A Thesis for the Degree of Ph.D in Engineering

Financial Time Series Prediction with Information Fusion

2014 December

Center for Open Systems Management
School of Science for Open and Environmental Systems
Graduate School of Science and Technology
Keio University

DENG SHANGKUN

Abstract

Our proposed prediction and learning method is a hybrid referred to as MKL-GA, which combines multiple kernel learning (MKL) for regression (MKR) and a genetic algorithm (GA) to construct the trading rules. In this study, we demonstrate that the evaluation criteria used to examine the effectiveness of a financial market price forecasting method should be the profit and profit-risk ratio, rather than errors in prediction. Thus, it is necessary to use a price prediction method and a trading rules learning method. We tested the proposed method on the foreign exchange market and stock market, and we tested MKR on crude oil market. The features used for prediction on FX market were extracted from the trading history of multiple markets and multiple time horizons, and the features used for prediction on stock market were from historical stock prices and volumes, as well as social network services (SNS). MKR is essential for utilizing the information contained in many of the features derived from different information sources and for various representations of the same information source. The GA is essential for generating trading rules, which are described using a mixture of discrete structures and continuous parameters. First, the MKR predicts the change in the FX market or stock market based on technical indicators such as the moving average convergence and divergence (MACD). Next, the GA generates a trading rule by combining the results of the MKR with several commonly used overbought/oversold technical indicators. For simulation, we show the application of MKL-GA on FX market and stock market, as well as application of MKL on two well-known crude oil markets. The experimental results show that the proposed method outperforms other benchmark methods in terms of the price prediction, returns and return-risk ratio.

Keywords: Financial Prediction, Multiple Kernel Learning, Genetic Algorithm, Hybrid Method

Contents

1. Introduction

Many business practitioners and researchers have developed various kinds of models to predict and analyze on financial time series, especially foreign exchange (FX) rate, stock prices and crude oil prices. The FX market is the most influential of the financial markets, especially in the current era of a rapidly expanding global economy, because of the vast number and values of its transactions, and the range of currencies that can be traded. In addition, there are many studies that estimate and predict the stock prices and stock volatility by using historical stock prices or volumes data. Stock market is also an important financial market that at the close of 2012, the size of the world stock market was about $55 trillion and lots of individual investors and investment companies invest in stock markets. Other than FX and stock market, crude oil is the world's most actively traded commodity, accounting for over 10% of total world trade **(Verleger, 1994)**. The reason for this large volume of trade in crude oil is two-fold: its key role in the world economy and the worldwide dependence on crude oil for meeting energy demands. The FX market, stock market, and crude oil markets are the financial markets whose influence cannot be avoided in life, not only for people those engaged in the financial filed, but also for entrepreneurs and policy-makers. If it was possible to predict the market to a certain degree, investors could gain profit, hedge their assets or policy-makers could amend their policies. Thus, prediction and trading in financial markets is an important research area.

In the last decade, numerous researchers have used machine learning technologies, such as support vector machine (SVM), genetic algorithm (GA) or some integrated models of them to predict stock price or exchange rate changes or to find trading rules for the stock, FX or crude oil trading, by mining historical price or transaction volume data. For example, **Kwok (2000) and Cao (2003)** applied an SVM to predict the stock price and obtained good results. **Shioda et al. (2011)** predicted the foreign exchange market states with SVM. **Fuente et al. (2006)** and **Allen and Karjalainen (1999)** applied GA to generate trading rules. **Chien and Chen (2010)** applied a GA-based model to mine associative classification rules with stock trading data. **Hirabayashi et al. (2009)** applied GA to finding trading rules for foreign exchange intraday trading by mining features from several technical indicators. **Dunis et al. (2013)** designed a SVM-GA hybrid model for stock price prediction and trading. Their proposed models obtained better performance than that of some conventional models; however, they utilized the features extracted from only the historical prices or transaction volumes.

In recent years, many researchers have used the multiple kernel learning (MKL) **(Bach et al., 2004; Sonnenburg et al., 2006)** method to address the problem of selecting suitable kernels for different feature sets. This technique mitigates the risk of erroneous kernel selection to some degree by using a set of kernels, deriving a weight for each kernel, and making better predictions based on the weighted sum of the kernels. One of the major advantages of MKL is that it can combine different kernels for different input features. Several researchers have applied MKL in their research fields. For example, **Joutou and Yannai (2009)** to applied MKL to food image recognition. **Foresti et al. (2009)** applied MK regression to wind speed prediction and their results outperformed those of some conventional methods. **Fletcher et al. (2010)** applied MKL to predicting the FX market based on the limit order book. **Luss and d'Aspremont (2009)** applied MKL to the prediction of abnormal returns from news using text classification. **Yeh et al. (2011)** applied MKL to predicting the stock prices on the Taiwan stock market, obtaining results that surpassed those of some conventional methods. Good performances in these literatures inspired us to use MKL to utilize the information from different sources. In this research, since the stock price and

FX rate movements may relate to not only the trading target (time series data or target time interval) but also other information sources (social network service or other FX pairs) or different representations of the same source (multiple time frames), to incorporate different types of features into a regression model, MKL is a very promising way for doing our task.

In addition, there is a high possibility for a stock (or other financial trading targets such as FX currency pair or crude oil futures) that has an upward/downward trend (to provide buy/sell signal, respectively) but it is in an overbought/oversold condition (to provide sell/buy signal, respectively), or vice versa. In order to solve the conflicts between trend signal and overbought/oversold condition, we design the trading strategy based on both trend signal and overbought or oversold conditions. Lots of previous research used SVM or MKL to find a trend signal, or use GA to find good trading rules. However, we thought that for yielding profit with smaller risk, we have better combine the signal of direction classification (go up/down) and overbought/overtrade condition signal (overbought, overtrade or neither of them). To do this, inspired by the hybrid SVM-GA model, we proposed a new hybrid model MKL-GA which combines MKL and GA. In the hybrid model, MKL is used to generate a trend signal, while GA is used to combine the trend signal and overbought/oversold condition signal.

Evaluation measures are very important to evaluate the performances of models. The root mean square error (RMSE) is a measure which is often used for evaluating prediction results. However, given that people will sell or buy stocks/FX/crude oil when they can predict the price/rate of it, the goodness of the predictions cannot be provided by differences of prediction and real values alone; a proper measure should be the trading profits based on the prediction. In addition, beyond the accumulated returns, most investors also pay close attention to the variability of returns. In other words, they hope the proposed model can increase profits as well as decrease associated risks while doing so. Therefore, to evaluate the appropriateness of prediction, we should not confine ourselves to RMSE and we should also use accumulated returns and returns to variability ratio or Sharpe ratio **(Sharpe, 1994)** that further considers risk free profits. In addition, for crude oil experiments in this research, since we conduct three different time horizons for prediction and trading, we used average profit percentage (APP) to compare the results of different time horizon prediction.

Section 2 of this thesis describes some well-known technical indicators, which are used widely by traders for technical analysis. A technical indicator for financial time series is a function that returns a value for given prices over a given length of time in the past. These technical indicators might provide traders with guidance on whether a FX/stock/crude oil commodity is being oversold or overbought, whether a trend will continue, and so on. In addition, the Section 2 also describes the evaluation measures (RMSE, accumulated return, Sharpe ratio and APP) used in this research, as well as providing background information for SVR, MKL, and GA.

In this study, the following three case studies were presented to show efficiency and feasibility of the proposed MKL-GA and MKL model for price/rate prediction and simulation trading for FX market, stock market, and crude oil markets.

Section 3 explains proposed hybrid method MKL-GA for short term FX rate prediction. This study can be summarized as follows: 1) we focused on predicting exchange rate changes over a short time horizon and we simulated trading using a trading rule, which was estimated and optimized with historical intraday data; 2) we employed a rolling window with a relatively short period of in-sample learning and of out-sample testing to follow rapid changes in trends, 3) we improved the sophistication of the trading rule, by integrating information from different sources using the predicted rate changes, a combination of order types, and some technical indicators; 4) we used many potentially useful variables to integrate information from different sources, including technical indicators for the different time horizons of different currency pairs; 5) we used multiple kernel learning for the large margin classifiers, without selecting variables manually; and 6) we used a GA to optimize the sophisticated trading rules.

