} \left\| x_j - c_i \right\|}{\sum_{j=1}^{s} u_{ij}}
   \]
   \( a_i \) is a subset with the center of \( c_i \).
   After the center of basis function and width been determined, hidden layer transformed in a fixed nonlinear way. The \( i \) hidden node output can be defined:
   \[
   b_i(x) = \frac{\exp \left[-\frac{\left\| x - c_i \right\|^2}{2\sigma_i^2}\right]}{\sum_{i=1}^{n} \exp \left[-\frac{\left\| x - c_i \right\|^2}{2\sigma_i^2}\right]}
   \]
3. Adjust the connected weights between hidden units and output units. The object function is:

$$E = \frac{1}{2N} \sum_{k}^{N} [y(x_k) - y(\hat{x}_k)]^2$$

That is general mean error function as well. $y(x_k)$ is the actual output of $x_k$, and $y(\hat{x}_k)$ is the expected output of $x_k$, moreover, $N$ is the total samples of training sets. The determination of parameters by using least square method can make RBF network approximate corresponding mapping relation, in other words, minimize $E$ by using gradient descend to correct the weights $\omega$ from hidden layers to output layer, which can be formed as follows:

$$\Delta \omega_i = -\eta \frac{\partial E}{\partial y(x_k)} \times \frac{\partial y(x_k)}{\partial \omega_i}$$

The adjustment of every step of weights $\omega$ can be decided by the formulas above ($\eta$ is the learning rate and its value domain ranges as decimals from 0 to 1).

$$\omega_i \leftarrow \omega_i + \Delta \omega_i, i = 1, 2, \cdots m$$

At last, the output of the whole network can be defined as:

$$y(x) = \sum_{i=1}^{m} \omega_i b_i(x) + \theta$$

$\theta$ is the threshold of output node.

### 4.3 User Interface Design

Firstly, as a private financial account, it is necessary for users to design a login interface to ensure assets secure, and it can be viewed as Pic 1.
Secondly, when login the personal homepage, it shows the quantity of stocks and options that a user holds. Users can look up their account information by using different modules at the left side. Besides, users can browse economic news and reviews, or stock index in different exchanges when they connect to relative servers. It can be shown in Pic 2.

When enter form module, users can look up the stocks or options they hold, they only need to input the name and number of certain stock, and after submission they can find the trend of it. It can be shown in Pic 3.
In the chart page, I choose two ways to demonstrate the results produced by *Matlab*. The first one to show the results is to create a connector by JavaScript which can upload the format of picture such as .png, .jpg and .jpeg. In this way, users can trace the performance of their own stocks in the past intuitionally, in addition, they also can acquire the predictions of their holdings in the future. In my project, I took the three days’ forecasting as an example to present the results of actual price and predicted prices by certain algorithm. It can be shown in Pic 4.

On the other hand, the other way to show the results is all by the web tool itself automatically. It made use of the datasets generated by *Matlab* and took the array of
real prices, simulation outcomes and forecasting outcomes as the input separately putting into the JavaScript document, after compiling the same results can be drawn on the web page. However, the drawback of this way is a little more complicated because user need to create an array of result and input them into the JavaScript document manually. Only in this way, the outcomes as follows can be presented (e.g. Pic 5 and Pic 6). It might be a little tough for the users who are not expert in computer.

Pic 5 chart module 2

Pic 6 chart module 3

The account page shows the whole stocks or options which the user hold. User can operate the trading by inputting the name or number to search certain stock, or choose any stocks using the boxes in front either. It can be shown in Pic 7.
4.4 Web Application

In order to simplify the operating complexity thereby improve the efficiency, I proposed a modified way to demonstrate the results visibly. When matlab programs produced the graphs meanwhile would generate corresponding data which can be made in the format of .csv or .xlsx. I tried to create a connector which can extract the data from databases straightforwardly, furthermore, the connector was established by JavaScript Language to recognize the format of .csv or .xlsx. It means that the web tool transfers the data to the graphs directly without doing anything else. The effects can be shown in Pic 8 and Pic 9.
5. Implementation

5.1 Environment Setup

The first step in my implementation process is setting up the environment. Required software and procedures are as follows:
- Matlab and objective data are needed to construct model and implement training and testing.
- After many repeated trials, obtain the optimized results and build databases to store them by excel.
- Configure Apache network environment, use HTML to design the interface and JavaScript to implement the functions of backstage, such as the connectors of different types of document.
- Finally, integrate the two tools as a comprehensive one.

