Prediction and Portfolio Optimization in Quantitative Trading Using Machine Learning Techniques

Van-Dai Ta†
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, TAIWAN
daitv88@gmail.com

Chuan-Ming Liu
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, TAIWAN
cmliu@ntut.edu.tw

Direselign Addis
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, TAIWAN
direselign@gmail.com

ABSTRACT

Quantitative trading is an automated trading system in which the trading strategies and decisions are conducted by a set of mathematical models. Quantitative trading applies a wide range of computational approaches such as statistics, physics, or machine learning to analyze, predict, and take advantage of big data in finance for investment. This work studies core components of a quantitative trading system. Machine learning offers a number of important advantages over traditional algorithmic trading. With machine learning, multiple trading strategies are implemented consistently and able to adapt to real-time market. To demonstrate how machine learning techniques can meet quantitative trading, linear regression and support vector regression models are used to predict stock movement. In addition, multiple optimization techniques are used to optimize the return and control risk in trading. One common characteristic for both prediction models is they effectively performed in short-term prediction with high accuracy and return. However, in short-term prediction, the linear regression model is outperform compared to the support vector regression model. The prediction accuracy is considerably improved by adding technical indicators to dataset rather than adjusted price and volume. Despite the gap between prediction modeling and actual trading, the proposed trading strategy achieved a higher return than the S&P 500 ETF-SPY.

CCS CONCEPTS

• Computing methodologies • Machine learning • Learning paradigms

KEYWORDS

Machine learning, Quantitative trading, Stock prediction, Portfolio optimization, Trading strategy.

SoICT '18, December 6–7, 2018, Da Nang City, Viet Nam

ACM ISBN 978-1-4503-6539-0/18/12...$15.00
https://doi.org/10.1145/3287921.3287963

1 Introduction

Quantitative trading as known as algorithmic trading is the process of buying and selling stocks/ assets based on the implementation of trading strategies in a disciplined and systematic manner. These trading strategies are developed through rigorous research and mathematical computations. When quantitative trading strategies were first introduced, they were wildly profitable and swiftly gained market share. High frequent trading (HFT) accounted for high percentage of daily trading volume with millions of shares. But as competition has increased, profits have declined gradually. As the difficulty of seeking profit in trading increases, machine learning is considered as the most powerful tool to gain a competitive advantage and boost investment return.

Machine learning offers the number of important advantages over quantitative trading by providing variety of effective computer algorithms that allow to move from finding associations/patterns based on historical data to identifying and adapting to market trends in a systematic manner. The algorithms learn to use predictor variables to predict the target variable.

There are variety of trading strategies which used machine learning for trading analysis, including linear regression, neural networks, support vector machine, deep learning [1], [2], etc. Among these strategies, regression is a typical and most common used machine learning technique in finance such as earning prediction, credit loss forecast, and stock market prediction. In this paper, we use linear regression, and support vector regression to conduct stock movement prediction. Based on the predicted results, multiple intelligent portfolio allocations and optimization techniques are used in order to optimize the portfolio return.

The rest of the paper is structured as follow: Section 2 presents basic concepts, and typical architecture of a quantitative trading system as well as quantitative trading strategy workflows. Section 3 gives the introduction to machine learning techniques, taxonomies, and how to select a suitable machine learning model. Section 4 describes the roles of machine learning in quantitative trading, and our proposed methodology. Experimental results are presented in Section 5. Both regression models are effectively

performed on the dataset by obtained high accuracy and attractive return compared to S&P500 ETF – SPY. Finally, conclusions and discussions are presented in Section 6.

2 Quantitative Trading System

Quantitative trading can be defined as the systematic implementation of trading strategies that human beings create through exhaustive research. In this context, systematic is defined as a disciplined, methodological, and automated approach. People conduct the research and decide which strategies will be used to perform on universe of stocks for trading system. Those people, the ones behind quantitative trading strategies, are commonly referred to a quants or quant traders [3]. The idea of quantitative trading is designed to leverage statistical mathematics, computer algorithms, and computational resources for high frequent trading systems which aims to minimize risk and maximize return based on the historical performance of the encode strategies tested against historical finance data.

Most investment strategies are created and implemented by human beings in which decision driven by psychology and emotion. In contrast, quantitative trading is designed to eliminate arbitrariness by creating a disciplined trading strategies that was tested by a computational model. In essence, a decision driven by emotion, indiscipline, passion, greed, and fear will be taken out from quantitative investment process. Moreover, investment banks use quantitative trading which contains a complex mechanism to derive business investment decisions from insightful data such as Goldman Sachs, Morgan Stanley, etc. Quantitative trading involves in using complex mathematics to derive buy and sell orders for derivatives, equities, foreign exchange rates and commodities at a very high speed.