Section 4 presents the experiments of proposed hybrid method MKL-GA for stock price prediction by proposed hybrid model. In summary, first, we extract features from both the time series data source and a social network source, in contrast to previous studies (e.g., **Choudhury et al., 2008**) which considered properties of social network to predict stock price movements, and **Luss and d'Aspremont (2012)** used text data of news for the prediction of abnormal returns. These literatures inspired us to attempt the prediction of stock prices by extracting features from time series data and a social network. Second, we use the MKR framework to optimally combine the features of time series data, news, and user comments, in contrast to other works (e.g., **Choudhury et al., 2008**) which used a single kernel for the SVM. Results from **Hann and Steurer (1996), Chen and Leung (2004), Kwok (2000), and Cao (2002)** prove that ANNs and SVMs (especially the latter) are good models to predict stock price change rates. However, given that the input features extracted from the time series of historical stock price change rates and those from a social network have different properties, we should consider using different kernels for input feature sets from different types of sources. However, it is not easy to assign good kernels manually. Therefore, we use MKL to solve this problem. Third, for generating trading signals, we use not only the predicted stock price change rates from the MKL model, but also three well-known overbought and oversold technical indicators. In addition, we consider thresholds over which the difference between the predicted value and current value should prompt an action: we buy if the combined decision value is greater than the buying threshold, and we sell if the combined decision value is less than the selling threshold. The best values of both thresholds are learned in the GA learning periods.

Section 5 provides the experimental of MKL method for crude oil price forecasting based on multiple crude oil markets and time frames. For the purpose of forecasting crude oil prices by considering features from different sources and different representations, we propose to extract and use the features from two main crude oil spot markets and three different time frames. The two markets in this context are WTI and Brent Crude oil markets, the two largest crude oil markets in the world. Although WTI crude oil is mainly supplied to North America and Brent Crude oil is mainly used in Europe, some interrelationship between these two markets cannot be ruled out, given the interdependence of worldwide oil markets in the highly integrated contemporary global economic system. For instance, the fluctuations in one market do not go unnoticed in the other market. Therefore, there is a strong case for referring to price movements in the other market for predicting crude oil prices in a particular market. In addition to extracting features from two different crude oil markets, the features of different time frames are also considered as useful information for prediction. In order to predict crude oil prices (WTI or Brent) in the target market, this study uses features from other crude oil markets besides features of the target market, and examines features from two time horizons other than the target time frame. Features from different sources or features of different time representations may have different properties and quality characteristics. The MKL model has been used in our study to address the problem of fusing information from different crude oil markets and time frames. For trading on crude oil, it is important to note that we did not consider genetic algorithm since the MKL based models have already yielded good performances.

2. Background

2.1 Technical Indicators

There are numerous influential trading technical indicators that are widely recognized and used by traders around the world. Some technical indicators are fairly straight forward to obtain and have proven successful in trading history. Among them, technical indicators such as the moving average (MA), rate of change (ROC), and moving average convergence and divergence (MACD) help traders to spot or follow trends, while the bias ratio (BIAS), Williams %R (WPR), and relative strength index (RSI) are used for identifying overbought and oversold conditions of a stock.

Table 2-1 shows the list of technical indicators used in this research. *Price(k)* refers to the closing price at time period *k*, *n* is the length of time frame to calculate values of indicator. Note that the indicators are applied to any time series including trading volume.

Table 2-1. List of technical indicators used in this research

Indicator	*Mathematical formula*	*Parameters*
Simple Moving Average (SMA)	$SMA_n(t) = \frac{\sum_{k=t-n+1}^{t} Price(k)}{n}$	*n* is the length of time frame
Exponential Moving Average (EMA)	$EMA_n(t) = Price(t-1) * a + (1-a) * EMA_n(t-1)$	Usually $a = 2/(n+1)$, n is the length of time frame
Rate of Change (ROC)	$ROC(t,N) = \frac{Price(t) - Price(t-N)}{Price(t-N)}$	*N* is the length of trading period.
Moving Average Convergence and Divergence (MACD)	$MACD(t) = EMA_{12}(t) - EMA_{26}(t)$ $MACDSignal(t) = EMA_9(MACD(t))$	
BIAS (bias ratio)	$BIAS_n(t) = 100 \times \frac{Price(t) - SMA_n(t)}{SMA_n(t)}$	*n* is the length of time frame
WPR(Williams %R)	$\%R_n(t) = \frac{Price(t) - high_{n\ periods}}{high_{n\ periods} - low_{n\ periods}} \times 100$	*n* is the length of time frame
RSI (Relative Strength Index)	$RSI_n(t) = 100 - \frac{100}{1 + RS_n(t)}$	$RS_n(t) = \frac{\text{Average of positive price changes in } n \text{ days}}{\text{Average of negative price changes in } n \text{ days}}$ *n* is the length of time frame

The MA is used to understand the present trend and thus, is a so-called trend-following index. It is used to emphasize the direction of a trend and to smooth out price fluctuations which are just random. The simple moving average (SMA) is a simple mean value with identical weights used for past prices, while the exponential moving average (EMA) is the average value of prices of a stock for a given length of time frame, attributing greater weight to newer changes and less weight to older ones.

The MACD is intended to predict changes in the market tendency. It provides two indicators: MACD and the MACD signal. MACD shows the difference between a fast and slow EMA of closing prices. "Fast" refers to a short-period average, and "slow," a long-period average. When MACD (t) is greater than 0, the short and steep uptrend is more influential than the long and gentle uptrend, which means that the stock price is likely to go up in the near future. Based on the default parameters, MACD is the difference between the 12-period and 26-period EMAs. The default values (12, 26, and 9) of MACD parameters can be changed based on the needs of the traders. In our research, we simply used the default values of MACD parameters because this value set is widely recognized and used in the world.

The rate of change (ROC) is a technical indicator showing the change rate between today's closing price and the closing price N days ago.

The BIAS, WPR and RSI are usually used to judge whether the stock is considered to be in possible oversold, overbought, or normal conditions. An extremely high or low value is a signal to the trader to buy when the stock is oversold and to sell when it is overbought. The parameter n of these indicators could be set by the traders.

2.2 Evaluation measures

To evaluate the goodness of performance of the models, we use the following four measures: root mean square error (RMSE) to evaluate the goodness of fit of models' prediction of price change rates, accumulated return and average profit percentage (APP) to evaluate the profit-making ability, and Sharpe ratio to evaluate ability to control the risk while yielding good profits.

The RMSE is a frequently used measure of the differences between the values predicted by a model or an estimator and the values actually observed from the entity being modeled or estimated.

In addition, we execute trading based on the trading signals that each model outputs and evaluate the return (loss or profit). In general, high return inevitably accompanies the potential for high risk. Therefore, we attempt to find a method that could decrease risk as well as increase profit. The Sharpe ratio is a measure of the excess return per unit of risk in an investment asset or a trading strategy, named after William Forsyth Sharpe (1994).

In addition to the magnitude measures, we also measure the usefulness of the prediction for making profits in trading. Suppose the predictor predicts the market price will go up (P_i is larger than R_i), we take up a long position in the financial market. If the predictor predicts the market price to go down (P_i is smaller than R_i), we take up a short position in the financial market. In this research, APP is used only for crude oil prediction, and it could be a useful measure of profitability for international companies, especially oil companies, airplane companies, and government energy organizations. Note that we do not consider transaction costs when calculating APP because unlike "trading" in stock or FX trading, we assume that government organizations and international companies trade crude oil without brokers. In the experiments, we do not do margin transaction, so the APP is calculated without leverage.

A summary of these three evaluation measures are shown in **Table 2-2.**

Table 2-2. Summary of evaluation measures

Evaluation measure	*Calculation*	*Description*
RMSE	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(PC(i) - ROC(i)\right)^2}{n}}$	$PC(\mathrm{i})$ is the predicted price change rate of target stock at time i, and $ROC(\mathrm{i}, 1)$ is the rate of change at time (i + 1), and n is the number of prediction times.
Accumulated return	$AC = \sum_{i=1}^{m} R_i$	R_i is the return in testing period i, and m is the number of testing periods.
Sharpe ratio	$S = \frac{E\left[R - R_f\right]}{\sqrt{var\left[R - R_f\right]}}$	R is the asset return, R_f is the return on a risk-free asset, $E\left[R - R_f\right]$ is the expected value of the excess of the asset return over the risk-free asset return, and $var\left[R - R_f\right]$ is the variances of the asset return. In our experiments, we used the Sharpe ratio as an evaluation criterion to evaluate the return-risk ratio performance of the models.
APP	$APP = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n} \frac{(R_{i+m} - R_i)}{R_i} \times sgn((P_{i+m} - R_i)(R_{i+m} - R_i))$ $sgn(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$	R_i is the price at time i, P_i is the prediction price at time *i.* Note that m denotes the steps of m-day ahead prediction (i.e., m=2 is for two-day ahead prediction). n is the number of prediction times.

2.3 Support Vector Regression and Multiple Kernel Learning

SVR is a version of the SVM **(Vapnik, 1995)** for solving regression problems with some distinct advantages. For example, SVR solves a risk minimization problem by balancing the empirical error and a regularization term, where the error is measured by Vapnik's ε-insensitive loss function. In addition, SVR usually estimates a set of linear functions defined in a high dimensional feature space. Furthermore, SVR is known for its ability to work well with many relevant features.