5.2 Basic Code Structure

Matlab libraries (e.g. cvx) were used in my implementation.

One example of simulation:

```matlab
%% BS model
S0=T1(end); % stock price
T=preNumber; % maturity
if type==1 % call
    K=S0*(1+abs(mu))`preNumber; % rate of return
    [Call,Put]=mybsfun(S0,K,T,volatility); % European option
    BSprice=Call
    % simulation of neural network
    priceroute=pricerouteMat(indexmin,end);
    [Call01,Put01]=bsfun2(r,T,K,priceroute); %
    NPprice=Call01
    % the actual price
    priceroute=T2(end);
    [Call01,Put01]=bsfun2(r,T,K,priceroute); %
    Realprice=Call01
```
One example of prediction:

```matlab
filename=[stockID,’.csv’]:
data=cvread(filename,1,:);% load data
price=data(1:M,4); % the final price
price=flipud(price):
% rate of return
returnmat=(price(2:end)-price(1:end-1))./price(1:end-1):
mu=mean(returnmat):
% volatility
volatility=std(returnmat):

%%% prediction
[bppre,T2,ybptrain,I1]=bpfun(price,N,preNumber): % EP
rp2pre
[rbfpre,T2,yrbftrain,I1]=rbffun(price,N,preNumber): % RBF
rp2pre
[smpre,T2,ysmptrain,I1]=svmfun(price,N,preNumber): % SVM
```

Pic 11

The process of how to load the data by JavaScript in Pic 13 and how to connect to the results in format of .jpeg or .png by HTML in Pic 12.
The process of how to connect to the database in format of .csv or .xlsx by HTML in Pic 14.
6. Testing and Evaluation

6.1 Experimental Results and Evaluations

The data I used in this project are the stock index from Yahoo finance. The transaction data of stock prices are from 23 June 2006 to 23 June 2016 and the homologous option prices are from 1 January 2016 to 23 June 2016. In this project, I mainly concentrated on the strike prices but the bid and trading prices were not as my important considerations. I studied three types of options, they were in-the-money, at-the-money and out-the-money, to test the final predicted performance. The data with $K = 5800$, $K = 6000$ and $K = 6200$ respectively represented the in-the-money, at-the-money and out-the-money options in the sample period. The Fig 8 shows the call option of strike price $K$, parallelly, the put option of strike price $K$ is plotted in Fig 9.

![Call option of strike price K](image-url)
The previous 150 days’ data were used as the training process for SVM and the last 30 days’ data were used for prediction of already trained SVM. It was worth noting that the cumulative process was only used for training SVM, however, one-step-ahead (sliding window size $L$) forecasting was used for predicting process. Every step of training process would generate tiny errors and once all these tiny errors accumulating together which will result in a relatively large deviation. Therefore, the aim of employing one-step-ahead forecasting was to prevent the problems of accumulative error from previous period from leading to large error in forecasting.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Forecasting performance of SVM model on the three call options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K = 6200$</td>
</tr>
<tr>
<td>SVM</td>
<td>0.00280</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Forecasting performance of SVM model on the three put options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K = 6200$</td>
</tr>
<tr>
<td>SVM</td>
<td>0.00500</td>
</tr>
</tbody>
</table>

From the table 1 and table 2, we can compare the forecasting performance (error) of different types of call or put options.
The residuals between true values and SVM predictions when $K = 5800$ can be seen in Fig 10.

![Fig 10](image1)

The residuals of in-the-money

The residuals between true values and SVM predictions when $K = 6000$ can be seen in Fig 11.

![Fig 11](image2)

The residuals of at-the-money
The residuals between true value and SVM predictions when $K = 6200$ can be seen in Fig 12.

![Fig 12 the residuals of out-the-money](image)

In SVM algorithm, I chose three parameters to study in order to compare different results and obtain the best architecture. The three parameters are: the amount of training inputs, the capacity and the e-insensitivity $\varepsilon$. I tried 2800 points (10 years), 1400 points (5 years) and 200 points (1 year) of data to train the SVM. Apparently, the more data I used, the more time it took. However, compared with these different amount of data, the training results are quite similar actually and the precision of 10 years’ points was a little more accurate than others, however, it took longer time than others as well. If the time was chosen as priority, you can use the less data; if the accuracy was chosen as priority, the more points can be taken. What we should note here is that only when data is many enough, the training results of SVM are quite similar. If not, the training results will present a larger deviation.