Quantitative trading strategies are investment strategies based on quantitative analysis of financial markets and prediction of future performance. The strategy and associated predictions depend on the time-scale of the investment, as exemplified by the following classes of quantitative strategies: fundamental analysis when a stock is trading under intrinsic value. The long-term strategies motivated quant has a quarterly time-scale. Systematic macro based on macroeconomics analysis or market events and trends to identify opportunities for investment. Systematic macro strategies are model-based and executed by software with limited human involvement. The above systematic macro has a monthly time-scale. Convergence or relative value trades and other statistical arbitrage (StarArb) strategies refer to trading in similar assets that are expected to converge in value. These strategies are examples of StatArb strategies which have time scales ranging from minutes to months. High frequency trading (HFT) as known as buying/selling a large amount of shares in a short time. The time scale of HFT in milliseconds and holding period of the traded shares is usually less than one second.

A typical quantitative trading system has three modules: a alpha model, a risk model, and a transaction cost model which feed into a portfolio construction model, which in turn interacts with the execution model as shown in **Figure 1**. The alpha model is designed to predict the future of the instruments the quant wants to consider trading for the purpose of generating returns. For example, in a trend-following strategy in the futures markets. On the other hand, the alpha model is designed to forecast the direction of whatever futures markets the quant has decided to include in a strategy. Risk models, by contrast, are designed to project limit the amount of exposure the quant has to those factors that are unlikely to generate returns but could drive losses. For example, the trend follower could choose to limit his directional exposure to a given asset class, such as commodities, because of concerns that too many forecasts he follows could line up in the same direction, leading to excess risk; the risk model would contain the levels for these commodity exposure limits. The transaction cost model is used to help determine the cost of whatever trades are needed to migrate from the current portfolio to new portfolio is desirable to the portfolio construction model. Almost any trading transaction costs money, whether the trader expects to profit greatly or a little from the trade. The alpha, risk, and transaction cost models then feed into a portfolio construction model, which balances the tradeoffs presented by the pursuit of profits, the limiting of risk, and the costs associated with both, thereby determining the best portfolio to hold.

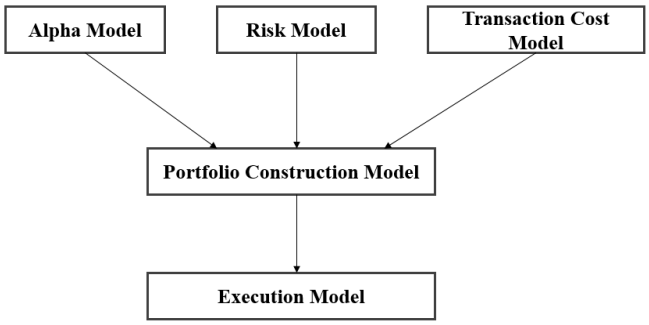

Figure 1: Quantitative trading modules.

Quantitative trading strategy workflow consists of 6 stages: data collection, data preprocessing, trade analysis, portfolio construction, back-testing, and execution as shown in **Figure 2**

Stage 1: Data collection

Multiple data sources can be collected through finance data API such as MorningStar, Quandl, Google, Yahoo Finance API or provided by security companies. Stock data are collected under various types of formats. It could be under fundamental, technical, macroeconomics, or even sentiment data in text form with variety of time-scales.

Stage 2: Data preprocessing

Collected data could be time series data, non-stationary data, unstructured data under the text or missing data. Therefore, it requires a heavy task for data normalization, scaling, and transformation in order to aggregate all the data sources.

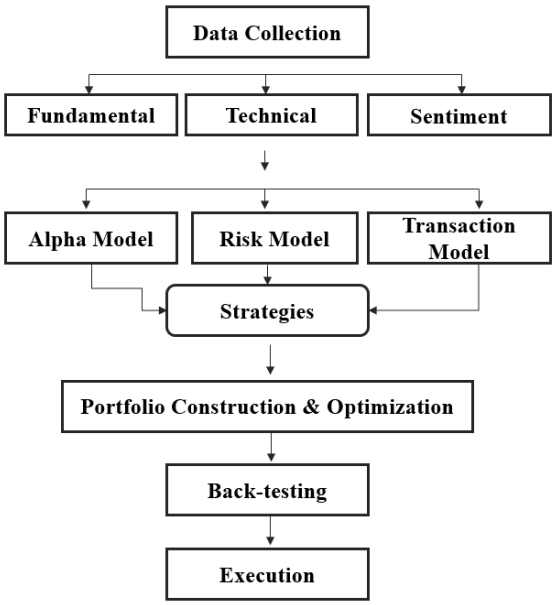

Figure 2: Typical quantitative trading workflow.