Generally, in a regression problem, suppose we are given a set of training examples:

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots (x_l, y_l)\}$$

where $x_i \in R^n, y_i \in R, i = 1,2, \ldots l,$ and each y_i is the output value for the input vector x_i, a regression model is learned from these patterns and used to predict the target values.

SVR is a kernel based regression method, which tries to locate a regression hyperplane with small risk in high-dimensional feature space. It possesses good function approximation and generalization capabilities. The ε-insensitive SVR is the most commonly used SVR. It finds a regression hyperplane with an ε-insensitive band. The image of the input data need not lie strictly on or inside the ε-insensitive band for making the method robust. Instead, images that lie outside the ε-insensitive band are penalized and slack variables are introduced to account for these penalties. In the following equations, SVR has been used to refer to ε-SVR. The objective function and constraints for SVR are as follows:

$$\min_{w,b} \quad \frac{1}{2}\langle \mathrm{w}, \mathrm{w} \rangle + C\sum_{i=1}^{l}\left(\zeta_i + \hat{\zeta}_i\right) \tag{2-1}$$

$$s.t. \begin{cases} (\langle \mathrm{w}, \varphi(x_i) + b \rangle) - y_i \le \varepsilon + \zeta_i \\ y_i - (\langle \mathrm{w}, \varphi(x_i) + b \rangle) \le \varepsilon + \hat{\zeta}_i \\ \zeta_i, \hat{\zeta}_i \ge 0, \quad i = 1, 2, \ldots, l \end{cases} \tag{2-2}$$

where l is the number of training samples, and C parameter is used for trade-off between model complexity and training error. ζ_i and $\hat{\zeta}_i$ are slack variables for errors that exceed ε.

Note that $\varphi(\cdot)$ is a possibly nonlinear mapping from the input space to a feature space. Also, $\langle . , . \rangle$ indicates the inner product of the arguments. The derived regression hyperplane is as follows:

$$f(x) = \langle \mathrm{w}, \varphi(x) \rangle + b \tag{2-3}$$

where w and b are weight vector and offset, respectively. In SVR, ε-insensitive loss function is described by

$$L_\varepsilon = \left| y_i - w * \varphi(x_i) - b \right|_\varepsilon = \begin{cases} 0 & \text{if } \left| y_i - w * \varphi(x_i) - b \right| < \varepsilon \\ \left| y_i - w * \varphi(x_i) - b \right| - \varepsilon & \text{othervise} \end{cases} \tag{2-4}$$

To solve **equation (2-4)**, one can introduce the Lagrangian, take partial derivatives with respect to the primal variables, set the resulting derivatives to zero, and turn the Lagrangian into the following Wolfe dual form:

$$\max_{a,\hat{a}} \quad \sum_{i=1}^{l} y_i (\hat{a}_i - a_i) - \varepsilon \sum_{i=1}^{l} (\hat{a}_i + a_i) - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\hat{a}_i - a_i)(\hat{a}_j - a_j) K(\mathrm{x}_i, \mathrm{x}_j) \tag{2-5}$$

$$s.t. \begin{cases} \sum_{i=1}^{l} (\hat{a}_i - a_i) = 0 \\ C \geq a_i, \quad \hat{a}_i \geq 0, \quad i = 1, \ldots, l \end{cases} \tag{2-6}$$

where $\hat{a}_i$ and a_i $i = 1, \ldots, l,$ are dual variables. Note that $K(\mathrm{x}_i, \mathrm{x}_j)$ is a kernel function, which represents the inner product $< \varphi(x_i), \varphi(x_j) >$.

Equation **(2-5)** can be solved by Sequential Minimal Optimization (SMO) **(Plat, 1998)**. Suppose $\hat{a}_i$ and a_i $(i = 1, \ldots, l)$ are the optimal values obtained; the regression hyperplane for the underlying regression problem is then given by the following:

$$f(x) = \sum_{i=1}^{l} (\hat{a}_i^* - a_i^*) K(\mathrm{x}_i, \mathrm{x}) + b^* \tag{2-7}$$

where $b^* = y_k + \varepsilon - \sum_{i=1}^{l} (\hat{a}_i^* - a_i^*) K(\mathrm{x}_i, \mathrm{x}_k)$ is obtained from any a_k^* with $0 \leq a_k^* \leq C$, and $f(x)$ depends only on training examples having non-zero coefficients (support vectors) through the representation of the kernel function K.

The SVR method uses a single mapping function φ and, hence, a single kernel function K. However, if the input vector features are from different sources or different representations of the same source, using a single kernel may not result in perfect fusion of the entire input information. Moreover, using different kernels for different input features may solve this problem and improve the forecasting performance. Therefore, instead of using a single mapping function, several mapping functions are combined to conduct aggregate mapping.

Furthermore, we use individual kernels for fusing the features from different sources or different representations. In addition to learning the coefficients $\hat{a}_i^*$, a_i^*, and b^* in equation **(2-7)**, the best combination of the coefficient of kernels are also learned by imposing a trace constraint or the entire kernel matrix:

$$K_{comb}\left(x,y\right)=\sum_{j=1}^{K}\beta_j K_j\left(x,y\right) \tag{2-8}$$

$$\text{with } \beta_j \geq 0, \ \sum_{j=1}^{K}\beta_j = 1 \tag{2-9}$$

where β_j are coefficients to combine sub-kernels $K_j(x,y)$. MKL will estimate the optimal coefficients from the training data. By preparing one sub-kernel for each feature set and estimating coefficients by MKL, we obtain the optimal combined kernel. The multiple kernel regression can be expressed as follows:

$$f(x)=\sum_{i=1}^{l}(\hat{a}_i^*-a_i^*)\sum_{j=1}^{K}\beta_j K_j\left(x_i,x\right)+b^* \tag{2-10}$$

A normal SVM is applied to a single feature type. In our experiments, we used one Gaussian kernel and one linear kernel for each feature set, and we used MKL to integrate the features of different markets and time frames. With MKL, we trained an SVM with an adaptively weighted combined kernel, which fuses different kinds of features.

Sonnenburg et al. **(2006)** proposed an efficient MKL algorithm to simultaneously estimate the optimal weights and SVM parameters by iterating the training steps of a normal SVM. In our experiments, we used the MKL library included in the Shogun toolbox **(Sonnenburg et al., 2010)**.

2.4 Genetic Algorithm (GA)

Goldberg (1989) provided an excellent discussion on the use of GAs for solving optimization problems. GAs start with an initial set of feasible solutions, called a “population.” The individual solutions in the population are known as chromosomes. Each chromosome, in turn, is made up of a number of genes that encode representations of a part of the solution. In every iteration (referred to as a “generation” in GA terminology), the current population evolves using reproduction strategies such as crossover and mutation. A fitness function is used to evaluate the chromosomes, and the survival of a chromosome from one generation to the next is biased in favor of the fittest chromosomes. In addition to reproduction strategies, an elitist strategy can be used to propagate the fittest chromosomes to the next generation. A combination of these strategies helps the population improve from generation to generation until the fittest member of the population represents a near optimal solution.

Steps 1 to 6 show the procedures of the GA based on **Goldberg (1989)**.

Step 1: Initialization.
Step 1 generates the initial population.
Step 2: Evaluation.
After the initialization step, each chromosome is evaluated using a fitness function.
Step 3: Selection.
Selection is a process in which suitable chromosomes are chosen from the parent population for the next generation. In this step, we adopt the tournament selection procedure. This step is repeated until the number of chromosomes selected is equal to the size of the population. In order to ensure the propagation of elite chromosomes, this model uses the Elitism Mechanism. This mechanism selects *P%* individuals, which have the relatively best fitness values, to be the offspring in the next generation, while the remaining individuals go through the genetic operations.
Step 4: Crossover and mutation.
Crossover operates by swapping the corresponding segments of a string representation of the parents and extends the search for a new solution. Mutation is a genetic mechanism. It randomly chooses a member of the population and changes one randomly chosen bit in its bit string representation.
Step 5: Evaluation.
Each chromosome is evaluated using the designed fitness function.
Step 6: Examination of termination criteria.
Steps 2 to 5 are repeated until the termination criteria are satisfied. The proposed algorithm is terminated if the maximum number of generations is achieved or if a solution with the highest fitness has remained unchanged for several generations.

3. Short-term foreign exchange rate prediction by MKL-GA method

3.1 Literature review

The foreign exchange (FX) market is the most influential of the financial markets, especially in the current era of a rapidly expanding global economy, because of the vast number and values of its transactions, and the range of currencies that can be traded. The FX market is the one whose influence cannot be avoided in life, not only for people those engaged in the financial field, but also for entrepreneurs and policy-makers. If it was possible to predict the market to a certain degree, investors could hedge their assets or policy-makers could amend their policies. Thus, prediction is an important research area.