The value for capacity and e-insensitivity $\varepsilon$ were needed to be chosen carefully, since they would bring about overfitting or underfitting upon SVM. Various results and comparisons were mentioned below.
<table>
<thead>
<tr>
<th>TP</th>
<th>Max Error (%)</th>
<th>Mean Error (%)</th>
<th>Error less than 10%</th>
<th>Error less than 5%</th>
<th>C</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>3353</td>
<td>61</td>
<td>135</td>
<td>80</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>300</td>
<td>1282</td>
<td>35.7</td>
<td>135</td>
<td>92</td>
<td>10</td>
<td>0.01</td>
</tr>
<tr>
<td>300</td>
<td>1152</td>
<td>34.4</td>
<td>138</td>
<td>94</td>
<td>10</td>
<td>0.005</td>
</tr>
<tr>
<td>300</td>
<td>1356</td>
<td>35</td>
<td>140</td>
<td>91</td>
<td>10</td>
<td>0.001</td>
</tr>
<tr>
<td>300</td>
<td>1339</td>
<td>34.9</td>
<td>141</td>
<td>93</td>
<td>10</td>
<td>0.0005</td>
</tr>
<tr>
<td>300</td>
<td>1540</td>
<td>38.16</td>
<td>137</td>
<td>92</td>
<td>15</td>
<td>0.01</td>
</tr>
<tr>
<td>300</td>
<td>1338</td>
<td>36.2</td>
<td>134</td>
<td>95</td>
<td>15</td>
<td>0.005</td>
</tr>
<tr>
<td>300</td>
<td>1518</td>
<td>36.4</td>
<td>133</td>
<td>93</td>
<td>15</td>
<td>0.001</td>
</tr>
<tr>
<td>300</td>
<td>1524</td>
<td>36.5</td>
<td>131</td>
<td>94</td>
<td>15</td>
<td>0.0005</td>
</tr>
<tr>
<td>300</td>
<td>1707</td>
<td>38.8</td>
<td>128</td>
<td>92</td>
<td>20</td>
<td>0.001</td>
</tr>
</tbody>
</table>

TP is the number of test points used, Max Error is the maximum between actual output and that predicted by SVM, Mean Error is the average error of the size of test points. From the comparisons and results, there were some inferences that could be made:

1. When the value of capacity was set at 1, the SVM framework was underfitting.
2. When the value of capacity was set at 20, the SVM framework was overfitting.
3. By contrast, the best performance was with the capacity at 10.
4. In order to reduce the mean square error (MSE), the e-insensitivity \( \varepsilon \) should be set below 0.005.

**BP:**

In back-propagation (BP) neural networks, I partitioned the data (10 years’ data) into three parts: the first part is used for training, the second part is used for validation and the last part is used for testing and forecasting. The data for validation is effective to avoid underfitting after training stage, in the meantime, the data for testing must not be put into the network in training process, otherwise, it will lead to overfitting. There is a necessity to remember that it is not a good idea to excessively diminish the training error. If you overfit the training process, once when certain data has been removed or added, the testing and forecasting process will generate larger deviations. It means the created networks lose the robustness of generalization. The volatility of BS was estimated by historical data or implied by functions based on certain period, while the volatility of BP networks captured it by itself, namely “black box”. On the other hand, the estimation of volatility \( \sigma \) was figured out under the encapsulated condition, which was different from conventional methods. In order to evaluate the volatility \( \sigma \), I employed two ways to build the BP network: the first one was that I partitioned the training, validation and testing data in random manner; the second one was that I sequentially partitioned the training, validation and testing data. The results of experiments indicated the ordered data outperformed the random data considerably. It turned out to be that the volatility depended on time series even though it existed in the neural networks. With random orders to create BP networks, it will essentially
destroy the trend of volatility and then lead to inaccurate predictions.

The data I used in this project are the stock index options from Yahoo finance. The transaction data points of call option are from 1 January 2016 to 23 June 2016. Likewise, in this project, I mainly concentrated on the strike prices but the bid and trading prices were not as my important considerations. In order to improve the performance of training, the data should be partitioned reasonably, in other word, the range of data should be narrowed. A call option is both in-the-money when $S > K$ and out-the-money when $S < K$ is easy to find, however, at-the-money option when $S = K$ is hard to find. So the at-the-money option we define when $S \approx K$ and $\alpha$ is usually to define the extent of at-the-money when $S \approx K$.