Stage 3: Trade analysis

Prediction model is the core of trade analysis stage, in which alpha, risk, and transaction models are implemented to develop effective trading strategies. Alpha model can be conducted by theory-driven or data-driven approach. In the theory-driven approach accesses the cleaned data from fundamental, technical, macroeconomics, sentiment data as the input for modeling. Based on the collected, alpha model aims to predict or forecast the price movement of stock then generate some strategies called passive trading strategies which can be used for back-testing before execution. In data-driven approach ingests the data sources in real-time and interactive with historical data. Data-driven approach is mostly focus on trading patterns as indicators and signal generation. By scanning the signals/ indicators in real-time that can generate real-time strategies for a trading system with high frequent trading. All quantitative trading systems seek to archive high alpha as profit. However, from time to time, risk exposures will not generate profits as expected, but risk can impact the return of trading strategy day to day. Risk model enables quants to define, measure and control risks. There are two generally accepted ways of measuring the amount of risk in the marketplace. The first is measures risk by computing the standard deviation of the returns of various instruments over time as known as volatility. The more volatility, the more risk is said to be present in the market. The second way to measure risk is to measure the level of similarity in the behavior of the various instruments within a given investment portfolio. If all the instruments in a portfolio are perfectly correlated, then as one bet goes, so go all the other bets.

The transaction cost model is designed to record all of trading transaction costs. There are three major components in the transaction cost model: commissions and fees, slippage, and market impact. Transaction costs are important to investors because they are one of the key determinants of portfolio returns.

Stage 4: Portfolio construction and optimization

Portfolio construction and optimization: Stock investors are interested in combining multiple stocks into a single investment portfolio instead of investing only a specific stock in the purpose of neglecting the volatility of investment portfolio. This can be done naively by having equal weights for all the stocks, or automatically adjusted weights in order to maximize the portfolio return, which is called portfolio optimization. Portfolio construction starts with three basic questions: reduce the risk, maximize the return, capital allocation.

Stage 5: Back-testing

Back-testing: A key difference between a traditional investment management process and a quantitative investment process is the possibility of back-testing a quantitative investment strategy to see how it would have performed in the past [4]. Therefore, before implementation, all the quantitative trading strategies are thoroughly back-tested. Back-testing is a simulation of a trading strategy used to evaluate the performance of the proposed strategy. While back-testing does not allow one to predict how a strategy will perform in the future conditions, but its primary benefit lies in understanding the vulnerabilities of strategy through a simulated encounter with real-world conditions of the past [5].

Stage 6: Execution

Quants build alpha models, risk models, and transaction cost models. These modules are fed into a portfolio construction model, which determines a target portfolio. But having a target portfolio on a piece of paper or computer screen is considerably different than actually owning that portfolio. The final stage of the quantitative trading system is to implement the portfolio decisions made by the portfolio construction model. Despite the fact that the trade generation can be semi-automated or even fully-automated, the execution mechanism can be manual or fully automated through a high performance application programming interfaces (APIs). For low frequent trading strategies, manual and semi-manual techniques are common. For high frequent trading strategies, it is necessary to create a fully automated execution mechanism, which will often be tightly coupled with the trade generator. The key considerations when creating an execution system are the interface to the brokerage, minimization of transaction costs and divergence of the performance of the live system from back-tested performance.

3 Machine Learning Techniques

Artificial intelligence (AI) studies intelligent agents that perceive their environment and perform different actions to solve tasks that involve mimicking the cognitive functions of humans [6]. AI systems need the ability to acquire their own knowledge by learning from raw data. A rational agent should not only perceive, but also

learns as much as possible from what it perceives. Learning is the ability to generalize using data in order to optimally act when presented with new data. Machine learning (ML) is the subfield of AI that studies perception, learning and action tasks as algorithms that learn from data. ML has been emerged from multiples fields as the combination of statistical modeling with dynamic programming as a response to real world industrial demands that called for methods to effectively handle large and high dimensional data. ML has been using in various application different industries such as natural language processing, computer vision, smart manufacturing, supply chain, and finance [7] [8] [9]. A computer algorithm is the core part of ML, as long as the development/improvement of ML, multiple advanced computational algorithms have been designed to statistically estimate complex functions that typically cannot be expressed in closed form ML computational models not only handle large-scale complex data as big data based on distributed environment, but also reduce in computation time as real-time or stream processing.

A concise and operational definition of ML as a process of learning is given in [10]. In the definition, a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

There are two general types of ML tasks T. The first type is perception tasks, in which the ML algorithm enables to learn from the dataset to perform one specific predefined action. For example, a classification algorithm that responsibility to classify all images into people and animal in the mixed image dataset. The second type of ML tasks is called action tasks that the final goal is to find a decisive action based on what learned from the perception tasks.