A wide variety of information affects price movements in the market, but not all of these to be considered if the market is efficient. The market or market participants will combine all of the information instantly if the market is efficient, so prices will be fixed to their proper values with only probabilistic fluctuations. Opinions differ about whether the financial markets of economically developed countries are really efficient. If a market is efficient, it is not possible to make correct predictions of the prices in the market based on the price history. The efficiency is realized only if rational traders, instantaneously and correctly react to a change in a manner that optimizes the outcomes, which adjusts the prices instantaneously to their proper positions. However, real humans are not necessarily rational and have limited capacities; hence, they contribute to inefficiencies in the markets in which they participate. An abundance of research suggests that there is inefficiency in FX markets. A previous study concluded that profits based on the moving average crossover rule and daily data declined from 1971 into the 2000s **(Olson 2004),** whereas studies of intraday data have produced mixed results (**Schulmeister 2007; Dempster and Jones 2002; Gencay et al. 2003; Riccardo et al. 1997; Neely and Weller 2003**). Thus, if the market has not been efficient since 2000, trading rules based on moving averages and Alexander’s filter have not been able to exploit the inefficiency. In this study, we focused on simulated intraday trading in the 2000s using more sophisticated trading rules and a more versatile prediction function with more endogenous feature variables (explanatory variables) to determine whether trading based on these rules would have yielded excess profits. We assume that the market is largely efficient so exogenous information is absorbed into market behavior, despite the slight inefficiencies attributable to the irrational behavior of traders. Therefore, the rates themselves and the indexes calculated from these rates were the only values that we needed to consider as explanatory variables.

Many researchers have failed to obtain excess profit from the markets since 2000, possibly because of the long period required for the optimization or estimation of the parameters required by the trading rules. A longer period of optimization (in-sample estimation) produces better trading rules, but only if the time series is stationary. However, information creation, transmission, and sharing are changing more rapidly than ever because of the rapid development of the Internet, proprietary networks of markets, and powerful computers, which means that any market trends have short periods and they are harder than ever to find.

We used moving windows in this study, but they were much narrower in width than those used in other studies (**Meese and Rogoff, 1983; Pesaran and Allan, 1995; Hann and Steurer, 1996; Yeh et al,. 2011**). The windows are the periods used for in-sample estimations and out-of-sample tests. Unlike rolling regression, the in-sample estimation period rolls forward along with the out-of-sample test period. The lengths of the in-sample estimation period and out-of-sample test period were fixed throughout our experiments. In this study, the learning period was 1000 trading hours (around eight weeks) and the out-of-sample test period was 500 trading hours (around four weeks).

We used a genetic algorithm **(Goldberg, 1989)**, a class of meta-heuristic algorithms, to optimize the trading rules. The trading rules used by many researchers are based on moving averages or Alexander's filter, where the parameters are estimated during the in-sample estimation period. Our trading rules were more complex and there was no closed form for estimating the parameters, so we had no option other than "training" our rules using a machine learning technique. Thus, we had to optimize specific objective functions by varying the parameters. In this study, we focused on the maximization of the profits and the profit-risk ratio. These objective functions (i.e., the profits or profit risk ratio) were discontinuous functions of a series of exchange rates, so we could not develop an iterative procedure to optimize the parameters by assuming gradual or continuous changes in the parameters due to gradual or continuous changes in the objective function. Therefore, we had to apply meta-heuristic methods, such as genetic algorithms or evolutionary algorithms.

Let us consider how we can make a trading rule more sophisticated. A trading rule is a rule that mimics how traders decide whether to buy or sell by observing and recognizing any patterns that emerge in the market behavior. We assume that a rule comprises the following: predictions of price changes; when and how to open a position based on the predicted, current, and historical values; and the condition where the position is closed by a limit or stop order based on the current price. In our study, prediction was independent of other factors, including leverage, limit orders, and stop orders, so the prediction function was identified initially before the trading rule was optimized.

The position taken by a trader is either long or short. A long (or short) position means that a trader buys (or sells) one currency (the base currency of a pair) and expects to sell (or buy) the currency in the near future at a higher (or lower) rate. A trader who buys a currency expects that the exchange rate of that currency will rise relative to a quoted currency in the near future. During simulated trading, the condition of the trading rule is examined when a fixed time period has passed after the position was taken. If the position to be taken at the time is different from the one taken at the previous time, the previous position is closed and a new position is opened. If not, the previous position is maintained. This periodic examination is a model for real traders who check the market behavior at fixed time intervals. If the market moves rapidly in an adverse direction, the trader may lose capital over a short period. To minimize losses, we can place a stop or stop-loss order. If the market moves rapidly in the expected direction but then recedes before the check time arrives, we miss a chance to take profits. To minimize the loss of opportunities, we can place a take profit order.

Leverage is another method used by the trading rule. A trader wants to invest variable amounts of capital depending on the fluctuating conditions. The trader will use leverage to invest more when they are more certain of a prediction than that when they uncertain. In the FX market, traders use a maximum leverage of 10 to 1 or 100 to 1, depending on the brokers and regulations. It should be noted that the effectiveness of leverage has been proven empirically **(Sermpinis et al. 2012).**

We reasoned that more explanatory variables (or feature variables) than those used currently by many researchers should be incorporated into the modeling functions, so the information underlying these variable would be integrated to facilitate reliable predictions and trading rules.

In current studies of prediction, the prices or their moving averages are recognized widely as useful information for making predictions. Some studies also incorporate information from the stock market or macroeconomic variables for FX prediction, such as the GDP or interest rate, but in this study, we focused on utilizing information from other currency pairs and other time horizons.

Researchers around the world have used technical indicators for many years to predict stock prices or FX rates **(Gehrig and Menkhoff, 2006; Park and Irwin, 2004)**. The results obtained using various adaptations of technical indicators to predict stock prices and FX rates are highly variable, but we decided to include some of these technical indicators in our proposed model because many traders who build the market use them.

We also incorporated variables related to different time horizons because different trading behaviors may appear over different time horizons. We assumed that some traders would receive information from outside at the same time as the other traders but their reaction times to the information may be variable. Observing and integrating the outside information and reacting to it over different time scales correspond to observing the market behavior and combining its descriptions over different time horizons. Traders often watch many different charts on displays while trading but the charts may have different time scales. Traders flip the time scales and watch the movements of prices over various time scales. The resulting decision may depend on charts with various time scales. In previous studies where the prediction horizon was set to one month, the moving average was calculated with a unit of one month, but we assumed that a price change in one month might be affected by an emerging trend at a one-month scale but also by emerging trends at two-month or half-month scales. Thus, to predict 1 hour ahead, we calculated a moving average every 2 hours or 30 minutes, in addition to a 1 hour moving average. Other time horizons could be used, e.g., 4 hour or 15 minute trends, but we could not search for a wide range of alternatives so we had to limit ourselves to a small range of choices.

We also used the rates of currency pairs other than the target currency pair as explanatory variables. For example, EUR/USD was used to predict USD/JPY. This is because a currency pair in a FX market is like a security on a stock market because all of the main currency pairs are correlated due to the fact that traders watch the movements of the rates between these currencies when opening or closing positions. Various Internet sites show real-time and historical correlations **(Online Source 1; Online Source 2; Online Source 3).**

Many researchers have studied whether there are correlations among different currency exchange rates **(Wu, 2007; Lee, 2003; Drozdz et al., 2007; Mizuno et al., 2006; Kwapien et al., 2009)**. However, the existence of a correlation does not necessarily mean that the exchange rates of another currency pair can be used to predict the target currency pair. Nonetheless, we decided to include variables calculated from other currency pairs because we could not exclude the possibility of their predictability.

The prediction function we used, which is explained below, was support vector regression (SVR) where the feature variables were derived from historical exchange rates. Artificial neural networks (ANN) are the most popular nonlinear prediction functions. In the machine learning field, large margin classifiers such as support vector machines (SVMs) are also known to obtain better out-of-sample classification accuracies in many cases.

SVR, an extension of SVM to regression problems, is known to deliver comparable or better out-of-sample accuracies than ANN. We decided to use SVR in our study because it allowed us to apply different nonlinear functions to different sets of explanatory variables, possibly by using different kernel functions in the SVM and SVR, i.e., by extending these to multi-kernel SVM and SVR.

In particular, we recognized the versatility of the multi-kernel method and the importance of the sparsity of representation produced by large-margin classifiers such as SVM. In general, potentially useful variables may increase the prediction accuracy when they are added to explanatory variables. If we add more than necessary, however, the mutual dependences between variables may have adverse effects. Thus, variable selection or feature selection are necessary. However, large margin classifiers are known not to decrease the accuracy even when there are many relevant and irrelevant variables **(Joachims, 1998)**. In other words, it was not necessary to select variables for large margin classifiers and regressions.

Large margin classifiers solve nonlinear cases using nonlinear kernels. If the feature variables have their own distinct characteristics, it is not reasonable to apply one kernel function to the features and create a classifier or a predictor.