### Table 4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>$\mathcal{S}/\mathcal{T}$</td>
<td>$\mathcal{S}/\mathcal{T}$, $\varrho$</td>
<td>13 variables</td>
<td>$\mathcal{S}_t$, $\mathcal{T}_x$, $\mathcal{V}$</td>
<td>$\mathcal{S}$, $\mathcal{T}$</td>
</tr>
<tr>
<td>Hidden</td>
<td>4</td>
<td>4,1,0,20</td>
<td>5</td>
<td>5</td>
<td>2,3</td>
</tr>
<tr>
<td>Output</td>
<td>$\mathcal{C}/\mathcal{X}$</td>
<td>$\mathcal{C}/\mathcal{X}$</td>
<td>$\mathcal{C}$</td>
<td>$\mathcal{C}$</td>
<td>$\mathcal{C}$</td>
</tr>
<tr>
<td>Activation</td>
<td>sigmoid</td>
<td>logistic</td>
<td>sigmoid</td>
<td>hyperbolic tanget</td>
<td></td>
</tr>
<tr>
<td>Measure</td>
<td>$R^2$, NRMSE, MAPE</td>
<td>$MAD$, $MBE$</td>
<td>$ASE$, $MAE$, $R^2$</td>
<td>$NMSE$</td>
<td></td>
</tr>
<tr>
<td>Market</td>
<td>SP500</td>
<td>SP100</td>
<td>SP500</td>
<td>Nikkei 225</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>87-01</td>
<td>January-02-December-04</td>
<td>January-02-December-02</td>
<td>December-04-January-05</td>
<td>January-05-December-05</td>
</tr>
<tr>
<td>No. of Data</td>
<td>6,000+</td>
<td>3,303</td>
<td>1,107</td>
<td>17,790</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5

Data partition according to moneyness (stock price/strike price). $\alpha=6\%$ is adopted

<table>
<thead>
<tr>
<th>$\alpha%$</th>
<th>Out $(S/X &lt; 1-\alpha)$</th>
<th>At $(1-\alpha &lt; S/X &lt; 1+\alpha)$</th>
<th>In $(S/X &gt; 1+\alpha)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11,036</td>
<td>4,762</td>
<td>1,992</td>
</tr>
<tr>
<td>4</td>
<td>9,899</td>
<td>4,016</td>
<td>3,875</td>
</tr>
<tr>
<td>5</td>
<td>9,372</td>
<td>3,607</td>
<td>4,751</td>
</tr>
<tr>
<td>6</td>
<td>8,772</td>
<td>3,324</td>
<td>3,875</td>
</tr>
<tr>
<td>10</td>
<td>6,424</td>
<td>2,288</td>
<td>9,078</td>
</tr>
</tbody>
</table>

Table 5 showed the number of moneyness in each dataset on the basis of the choices of $\alpha$. I employed $\alpha = 6\%$ in the following experiment because the range of data is most narrowed.
Table 6 showed the comparisons between BP neural networks and Black-Scholes model. The above results indicated that BP outperformed the BS for in-the-money and out-the-money option. In contrast, BS performed better than BP for at-the-money. In my research, I mainly study three standard types of options which are referred to as in-the-money, at-the-money and out-the-money. A call option is in-the-money when \( S > K \), at-the-money when \( S = K \) and out-the-money when \( S < K \). What is more, I achieved the similar results and same conclusions as the previous research. If the underlying asset price is below the strike price, the option holder will not exercise the contract because exercising option would be unprofitable. If the underlying asset price is above the strike price, exercising option would be profitable only. (Jingtao Yao etc., 2000).

**RBF:**

Perform k-means clustering to find centers for basis function:

By comparing the differences between training and test errors at different values of the number of basis function, \( K \) from figure 13, it can be observed that the training and test errors will go down with the values of \( K \) going up. Furthermore, when the values of \( K \) approach to infinity, the training errors of RBF is approaching to 0 and it similar to the ones of Linear. In contrast, the test errors will not always reduce with the values of \( K \) increasing, moreover, it will approach to a stable level when \( K \) reach a
certain number (may be 100). It means that the prediction on test errors is always higher than it on training errors.

<table>
<thead>
<tr>
<th></th>
<th>RBF library</th>
<th>RBF K-mean clustering</th>
<th>RBF gradient descent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSE</strong></td>
<td>0.2935</td>
<td>0.2387</td>
<td>0.1806</td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>0.2902</td>
<td>0.2359</td>
<td>0.1789</td>
</tr>
</tbody>
</table>

![RBF errors based on historical volatility](image1)

![RBF errors based on historical volatility](image2)

**Fig 15**

**Fig 16**

<table>
<thead>
<tr>
<th></th>
<th>RBF library</th>
<th>RBF K-mean clustering</th>
<th>RBF gradient descent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSE</strong></td>
<td>0.2875</td>
<td>0.2289</td>
<td>0.1803</td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>0.2903</td>
<td>0.2431</td>
<td>0.1778</td>
</tr>
</tbody>
</table>
The results above showed that the RBF toolbox had the worst ability of generalization.