The performance measure P is specific to the task T. For example, in binary classification task to classify people or animal, one measure of the model accuracy can be the error rate which defined as the ratio of incorrectly classified examples to the total number of examples. The error rate also can be viewed as an estimation of the rate of expected $0-1$ loss. That gives the error 1 for each misclassified example and 0 for correct classified example.

$$Error\ rate = \frac{N_{incorrectclassified}}{N_{total}} \quad (1)$$

$$Accuracy = \frac{N_{correctclassified}}{N_{total}} = 1 - Error\ rate \quad (2)$$

However, such performance measure is inconvenient in practical because it might change this continuously when parameters of the models are changed continuously. In other words, such a performance measure metric would be a non-differentiable function of its parameters, so that no gradient-based optimization could be applied in this setting. Therefore, a smooth and differentiable alternative to the error rate as a performance measure is considered instead of the probability or low probability of the observed data under the assumptions of the model. This leads to a differentiable objective function for the process of tuning all the parameters to the data, which can be efficiently done by using gradient based optimization techniques. In regression, one possible choices for performance measure is a mean square error (MSE) as represented in (3). Y_i are the true observed outputs and $\hat{Y}_i$ are model estimates for these outputs. The sum runs over all observations, so that the MSE is 0 only when all data points are matched exactly, without any errors.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i - \hat{Y}_i}{Y_i}\right)^2 \quad (3)$$

The performance measure P improves with experience E as a result of learning. Here the learning from the experience is the ability to generalize. In general, learning from experience can roughly fall into three major categories: supervised, unsupervised, and reinforcement learning.

In supervised learning, each training data consists of a pair of input objects and the desired output value or target. The main objective is to produce a function that will map input values to the output value in a way such that when unknown new data is fed the function enables to make a reasonable prediction about the target value. Such type of learning is called supervised learning because training data sets are predefined and can be thought of as an advisor supervising the learning process. The second one where the expected outcome is not defined is called unsupervised learning. Unsupervised learning refers to a broad array of machine learning algorithms which are used to draw inferences from data sets consisting of input data without labeled responses. This implies that the algorithm is not presented with the right output for a sample input, but instead is forced to learn the correct way to produce an output in an unsupervised manner. Unsupervised learning mostly forms the significant part of learning for the human brain and hence is an important segment of machine learning. Reinforcement learning involves techniques that try to retro-feed the model to improve performance. In order to accomplish this, the model needs to be able to interpret signals, decide on an action and then compare the outcome against a predefined reward system. Reinforcement learning tries to understand what needs to be done in order to maximize the rewards.

3.1 Machine Learning Taxonomy

Both supervised and unsupervised learning are about perception tasks because in both of them an algorithm should perform one particular action. For example, classify an image, spam email, and so on. On the other hand, reinforcement learning is all about action tasks. The first task, and one the most commonly encountered in machine learning is regression with the objective to learn a real valued function that maps an N dimensional space $f: R^n \rightarrow R$ by given the training set of input output pairs (X_i, Y_i), where Y is a real number, and X is a vector in R^n. Regression model can be used for sale demand forecast and similar tasks [11]. Another very common supervised learning task is classification with the objective is nearly the same as regression except that in classification Y_i are discrete numbers rather than continuous various as in regression. For

classification, typical industrial applications are spam detection, image recognition or document classification [12].

In unsupervised learning, clustering is one of the most popular task as known as segmentation in business. Dataset divides into clusters, or partitions, in which a cluster is a set of a homogenous group of data points. In terms of functions it looks exactly the same as the classification does, except that not class labels are given to the algorithm. Clustering methods are often used for customer segmentation, or anomaly detection [13]. Another class of unsupervised learning algorithms is called representation learning. This term includes a large number of different approaches whose main idea is nevertheless always the same. Specially this task is to map an initial and dimensional data where N can be a real large number and to a lower dimensional space of dimension k where $k \leq n$. These tasks are also referred to as dimensional reduction or future instruction tasks. Representation learning methods are used among other things for text recognition and machine translation.

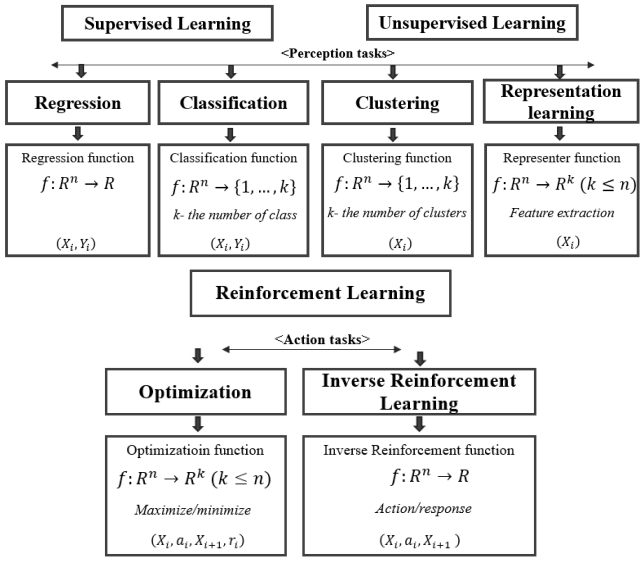

Figure 3: Landscape of machine learning.