A multiple kernel method was proposed to solve this problem, which uses linearly combined kernel functions instead of a kernel function in the SVM and SVR settings. In our study, a moving average and a technical indicator relative strength index had different natures, so the best kernels for these diverse variables should be also different. Therefore, we formed groups of similar features, each of which corresponded to a kernel function, before we applied multiple kernel learning.

This study can be summarized as follows: 1) we focused on predicting exchange rate changes over a short time horizon and we simulated trading using a trading rule, which was estimated and optimized with historical intraday data; 2) we employed a rolling window with a relatively short period of in-sample learning and of out-sample testing to follow rapid changes in trends, 3) we improved the sophistication of the trading rule, by integrating information from different sources using the predicted rate changes, a combination of order types, and some technical indicators; 4) we used many potentially useful variables to integrate information from different sources, including technical indicators for the different time horizons of different currency pairs; 5) we used multiple kernel learning for the large margin classifiers, without selecting variables manually; and 6) we used a genetic algorithm (GA) to optimize the sophisticated trading rules.

The use of a GA to search for the best parameters corresponds to finding the best models in a probabilistic manner. However, the search is not exhaustive, so it is possible that a much better model still exists.

3.2 Background

The FX market is considered to be the largest financial market in the world. This market has the advantage that its traders can trade practically around the clock on business days. Therefore, compared with trading in stock market, traders have more chances to trade in FX markets. However, compared with stock markets, the FX market is also a high-risk market.

To predict FX rates directly, researchers have used methods from statistics and signal processing, such as the autoregressive (AR) model **(Champernowne, 1948)**, the autoregressive moving average (ARMA) model **(Box and Jenkins, 1994)**, and the autoregressive integrated moving average (ARIMA) model **(Box and Jenkins, 1994)**. ARMA model is a special case of ARIMA model where the degree of integration is 0. These models are not sufficiently powerful for stock or FX market forecasting **(Ravindran et al., 2008)**, probably because stock time series and FX time series have different characteristics than these models assume. In addition, autoregressive conditional heteroskedasticity (ARCH) for modeling volatility was proposed by **Bollerslev (1986)**, which is, when combined with ARMA/ARIMA, considered to be more effective for modeling the dynamics of FX rates. However, the experimental results obtained using these methods are highly variable **(Bonilla et al. 2011; Ravindran et al. 2008)**. Thus, we decided to pursue the use of nonlinearity for expressing exchange rate. From a methodological point of view, ANNs are alternative methods for describing regression functions, where the space of the regression function is difficult to assume. Many studies have shown that ANNs significantly outperform linear models such as ARMA and a naïve random walk model **(Hann and Steurer, 1996; Zhao et al., 2009; Chen and Leung, 2004; Kodogiannis and Lolis, 2002; De et al., 2009; Koskela et al., 1997; Mehdi K. and Mehdi B., 2011)**, while there are reports of good FX rate forecasting performance with ANN **(Wong and Selvi, 1998; Azoff 1994)**. The most commonly used types of neural networks are feedforward networks with sigmoid functions or radial basis functions (RBFs).

SVMs are alternatives to ANN, which sometimes perform better **(Kwok, 2000; Cao, 2003)**. SVMs are renowned for its ability to perform well when there are many relevant features **(Joachims, 1998; Pai and Lin, 2005)** designed a hybrid method, which combined ARIMA with SVM. SVMs are known to be robust to the overlearning caused by many relevant or irrelevant features, so they can improve predictions that use many features. We verified the validity of using a SVM with many features but we concluded that it would be better to try different learning kernels for different sets of features after we analyzed the results of preliminary experiments where we used different kinds of features, as described below.

In recent years, many researchers have used the multiple kernel learning (MKL) **(Bach et al., 2004; Sonnenburg et al., 2006)** method to address the problem of selecting suitable kernels for different feature sets. This technique mitigates the risk of erroneous kernel selection to some degree by using a set of kernels, deriving a weight for each kernel, and making better predictions based on the weighted sum of the kernels.

One of the major advantages of MKL is that it can combine different kernels for different input features. Several researchers have applied MKL in their research fields. For example, **Joutou and Yannai (2009)** to applied MKL to food image recognition. **Foresti et al. (2009)** applied MK regression to wind speed prediction and their results outperformed those of some conventional methods. Recently, some researchers have applied MKL to predicting the FX and stock markets. For example, **Fletcher et al. (2010)** applied MKL to predicting the FX market based on the limit order book. **Luss and d'Aspremont (2009)** applied MKL to the prediction of abnormal returns from news using text classification. **Yeh et al. (2011)** applied MKL to predicting the stock prices on the Taiwan stock market.

A technical indicator for FX rates is a function that returns a value for given FX rates over a given length of time in the past. These technical indicators might provide traders with guidance on whether a currency pair is being oversold or overbought, whether a trend will continue, and so on. The moving average is the simplest and best-known technical indicator. It is also the basis of many other trend-following or overbought/oversold indicators. The moving average is inherently a follower rather than a leader, but it reflects the underlying trends in many cases.

There are many well-known advanced technical indicators such as moving average convergence/divergence (MACD) **(Stawicki, 2007)**, the relative strength index (RSI) **(Wilder, 1978)**, Williams %R **(Williams, 2005)**, and the bias ratio (BIAS). Williams %R and BIAS are both based on a moving average and they have the specific abilities to provide an early warning of overselling or overbuying. We extracted features such as these technical indicators from the time series data of currency pairs other than the target pairs. Previous studies of the prediction of FX rates generally considered only the target currency pair. However, there should be various correlations between the target currency pair and other currency pairs in the financial markets because these currencies are traded against each other in the global market.

In addition, we extracted features from the same time series data, but over different time frames. Features with a longer or shorter time horizon can also be useful for predicting a trend or a rate with a 1-hour time horizon. Therefore, features at 30-min and 2-hour scales are extracted from three or five FX currency pairs. Most previous studies have only used the features from the target trading time horizon, e.g., only the daily time series data are used to predict the rate one day in advance.

Evaluation measures are very important for gauging the accuracy and efficiency of prediction methods. The hit ratio and RMSE are often used to evaluate predictions of FX rates and stock prices. The hit ratio is the proportion of correct predictions about changes in direction made by a predictor. However, even if our predictions have a large hit ratio, we might not make a profit if we fail to predict the magnitudes of changes accurately. Similarly, even if the RMSE of the proposed model is small, we might not make a profit without predicting the directions of changes accurately.

The action taken when someone can predict the FX rate accurately is to buy or sell, so the appropriate measure of the correctness of the predictions should be the profit. In any simulation, there should be a trading strategy or rule that simulates the FX transactions on which the profit depends.

In this study, we measured the quality of predictions made with trading strategies for buying and selling foreign currencies over short periods (our target time interval was one hour), where we assumed that we started trading with a certain amount of Japanese yen. We applied MK regression to predicting the exchange rate changes using features such as MACD indicators, before we applied the GA to search for a trading strategy with an optimal combination of the MK regression results and the values of some overbought/oversold technical indicators. We include leveraging (a trading tool) when we place an order. We evaluated the profit with the proposed method and the performance in terms of the return-risk ratio.

In our trading strategy, we include the predicted FX rate changes and the overbought/oversold technical indicators. We also included thresholds that prompted specific actions. We bought if the combined value from the prediction and the overbought/oversold indicators was higher than the buying threshold, whereas we sold if the combined value was lower than the selling threshold. Furthermore, to control the risk from an unexpected jump in the FX rate, we also use take-profit and stop-loss orders, and we include their levels in the trading strategy.

We used a GA to learn the trading strategies, where the resulting profit was the fitness value of the chromosomes, because the profit from transactions comprised many discrete events. In addition, our GA design considered different leverages for buying and selling and it had settings for profit-taking and loss-stopping. Thus, our model was more complex than those used in previous studies so it was expected to perform better.

GAs have been used successfully for economic and financial prediction **(Frick et al., 1996)**. A specific feature of the GA is that its hypothesis space is discrete and structured. The GA uses techniques inspired by evolution, i.e., inheritance, mutation, selection, and crossover **(Banzhaf et al., 1998)**. Many researchers have made predictions about the stock market or FX market using GAs and they have reported good performance. **Fuente et al. (2006)** used a GA method for trading on stock markets, while **Allen and Karjalainen (1999) and Hirabayashi et al. (2009)** employed GAs to generate trading rules. **Badawy et al. (2005)** used a GA to select appropriate technical indicators at a particular trading time. **Allen and Karjalainen (1999)** reported an indicator combination scheme that delivered greater returns than a simple buy-and-hold strategy when the transaction costs were below a certain threshold. **Chu et al. (2000)** proposed an intelligent trading advisor based on several indicators and historical stock prices.