After many trials by using matlab library, I found that the preferable number of hidden neurons were 25 when historical volatility was made use of to train the networks, while the preferable number of hidden neurons were 34 when implied volatility was utilized.

When use k-mean clustering to set up central vector of hidden layer in RBF networks, $k$ can be regarded as the number of nodes of hidden layer. The expansion constant is the shortest distance between the nodes of hidden layer.

After many trials by using k-mean clustering, I found that the preferable number of hidden neurons were 17 when historical volatility was made use of to train the networks, while the preferable number of hidden neurons were 18 when implied volatility was utilized.

After numerous tests we can find aiming at historical volatility: Under historical volatility circumstance, radial basis function networks slightly outperformed back-propagation networks in the majority, because when sample data was adequate enough, RBF had a better stability.

After numerous tests we can find aiming at implied volatility: Under implied volatility circumstance, back-propagation networks were superior to radial basis function networks in the majority, because the hidden units of BP generated a $\sigma$ which reflected the market discipline even better.

6.2 Comparative Results

Black-Scholes model evaluation:
Figure 19 and Figure 20 show the estimated call and put price by using Black-Scholes model compared to the real option prices respectively. For the options with sensible strike prices, the Black-Scholes model is able to accurately predict the actual option prices. Figure 19 and Figure 20 are one of the estimated price pairs for different strike price. All of the call/put pairs appear a similar trend. The estimated call option price is higher than real call option price, but the estimated put option price is lower than real put option price. If the maturity is longer, the difference will be greater. While approaching to the maturity date, the difference of the estimated price and real price become smaller and smaller.

For the Black-Scholes model, the volatilities used in the same day for different strike price are the same. The Fig 22 shows the volatilities from historical data and the implied volatilities computed from Black-Scholes model. The implied volatility is less than the historical volatility. The reason for this phenomenon is that some implied volatility cannot be computed. In Fig 22, the historical volatility is more stable than the
implied volatilities. This is because the historical volatilities come from a stable historical data but the implied volatilities are the result computed from Black-Scholes model which use the true value. When the strike price increases, the volatilities become larger, and the volatility smile can be seen in Fig 23. Fig 24 shows that the implied volatilities on a particular day for different strike prices are not the same.
Support vector machine results:

Fig 25 and 26 show the simulation result and prediction result respectively.

Back-propagation neural network results:

Figure 27 and 28 show the simulation result and prediction result respectively.
Radial basis function neural network results:

Figure 29 and 30 show the simulation result and prediction result respectively.

6.4 Critical Evaluation

Option is a nonlinear financial instrument and it has a high transaction risk. Owing to its characteristics of nonlinear prices and high risk, it is significant for investors to have an accurate forecasting on stock prices as much as possible to hedge their risks.

After some experiments, there are some general conclusions about can be drawn:
1. The Black-Scholes model is an extremely effective method for option pricing of at-the-money and it is especially suitable for the maturity exceeding two months, in other words, the customer held it when the time to expiration had better not less than two months.
2. It occurred some significant deviation for option pricing of in-the-money and out-
the money, and the errors between actual prices and estimated prices presented a larger tendency.

3. The results of too short period always had a relatively large deviation, so that there was a suggestion for users who held the options no less than a month when using BS formula to calculate it.

4. When you used one or both of the historical and implied volatility approaches to estimate the \( \sigma \) of which if the value was too high or too low, Black-Scholes model would not be applicable.

5. The experimental results manifested that neural networks outperformed the Black-Scholes formula on the options of in-the-money and out-the-money.

6. On the contrary, the experimental results showed that neural networks were worse than Black-Scholes model on the options of at-the-money. It implied that the conventional methods such as Black-Scholes model cannot be completely substituted by artificial intelligent algorithms. There were the complementary relations between each other and one’s weakness might be made up by the strength of the others.

7. Time sequence was a vital factor for both parametrically conventional methods and nonparametrically improved algorithms. The performance of sequential models was superior to random models basically.