The last category of machine learning algorithm is reinforcement learning. A function that is optimized in the enforcement lowering is called policy function. This function describes what an algorithm should do by given the current state of the dataset in order to maximize its total reward over a period of time. In this case, the training data should be in the form of tuples that contain the current state X_i action taken next state X_{i+1} in reward from taking this section. In addition to such direct reinforcement learning, there is also a varying interest in alternative formulation known as inverse reinforcement learning (IRL). In this method, everything is the same as for direct reinforcement learning, but there is no information on rewards received by the agent upon taking actions. Reinforcement learning methods are widely used in robotics and computational advertising [14]. The landscape of ML is shown in **Figure 3**.

3.2 Model Selection

Model selection is an important part of machine learning. It does not mean different choosing from the type of model like selecting *K*-nearest neighbor, Naïve Bayes, or Trees, SVM, etc. Model selection is choosing different complexity levels so it can be called complexity selection. Here, the complexity can be expressed by the flexibility to fit or explain data. A complex model with more features tends to have low bias because these models are more flexibility and more capacity to adjust data. However, the flip side of this is that adding more features would generally increase variance. In contrast, a simple model with fewer features tends to have high bias and low variance. Therefore, building a right model requires the right level of model complexity that matches the data complexity to optimal the tradeoff. In some cases, the right level of model complexity and model architecture could be established beforehand guided by some characteristics of the data. For example, classification is normally used for lower dimensional data and neural network often used for high dimensional data. However, no single machine learning classification algorithm can be universally better than the other one in all domains [15]. The advance of computational resource allows to build a complex model with low bias and high variance. One possible way to reduce the variance is normalized or bound somehow the value of model parameters, so the model output would vary less with a variance of input data.

4 Machine Learning for Stock Prediction and Portfolio Optimization

4.1 Machine Learning for Stock Prediction

A machine learning algorithm allows system to search for trading patterns within a complex data sets such as fundamental, technical and even sentiment data. Therefore, machine learning enables quants to build up multiple trading strategies. On the other hand, event-driven modeling improve the prediction confidence in order to maximize the profit and minimize risk [16], [17].

In quantitative trading system, the huge universe of stock data puts a heavy task for feature selection or indicator selection whether it is based on fundamental analysis, technical analysis, or sentiment analysis. Some popular types of machine learning algorithms for training such as decision trees, random forest/ gradient boost, neural network (LSTM), evolution algorithms have been used to solve improve this task [18], [19].

Wrapper is the way of leveraging a machine learning algorithm to select and evaluate indicator subsets. Wrapper techniques investigate the relationships to fine the best collection of indicators instead of looking at indicators individually [20], [21].

Ensemble learning is the way to combine multiple uncorrelated classifiers to generate a single or more robust signal. The advantages of using ensemble learning over a wrapper of random

forest is that we can use a lot of different classifiers to find different patterns and information from data [22], [23].

In addition, combining pattern recognition and association rule learning is a very powerful way to leverage machine learning algorithms while still being able to interpret the output and trading strategies [24].

Since the stock prediction model is considered as the core component of alpha model in quantitative trading. The prediction model investigates the input data such as technical, fundamental, macroeconomics, or sentiment data. The outcome predicted values could be input of the portfolio construction. To demonstrate the simple quantitative trading system, linear regression and support vector regression in Scikit-learn [25] are used to simulate how the quantitative trading perform. In order to optimal the system performance, three optimization models are used with the target of maximize return and minimize risk.

In this work, X is the set of N trading stocks in the large dataset, denoted as $X = \{X_1^T, X_2^T, \dots, X_N^T\}$, where T is the set of stock indicators or features which can be fundamental, technical, or sentiment indicators, the higher value of T will increase the dataset dimensions. Sample data can be collected by different k-factor (hourly, daily, weekly monthly) rate.

In this work, regression models are used to prediction the movement of stocks for multiple period of time $[t_0: t_k]$. Dataset is split into training and test set then fit the training set to the regression models. As the result, the set prediction accuracy P for X is calculated by (3), denoted as $P = \{p_1, p_2, \dots, p_N\}$. $\bar{P}$ is the prediction accuracy on average for each prediction model which will be the key to evaluate both prediction models. Then cumulative return for each stock is calculated for the predicted period $[t_0: t_k]$, denoted as $R = \{r_1, r_2, \dots, r_N\}$. To construct the portfolio, M ($M < N$) stocks with the accuracy rate greater than average predicted accuracy $\bar{P}$ and highest cumulative return in forecast set R are selected to construct a portfolio.