In summary, we first predicted the target FX rate changes using features extracted by multiple kernel regression from a time series of the exchange rates of the target currency pair and other currency pairs. Next, we used the GA to learn the trading strategy where the parameters were related to the predicted FX rate changes, overbought/oversold technical indicators, buying and selling thresholds, and the levels of take-profit and stop-loss orders. Our evaluation measures were the returns and return-risk ratios of the trades based on the trading strategy.

In this study, we assumed that the dynamics of the FX market changed slowly and they were considered to be relatively stable for at least 1500 trading hours (around 3 months or 12 weeks). However, this property is not supported by some previous studies. Indeed, there is evidence that the money market can change abruptly. However, our experimental results demonstrated that the proposed system obtained steady profits so this assumption was not unrealistic.

3.3 Proposed method

3.3.1 Trading Strategy of proposed method

The signals generated by various indicators might not always be in agreement, so it is necessary to develop a mechanism that resolves any conflicts that might occur.

The final decision, *D*, is a linear combination of the overbought/oversold indicators and the FX rate changes predicted from the MK regression:

$$D = \sum_{i=1}^{N} w_i e_i \tag{3-1}$$

where w_i are the weights learned by the GA and e_i are the values of the MK regression, as well as the values of the overbought/oversold technical indicators we considered (RSI, William %R, and BIAS). Note that the indicators we used were in ratio forms, i.e., RSI/100, WPR/100, and BIAS. Furthermore, it should be noted that the MK regression outputs are predicted FX rate changes or stock price changes. According to these conventions, e_i in equation **(3-1)** are all dimensionless so they are consistent.

Exchange rates do not change as much as stock prices. Therefore, it is too difficult to obtain high returns from the FX market without leverage. In our experiments, we used leverage in the same way as traders. However, this makes the risk of FX trading very high, so we had to implement a mechanism to control the risk. Thus, we set the stop-loss level and take-profit level relative to the contracted price and implemented these as features in the GA, i.e., as genes in the GA chromosomes.

After the weights and other parameters were learned by the GA, we obtained the decision value, *D*, which was related to the hours covered by the testing period. If the value of *D* was less than or equal to the threshold value for buying, we bought with the buying leverage. If the value of *D* was greater than or equal to the threshold value for selling, we sold with the selling leverage. If the value of *D* is between the threshold values for buying and selling, we did not open a new position. In addition, because the target prediction horizon was one-hour, our trading strategy closed the position one hour after we opened the position. However, if a trading signal was the same as the trading signal from one hour earlier, our trading strategy held the position until the next hour.

3.3.2 Features used for multiple kernel learning

USD/JPY was selected as the target currency pair in our experiments. When traders trade currencies, they consider the target currency pair and changes in other important currency pairs. In addition, traders watch different time horizons for indicators, rather than the target trading period alone.

Therefore, we extracted features from different currency pairs and different time horizons. The historical FX data we used were list of deals with time stamps in seconds in ICAP data or 1-min interval data, which contained the opening, high, low, and closing values for the interval in Forexite data. We processed these data and obtained the values of indicators with different time horizons at the same time point.

In our experiments, the target trading currency pair was USD/JPY and the trading period was one hour. First, we used three main currency pairs as the input feature sources: GBP/USD, EUR/USD, and USD/JPY. This was simply because GBP/USD and EUR/USD are the other two most widely traded currency pairs in FX markets. In addition, AUD/USD and USD/CHF are two direct currency pairs traded commonly in FX markets. For comparison, we considered AUD/USD and USD/CHF, as well as the three main currency pairs, as input feature sources in the experiments using our proposed method.

We calculated the 1-hour, 30-min, and 2-hour indicators for each time point of the 1-hour time horizon for the USD/JPY currency pair. **Table 3-1** shows the MK regression input features for each currency pair with each time horizon. The number of dimensions is eight.

Table 3-1. Input features for each currency pair and time horizon

No.	*Feature*	*No.*	*Feature*
1	MACD-value at time t	5	MACD-value at time $(t-2)$
2	MACD-signal at time t	6	MACD-signal at time $(t-2)$
3	MACD-value at time $(t-1)$	7	MACD-value at time $(t-3)$
4	MACD-signal at time $(t-1)$	8	MACD-signal at time $(t-3)$

For each time point over the USD/JPY 1-hour time horizon, we calculated the MACD indicators for three different currency pairs with three different time horizons. Finally, we obtained eight input features for the three different currency pairs with the three different time horizons, and we had one target variable: the exchange rate for the USD/JPY 1-hour time horizon. The output of the MK regression was the change rate per 1 hour for USD/JPY, as follows.

$$Change\_Rate = [Price(t+1) - Price(t)] / Price(t) \tag{3-2}$$

For each input feature set of a currency pair and a time horizon, we assigned one Gaussian kernel and one linear kernel, and we assigned the default values to the parameters of the Gaussian kernel.

3.3.3 Design of GA chromosomes

We designed the chromosome shown in **Table 3-2** for the trading strategy.

Table 3-2. GA chromosome design

No	*Length*	*Value Range*	*Meaning*
1	5 bits	−1 to 1	RSI weight
2	5 bits	−1 to 1	WPR weight
3	5 bits	−1 to 1	BIAS weight
4	5 bits	−1 to 1	MKL weight
5	5 bits	1 to 32	Leverage ratio for buying
6	5 bits	1 to 32	Leverage ratio for selling
7	5 bits	1% to 5%	Take-profit point in percentage
8	5 bits	1% to 5%	Stop-loss point in percentage
9	5 bits	−3.3 to 3.3	Threshold of D for buying
10	5 bits	−3.3 to 3.3	Threshold of D for selling
11	5 bits	1 to 32	Parameter of RSI
12	5 bits	1 to 32	Parameter of WPR
13	5 bits	1 to 32	Parameter of BIAS

The genes were represented as follows.

1) Numbers 1 to 4 (20 bits) represent the weights of the indicators and the MK regression results. The range of the weights for all of the indicators and the MK regression value was from −1 to +1, where the least significant bit represents 2/32 = 0.0625.
2) Numbers 5 and 6 (10 bits) represent the buy and sell leverages. The type was integer and the values ranged from 1 to 32.
3) Numbers 7 and 8 (10 bits) represent the take-profit and stop-loss levels for the contracted price in terms of the ratio relative to the current rate. The percentage for the take-profit and stop-loss relative to the contracted price ranged from 1% to 5%.
4) Numbers 9 and 10 (10 bits) represent the threshold values for buying and selling. Each threshold ranged from −3.3 to +3.3. In our GA design, we set the constraint that the buying threshold was less than the selling threshold.
5) Numbers 11 to 13 (15 bits) were used for the RSI, WPR, and BIAS parameters. The values ranged from 1 to 32.

In the GA training step, we set the population size to 200 and the maximum number of generations to 100 because we considered that there was a trade-off between the time cost and optimization performance. Another termination criterion was that the fitness value of the best individual did not improve for 10 successive generations. We initialized these individuals with random chromosomes according to the gene structure. To keep very fit individuals, the elite 10% (the top 10% of individuals in terms of fitness) were reserved automatically at each generation. The fitness value was the profit accumulated during GA learning.

3.3.4 Design of the proposed method

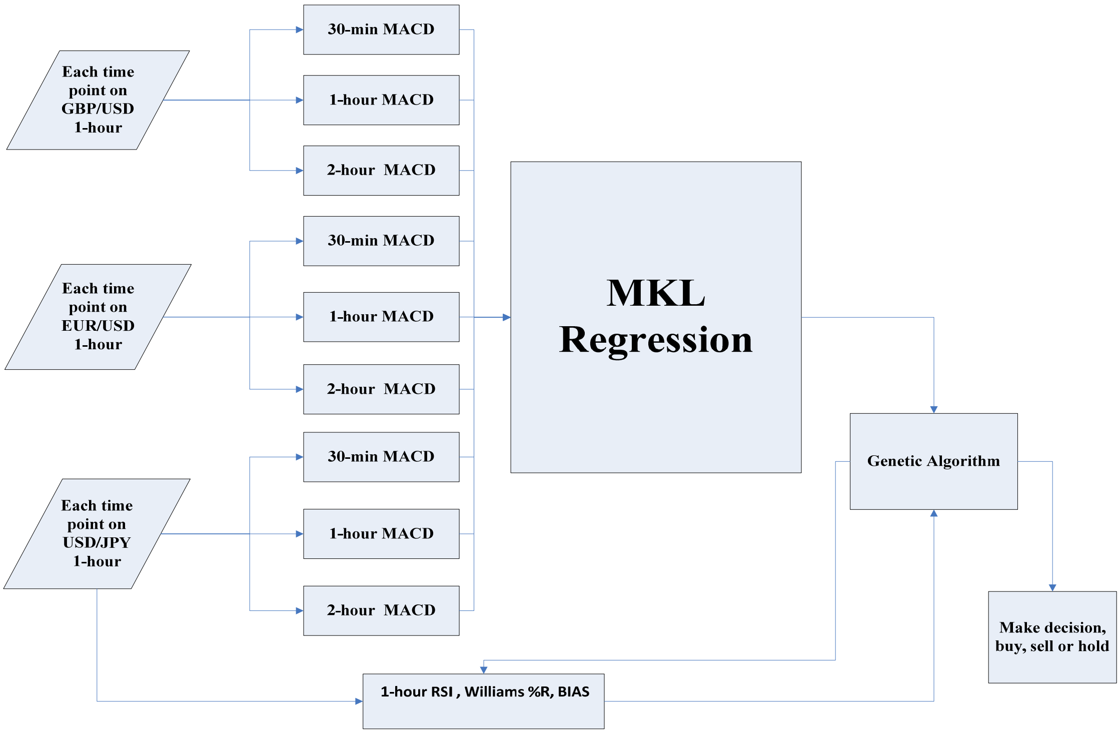

Fig.3-1. Diagram of proposed hybrid method

Fig. 3-1 shows our proposed hybrid method, while **Fig. 3-2** is a flowchart of the MKL-GA hybrid method. The following are the training and testing procedures used by the proposed hybrid method.