8. Compared with neural networks (e.g. back-propagation and radial basis function), support vector machine had a prominent advantage that it took less time than the former, when the accuracy was similar. This was because it was feasible to use less training data to find the best values for capacity \( C \) and \( \varepsilon \)-insensitivity \( \varepsilon \) so as to establish SVM.

9. Compared with back-propagation network and after 20-40 times’ test, radial basis function network ran approximately 5-10 times faster than the former while using the same space dimensions and memory requirements. With the growing demand for storage space and computational speed, it is becoming popular and necessary to use fast and efficient algorithm.

10. One of the conspicuous advantages of neural network does well in nonlinear inputs and outputs relationship modelling when it explores the dynamic financial applications. On the contrary, neural network involves the following several disadvantages:

    1). It is dependent on the number of parameters (e.g. network size, number of inputs, numbers of hidden neurons and initial weights being chosen).
    2). To a great extent of being probably trapped in local minima leading to an extremely slow convergence.
    3). Overfit the training data will result in a poor generalization capability.
7. Conclusion

I am convinced that the nonparametric approach of option pricing with support vector machine and neural networks is much more effective and simple to handle. In the first place, I employed a conventional option pricing method -- Black-Scholes model to forecast the prices. In the second place, I allowed SVM, BP networks and RBF networks to lessen the forecasting errors of conventional method, since the conventional method is a kind of parametric approach to imitate the trends of real option prices. The nonlinear approximation can decrease the forecasting errors, that is the reason why SVM, BP networks and RBF networks can be used. In the end, massive experimental studies and results have showed the abilities of improved forecast accuracy of the three approaches and the data come from the London Stock Exchange Group (LSEG).

The experimental results suggest that the three refined algorithms referred to in this project can beat the Black-Scholes model in volatility markets, whereas, BS model works well when all the assumptions met. I maintain that the relationship between BS and other optimized approaches should be complementary, because in the stable and ideal market, the conventional methods such as BS model should be always as the first choice, and other optimized approaches perform better in volatility conditions and are good at forecasting in the future. Moreover, the decreasing of parameters in modified approaches substantially reduces the burdens of users for determining parameters and establishing models.

In the securities market, Support Vector Machine (SVM), Back-propagation neural networks (BP) and Radial Basis Function neural networks (RBF) approaches have been authenticated that they were able to noticeably refine the Black-Scholes model (BS) and the forecasting results and accordingly improve the forecasting accuracy by shrinking the nonlinear data error.

7.1 Future Work

Firstly, in this project, the study mostly concentrates on one output of different algorithms. To make the approaches more efficient and comply with the actual rules of reality, the modified models should not be only based on one output, nevertheless need to satisfy a majority of outputs or weighted outputs. Secondly, in order to make the operation much easier and convenient for users to use the integrated tool, the matlab server should be connected to the web sites directly, in other words, when users want to see the optimized analysis outcomes or implement
their transactions, they can only operate the web tools other than both of them. The web tool can invoke the matlab server and obtain the results it produced.

Thirdly, although I have designed the architectures of three different algorithms and they turn out to be much more effective and efficient than pure Black-Scholes, at the same time users do not pay attention to the processes of different approaches, user still need to change the stocks which they focus on and update them in matlab programming by themselves. In practice, it still burdens the users who are not skilled in computer and programming a lot, even though it is no need for them to take too much energy on the process of implementation. Therefore, the future work that I should do is to explore a new way to solve these problems and let the people out of the codes thoroughly.

7.2 Critical Reflection

Although the volatility $\sigma$ is actually not reflected in the neural network straightforwardly, it can be computed by neural network finally. I still cannot explain how and why it occurred distinctly, since neural network is usually regarded as a “black box”. Sometimes this can be considered as a big drawback due to lacking of insight. Additionally, because of the nonlinear property of neural network, it is often not possible to extract simple rules to describe how it worked and why the predictions it made is accurate.

At the same time, numerous empirical studies have manifested that a combination of different forecasting approaches exceeded single forecasting model, on the grounds that separate forecasting models can supplement each other in obtaining diverse data. A typical example was presented that the linear and nonlinear time series models for forecasting the monthly return rate by Terui and Dijk (2002) and Zhang (2003). Terui and Dijk (2002), which proved the above conclusion: the combined forecasting models have stronger superiority than individual model. As a consequence, in order to observably improve the algorithmic performance further, different algorithms can be combined with each other (e.g. Gaussian distribution or Kalman filter with SVM or gradient descend or genetic algorithm with neural network) in the future.
8. References