4.2 Portfolio Construction and Optimization

Portfolio Allocation

An investment portfolio is just a set of allocation in variety of securities or stocks. Key statistics of a portfolio: daily return, cumulative return, average daily return, standard daily return. Daily or cumulative return may indicate as one part of investment reward and evaluate the performance of a portfolio. However, high return may come with high volatility or risk in investment. Sharpe ratio is a measure for calculating risk-adjusted return, and this ratio has become the industry standard for such calculations. The Sharpe ratio is defined by the following relation:

$$SR = \frac{R_p - R_f}{\sigma_p} \quad (4)$$

Where $R_p$is the expected portfolio return, R_f is risk free return. σ_p is portfolio standard deviation. Risk free return as the return received by putting money in the investment such as bank savings account, LIBOR, treasury bonds that are essentially risk-free. Because the risk-free is really low value, we can assume σ_p is 0. The Annualized Sharpe Ratio (ASR) can be obtained by multiplying against a k-factor based off the sampling rate. The ASR is calculated as follows:

$$ASR = \sqrt{k}\,\frac{R_p - R_f}{\sigma_p} \quad (5)$$

Portfolio optimization

Instead of allocating equally weights for all stocks in the portfolio, we can use optimization techniques to maximize some performance measure which is called portfolio optimization. Monte Carlo Simulation (MCS) works as randomly assign a weight to each security in the portfolio, then calculates its mean daily return and standard deviation of daily return. Sharpe ratio is calculated and selected through thousands of randomly selected allocations. Another approach for portfolio optimization using mathematical calculation called Efficient Frontier as known as Mean-Variance Optimization [26].

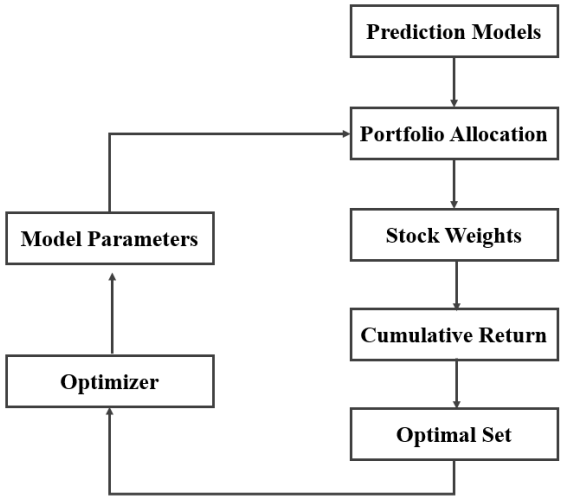

Figure 4: Optimization method.

Assume S is the set of stocks in the constructed portfolio, denoted as $S = \{X_1, X_2, \dots, X_M\}$. The return of the portfolio is calculated:

$$R = \sum_{i=1}^{M} w_i X_i \quad (6)$$

Where $W = \{w_1, w_2, \dots, w_M\}$ is the set of weights as the allocation ratio for the portfolio. The objective of optimization portfolio is try to find the optimal set Win order to maximize portfolio return in (6).

A general portfolio optimization method is shown in **Figure 4**. Based on the stock prediction models, the prediction results are input for portfolio allocation model to generate initial stock weights then the cumulative return of portfolio are calculated to obtain an optimal set as a fitness measure of a portfolio. In order to archive an optimal results as long as finding the best model parameters of allocation through the optimizer.

5 Experiment and Results

5.1 Data Preparation

In this work, we take 500 stocks data from S&P500 as our predicting data, which are represented by 500 identical stock tickers. The 10-year daily historical data are collected by Quandl API from Jan 1st, 2008 till Mach 27th, 2018 with 2576 total trading days.

5.2 Experimental Design

In this simulation, two regression models linear regression (LR) and support vector regression (SVR) are used to predict stock movement for different time periods as shown in **Table 1**.

Table 1: Prediction time periods

Period	Start	End	Days
1	2018-02-20	2018-03-27	26
2	2018-01-11	2018-03-27	52
3	2017-04-12	2018-03-27	78
4	2017-10-25	2018-03-27	104
5	2017-09-20	2018-03-27	129

Based on the predicted accuracy for each trading period, 4 tickers with high predicted accuracy, and highest predicted return are selected to construct a monthly portfolio. The test size ratio used in both regression models is 1:4 (25%).

In order to evaluate the impact of technical indicators on predicted result, moving average indicator for *9, 20, 50, 150* days are used. Basic statistical values: *High-Low Percentage Changed (HL_PCT)* and *Open –Close Percentage Changed (OC_PCT)* are also added to the main stock's features. Therefore, there are two different datasets are used to evaluate the performance of both regression models. One is 500 stocks' data with the main features. Another is main features and the added technical indicators.

First, we calculate the prediction accuracy on average on each dataset, then select the model based on the higher prediction accuracy $\bar{P}$. We construct a portfolio by selecting 4 stocks with the highest predicted return as well as predicted accuracy higher than the average $\bar{P}$.