1) We obtained the exchange rate change predictions for the 1-hour USD/JPY one step ahead by applying multiple kernel regression.
2) We created a set of random values for the chromosomes as the first generation.
3) For each chromosome, we applied the trading strategy to the training data at each specified time during the training period by calculating the overbought/oversold technical indicators, computing the decision value *D*, and making decisions.
4) We computed the profits accumulated during the trading period as the fitness value. Tournament selection was used as the selection method for the GA in our system. This method selected a number of individuals from the population. A “tournament” was performed and the winner was selected to perform the crossover. In addition, the top 10% chromosomes (those that reached the top 10% in terms of profit) were retained directly for the next generations. New chromosomes were created by applying a crossover operation (we used two-point crossover as our crossover method) to the chromosomes selected from the current generation. Mutation was performed by converting chromosome’s 0 to 1, or 1 to 0. Crossover was repeated until a new generation was produced. Portions of the chromosomes were mutated or flipped randomly. The probabilities of crossover and mutation were 60% and 1%, respectively.

5) Steps 3 and 4 were repeated until the maximum number of generations (100) was reached or the fitness of the best individual did not improve for 10 successive generations. The best chromosome was then selected to represent the optimized trading strategy.
6) We calculated the profit and loss by applying the resulting trading strategy to the testing data.

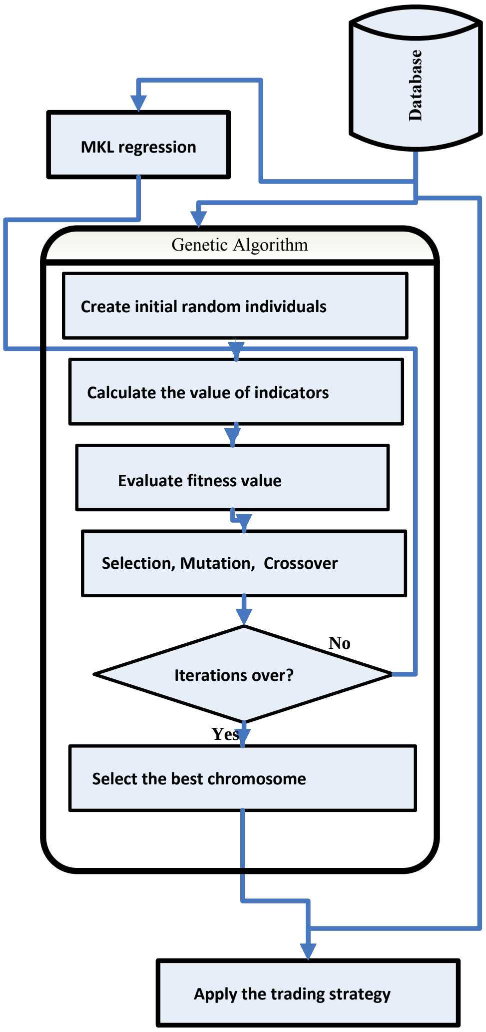

Fig.3-2. Flowchart of the MKL-GA hybrid model.

3.4 Experiment design

3.4.1 FX data

The exchange rates used in our study were obtained from ICAP and Forexite. The ICAP data comprised a list of best offers, best bids, and dealt prices, if any, for every second. These were very reliable but were limited to USD/JPY, EUR/USD, and GBP/USD from 2008 to 2011. The Forexite data were freely downloadable from their website and have been used by many researchers. For 2008 to 2011, we used ICAP data for USD/JPY and EUR/USD, and Forexite data for GBP/USD, because the number of deals in the ICAP GBP/USD data was lower than we needed and the correlations between the ICAP data and Forexite data for GBP/USD were very high. The calendar dates (daily, hourly, and minute frequencies) of the ICAP data and Forexite data were consistent. We also checked the correlations between the ICAP data and Forexite data for USD/JPY and EUR/USD. Moreover, we compared the results of our experiments using both datasets and we concluded that we would be able to obtain equivalent results from the ICAP data and the Forexite data.

We constructed 1-hour and 30-min data from the ICAP data for USD/JPY and EUR/USD from 2008 to 2011, and we transformed the Forexite 1-min data for the five main currency pairs (USD/JPY, GBP/USD, EUR/USD, AUD/USD, and USD/CHF) into three different sets of time horizon data for other years. We also calculated the indicators at each time point for each time horizon.

Table 3-3. Training and testing periods

Period		Process	Length of period
A		MK regression training	1000 trading hours (around 8 weeks)
B		MK regression testing (prediction)	500 trading hours (around 4 weeks)
	B-1	MKL-GA training	250 trading hours (around 2 weeks)
	B-2	MKL-GA testing (trading)	250 trading hours (around 2 weeks)

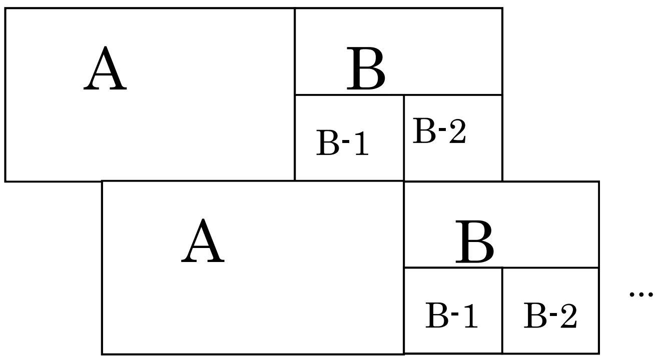

Fig.3-3. Rolling windows used for training and testing (prediction).

To separate the training and testing data, we used a rolling window method. We performed MK regression with 1000 trading hours (around 8 weeks) of data and we obtained the predicted values for the subsequent 500 trading hours (around 4 weeks). Based on the values predicted by the multiple kernel regression, the first 250 trading hours (around 2 weeks) of predictions were used as GA training and the remaining 250 trading hours (around 2 weeks) of predictions were used for GA testing (**Fig.3-3 and Table 3-3**), i.e., for testing the overall MKL-GA procedure. For each successive experiment, we moved the training and testing periods forward by 250 trading hours (around 2 weeks), which was the same as the MKL-GA trading period and this made the trading period continuous. **Table 3-4** shows the combined periods for the training and testing datasets.

Table 3-4. Datasets used in the experiments

Dataset	*Training periods*	*Testing (prediction and trading) periods*
Dataset 1	2008/01 – 2008/11	2008/03– 2008/12
Dataset 2	2009/01 – 2009/11	2009/03 – 2009/12
Dataset 3	2010/01 – 2010/11	2010/03 – 2010/12
Dataset 4	2011/01 – 2011/11	2011/03 – 2011/12

In addition, the spread (i.e., the bid/asked difference) was fixed in the experiments at 0.03 JPY per USD trading. According to EBS data for the years from 2008 to 2011, the average spreads of USD/JPY were 0.0235, 0.0246, 0.0174, and 0.0117 JPY per USD, respectively. These averages were calculated as the duration-weighted averages of the differences between the best offers and the best asking prices at the end of every second reported by ICAP. In FX trading, the spread is usually the transaction cost, which was considered in the trading simulations in the present study.

In this experiment, we treated the swap point as negligible because the swap points of our target USD/JPY were quite small. Theoretically, the swap point is the difference between the bank rates of two currencies. For JPY, the bank rate is the uncollateralized overnight call rate, whereas that for USD is the federal funds rate. The JPY rate was 0.51% at its highest in March 2008 but less than 0.1% from 2009 to 2011. The USD rate was 3.94% in January 2008, 0.16% in December 2008, and 0.21% to 0.07% from 2009 to 2011 **(Online Source 4)**. Thus, the difference was very small, except during 2008. The long positions and short positions tended to have almost equal lengths, so the profits and losses from interest that we received and paid were almost balanced. It means that the swap points were almost negligible overall.