In order to maximize the return and minimize the volatility of the constructed portfolio, we use three different portfolio optimizers: Equal-weights (EQ) portfolio does not require any optimization technique since all the stocks in the portfolio are equally weighted. Monte Carlo Simulation (MCS) is used to find the optimal weights through thousands of randomly selected allocations. Mean Variance Optimization (MVO) is used to find adaptive weights portfolio which adapts the stock weights using the prediction models. The mapping is a mathematical function whose parameters are found through optimization as shown as in **Figure 4.**

5.3 Experiment Results

Model selection

As shown in **Figure 5**, the prediction accuracy on average decrease gradually by the prediction periods for both regression prediction models. In each period, the predicted accuracy is slightly higher with the support of technical indicators. Therefore, adding more features such as technical indicators could improve the predicted accuracy. In short-term trading, LR model obtains a higher accuracy than SVR one. However, SVR predicted accuracy tends to be higher in long-term prediction.

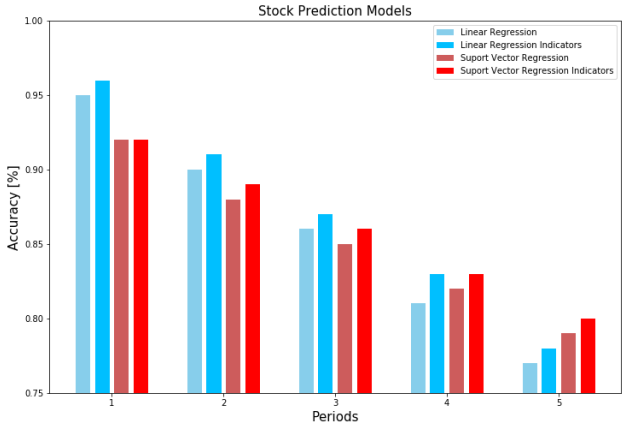

Figure 5: The prediction accuracy for regression models.

Table 2. Portfolio performance measurement results

Period	Optimizer	Prediction		Actual		S&P500 SPY	
		SR	R(%)	SR	R(%)	SR	R(%)
1	EQ	0.25	21.9	0.31	18.7	(1.77)	(3.6)
	MCS	0.55	22.4	0.32	13.7		
	MVO	0.55	22.4	0.33	13.7		
2	EQ	0.24	45.9	0.33	45.5	(1.14)	(5.2)
	MCS	0.89	47	0.23	38.5		
	MVO	0.96	47.3	0.25	38.5		
3	EQ	0.20	76.9	0.33	66.6	0	(0.4)
	MCS	0.38	58.7	0.31	57.0		
	MVO	0.39	59	0.31	57.4		
4	EQ	0.10	62	0.12	13.7	0.54	3
	MCS	0.12	78.7	0.10	13.9		
	MVO	0.13	79.7	0.11	13.7		
5	EQ	0.06	31.2	0.06	9.0	0.76	5.2
	MCS	0.08	61	0.07	9.8		
	MVO	0.08	61.6	0.07	9.6		

Portfolio Evaluation

Because LR obtains the higher predicted accuracy on average in the predicted period 1,2 and 3 then portfolio 1,2, and 3 are constructed based on the LR prediction results. Portfolio 4 and 5 are constructed based on the SVR prediction results.

As presented in **Table 2**, the short-term portfolio investment obtains higher Sharpe ratio (SR) and cumulative return. However, the predicted SR tend to be lower and less attracted in long-term investment. That can be caused by the low predicted accuracy leads to have higher risk or more volatility. The predicted optimal SR and cumulative return are nearly the same in both optimizers (MCS, MVO) and improve the results from EQ optimizer.

Despite the performance measures in actual are lower as predicted, all the portfolios obtain a profitable return and high SR. Especially the portfolio 3 with more than 60 percent cumulative return while the benchmark S&P 500 ETF-SPY lost 0.4 percent.

6 Conclusions and Discussions

This paper presents the fundamentals of the quantitative trading system in terms of system architecture, benefits, and trading workflows. The taxonomy of machine learning techniques and application in finance are also described in this work. Machine learning plays an important role and becomes the most powerful tool to build up trading strategies by improving the prediction accuracy. In the experiment, both linear regression and support vector regression models are used to predict the stock price. As the results, both regression prediction models are show effectively in prediction with high accuracy on average. The linear regression model performs better than support vector regression in the short-term prediction. However, the support vector regression model tends to perform better than linear regression in long-term prediction. Using indications should improve the prediction accuracy. To evaluate the confident of the prediction results, five different portfolios are constructed. Despite of the fact that, the Sharpe ratios and returns are not exactly as predicted, the proposed strategy seems to work pretty well by achieving attractive returns and high annualized Sharpe ratios in the short-term trading. All the portfolios are performed more effectively in compared to S&P 500 EFT-SPY. Back-testing and diversification are considered as the target for further research.

There are several challenges for building effective quantitative trading strategies through machine learning. First, market data is non-stationary while most machine learning techniques assume the data generating processing is stationary. Second, market data exhibit high noise to signal ratio. The prediction models can perform well on the historical data set. However, the stock market is always fluctuated by many factors such as market psychology, macroeconomics, even political issues. Therefore, high performance on the historical dataset does not guarantee to earn a desirable profit in practical. Third, back-testing is not only as the tool to evaluate the discovered strategy, but also help to avoid false positives. Finally, developing the flexible, efficient trading strategies are critically important for quantitative trading. It is the most challenge task in the quantitative trading system. Since the diverse of data sources and format, and different characteristics of data. That will make the prediction getting more complex. In summary, the advanced of computational resources, and machine learning enable to design an efficient quantitative trading system.

REFERENCES

[1] W. Huang, Y. Nakamori, S.Y. Wang (2005). Forecasting stock market movement direction with support vector machine, 32(10), pp. 2513–2522.

[2] E. Chong, C. Han, F.C. Park (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies, 83(), pp.187-205.

[3] R.K. Narang. 2009. *Inside the Black Box: The Simple Truth about Quantitative Trading*. Wiley Finance Press, New Jersey, Chapter 1.

[4] E. Chan. 2008. *Quantitative Trading: How to Build Your Own Algorithmic Trading Business* (1st ed.). Wiley Press, Chapter 3.

[5] P. Treleaven, M. Galas, V. Lalchand (2013). Algorithmic Trading Review, 56(11), pp.76-85.

[6] S. Russell, P. Norvig. 2009. *Artificial Intelligence: A Modern Approach* (3rd ed.). Pearson Press. Chapter 1.

[7] J. Michels, A. Saxena, A.Y. Ng (2005). High speed obstacle avoidance using monocular vision and reinforcement learning. In proceeding of the 22nd international conference on Machine learning. German, pp. 593-600.

[8] M. Abadi, P. Barham, el. (2016). TensorFlow: A system for large-scale machine learning. In proceeding of the 12th international conference on operating systems design, and Implementation. USA, pp.265-283.

[9] P. Wiriyathammabhum, D. Summers-Stay, C. Fermüller, Y. Aloimonos (2016). Computer Vision and Natural Language Processing: Recent Approaches in Multimedia and Robotics, 49(4), pp. 71

[10] T. Michell. 2010. *Machine Learning* (1st ed.). McGraw-Hill. p. 2.

[11] A. Kazem, E. Sharifi, F.K. Hussian (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting, 13(2), pp. 947-958.

[12] M. Wurzenberger, el. (2017). Incremental Clustering for Semi-Supervised Anomaly Detection applied on Log Data. In processing of the 12th international conference on availability, reliability and secure.

[13] E. Bigdeli, M. Mohammadi, B. Raahemi, S. Matwin (2017). A fast and noise resilient cluster-based anomaly detection. 20(1), 183-199.

[14] J. Kober, J. Andrew, J. Peters (2013). Reinforcement learning in robotics: A survey. 32(11). Pp.1238-1274.

[15] D.H. Wolpert, W.G. Macready (1997). No free lunch theorems for optimization, 1(1). pp. 67-82.

[16] Xiao Ding, Yue Zhang, Tin Liu, Junwen Duan (2015). Deep learning for event-driven stock prediction. *IJCAI*, 2327-2333.

[17] Jason W. Leung (2016). *Application of machine learning: automated trading informed by event driven data.* Doctoral dissertation, Massachusetts Institute of Technology.

[18] Charles X. Ling, Victor S. Sheng, and Qiang Yang, Senior Member (2006). Test Strategies for Cost-Sensitive Decision Trees. IEEE Transactions on Knowledge and Data Engineering(TKDE), 18(8), 1055-1067.

[19] Yong Hu, Kang Liu, Xiangzhou Zhang, Lijun Su, E. W. T. Ngai, Mei Liu (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. Applied Soft Computing, 36, 534-551.

[20] Omer Berat Sezer, A. Murat Ozbayoglu, Erdogan Dogdu (2017). An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework. *Proceedings of the SouthEast Conference*, ACM, 223-226.

[21] Van-Dai Ta, Chuan-Ming Liu (2016), Stock market analysis using clustering techniques: the impact of foreign ownership on stock volatility in Vietnam. *Proceedings of the Seventh Symposium on Information and Communication Technology.* ACM, 99-106.

[22] Muhammad Asad (2015). Optimized Stock market prediction using ensemble learning. *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on.* IEEE, 263-268.

[23] Thomas G. Dietterich (2000). Ensemble Methods in Machine Learning - Oregon State University. *International workshop on multiple classifier systems.* Springer, Berlin, Heidelberg, 1-15.

[24] Roberto Cervelló-Royo, Francisco Guijarro, Karolina Michniuk (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert systems with Applications*, 42(14), 5963-5975.

[25] F. Pedregosa et al. (2011), Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

[26] Markowitz (1987), Harry M. Mean-Variance Analyses in Portfolio Choice and Capital Markets. Oxford: Basil Blackwell, Inc.