We tested if the foreign currency exchange rate time series that we target is well approximated by linear models. The BDS test was proposed by Brock, **Dechert and Schienkman (1996)** and is now widely known for its power to test against a wide range of nonlinear models (**Barnett et al., 1997**). We apply the BDS test to the residuals from the ARIMA-type model fitted to the foreign currency exchange rate series, a possibly nonlinear time series. Theoretically we need to filter out all possible linearity which is difficult at best, but filtering by ARIMA models is accepted for the linear whitening **(Barnett, et al. 1997).**

In the tests we used the embedding dimensions from 2 to 5, and ε (the distance threshold) from 0.5, 1, 1.5 and 2 standard deviations of the data set, which are widely used and are default values for BDS test in the "tseries" package in R which we used for the analysis in the current study. The degrees of the ARIMA model fitted are from (1, 0, 0) to (7, 1, 7). As is explained below, we ran BDS tests for each testing period with varying degrees and obtained 16 p-values, each corresponding to a combination of embedding dimension and an epsilon.

When investigated yearly, the p-values in the test results are very small, i.e., even the largest one was as large as 2.42×10^{-20}. The null hypothesis that the residuals are i.i.d. is rejected. When tested every 1500 hours, which is the length of the period when our proposed method learns and predicts, majority of the periods exhibit small p-values, i.e., in 64 periods among the total of 76 periods, all the p-values are smaller than 0.01 and in remaining 12 periods, although some p-values are greater than 0.01, p-values in at most four cases (among 16 cases of combination of four kinds of embedding dimensions and four kinds of epsilons) are greater than 0.01. The 12 cases distribute as one in 2008, eight in 2009, three in 2010, and none in 2011. We therefore concluded that in hourly time series of USD/JPY exchange rate from 2008 to 2011, there might be nonlinearity dependence.

3.4.2 Initial capital and trading amount

We set the trading amount as the base amount multiplied by the leverage. We set the maximum leverage as high as 32, so the ratio of the base amount relative to the initial capital was set to 1/20 for safety.

3.4.3 Proposed method and baseline methods

The proposed and baseline methods are shown in **Table 3-5**, while **Table 3-6** provides a summary of the combinations of trading strategies and predictions.

Table 3-5. List of the methods used in the experiments

No	*Method*	*Description*
1	SVR-1-STS	SVR and a simple trading strategy. One currency pair is used.
2	SVR-3-STS	SVR and a simple trading strategy. Three currency pairs are used
3	SVR-5-STS	SVR and a simple trading strategy. Five currency pairs are used
4	SVR-1-GA	Hybrid of SVR and GA. One currency pair is used.
5	SVR-3-GA	Hybrid of SVR and GA. Three currency pairs are used.
6	SVR-5-GA	Hybrid of SVR and GA. Five currency pairs are used.
7	MKL-3-STS	MK regression and a simple trading strategy. Three currency pairs are used.
8	MKL-5-STS	MK regression and a simple trading strategy. Five currency pairs are used.
9	MKL-3-GA	Hybrid of MK regression and GA (proposed method). Three currency pairs are used.
10	MKL-5-GA	Hybrid of MK regression and GA. Five currency pairs are used.
11	Buy and Hold	Buy a currency pair and hold until the last time
12	Sell and Hold	Sell a currency pair and hold until the last time
13	ANN	ANN and a simple trading strategy
14	ARIMA	ARIMA and a simple trading strategy

Table 3-6. Summary of the combinations of trading strategies and predictions (method numbers are shown)

		Trading Strategy		
		simple	Trained by GA with SVR prediction	Trained by GA with MKL prediction
prediction	SVR	1, 2, 3	4, 5, 6	
	MKR	7, 8		9, 10

We also considered a simple trading strategy (STS) in our baseline methods to compare with the trading strategy generated by our proposed method. In this simple trading strategy, we opened a position based solely on the prediction of the result from the SVR or the MK regression. We bought the base currency of the target currency pair if the prediction was going up and sold if the prediction was going down. We closed the position after a specified period, i.e., one hour in our experiments.

In our experiment, the MKL-GA with features extracted from three main currency pairs (USD/JPY, EUR/USD, and GBP/USD) was our proposed method (Method 9), while our proposed method with features extracted from five currency pairs (Method 10) was also used in the experiments for comparison. In addition, we compared the results of a method trained in a similar way to our proposed method but trading was conducted using FX rates predicted by SVR instead of MK regression. For the simple trading strategy with SVR, we used SVR to predict the change in the rate at the succeeding time (Methods 1 to 3). In the simple trading strategy with SVR, if the predicted rate change was greater than zero, we opened a long position for our target currency pair (USD/JPY), whereas if the predicted rate change was less than zero, we opened a short position for our target currency pair.

Methods 4 to 6 were used to determine whether MK regression was necessary for our proposed framework. The SVR was trained and its predictions were used to train the GA. Methods 7 and 8 were used to determine whether a trading strategy trained using the GA was superior to a simple trading strategy if the MK regression predictions were used.

Methods 11 and 12 used the simplest trading strategy: simply buy or sell the target currency pair (USD/JPY) and wait until the end time. The holding time was 250 trading hours (around 2 weeks), like that shown in Part D of Table 5. Methods 13 and 14 used the simple trading strategy to trade based on the predictions of an ANN method and an ARIMA method, respectively. ARMA and ARMA-GARCH models were chosen so that we could compare the results of proposed method with that of conventional linear and nonlinear models. For ANN, we tried varying number of units in the hidden layer and we show the results with a highest average profit.

3.5 Experimental results

First, we evaluated the results of the MK regression part of our proposed method. We used RMSE as the evaluation measure and we compare the results of Simple Random Walk (SRW), MKL-3, MKL-5, SVR-1, SVR-3, and SVR-5. In addition, in order to compare the results of proposed method with those of some conventional models introduced in background section, we did experiments by method ARMA and ARMA-GARCH. For ARMA model, we used AICc for finding optimal parameters *p* and *q* in each training period and used them for prediction. For ARMA-GARCH model, we tried several combinations of parameters as ARMA, and we show the results of ARMA(1,1) with GARCH(1,1) since there are very small differences in the results.

3.5.1 RMSE results

Table 3-7. Average RMSE in percentage for each method in each year

Year	*ARMA*	*ARMA-GARCH*	*SRW*	*SVR-1*	*SVR-3*	*SVR-5*	*MKL-3*	*MKL-5*
2008	0.28551	0.212021	0.207860	0.297933	0.822765	0.699106	0.216617	0.206236
2009	0.20516	0.155264	0.150891	0.193786	0.648358	0.775471	0.157167	0.154017
2010	0.16548	0.126683	0.124943	0.172585	0.619003	0.664844	0.131836	0.126947
2011	0.12548	0.108685	0.101184	0.123165	0.420466	0.432392	0.108152	0.102514

Table 3-8. Standard deviation of the RMSE in percentage for each method in each year

Year	*ARMA*	*ARMA-GARCH*	*SRW*	*SVR-1*	*SVR-3*	*SVR-5*	*MKL-3*	*MKL-5*
2008	0.11149	0.07575	0.076772	0.196957	0.432763	0.215905	0.071405	0.076052
2009	0.03618	0.02295	0.023337	0.082710	0.168785	0.155797	0.021742	0.023879
2010	0.06101	0.03595	0.040646	0.103058	0.156235	0.174371	0.038881	0.037304
2011	0.05182	0.02947	0.032658	0.058568	0.139091	0.101686	0.031470	0.031909

Based on the average RMSE (in percentage) results for the testing periods from 2008 to 2011, which are shown in **Tables 3-7 and 3-8**, we found that the error (RMSE) with MK regression was much smaller than that with SVR. Thus, MK regression outperformed SVR, which was very important for the success of the subsequent GA learning. We also found that the RMSE results with MK regression are better than those with ARMA and ARMA-GARCH. Many empirical studies have reported that SRW is better than or competitive with other models used to predict FX rates and the average RMSE results with MKL-3 and MKL-5 were very close to that with SRW, but we could not use SRW for trading because SRW gave us no indication of the action to take.

3.5.2 Profit and loss ratios with the proposed method and benchmark methods

In one experiment set, we consider 1000 trading hours (around 8 weeks) for MKL learning and 500 trading hours (around 4 weeks) for MKL testing, which includes 250 trading hours (around 2 weeks) for GA training and 250 trading hours (around 2 weeks) for GA testing. Thus, for each year we conducted 19 experiments. We calculate the average and standard deviation of the profit of each experimental set for each method and each year. The profits were measured relative to the initial investment and were not scaled for years or weeks. The results are shown in **Tables 3-9 and 3-10**.

Table 3-9. Average profits with each method during 250 trading hours (around 2 weeks) in each year (19 experiments per year). The profits were measured relative to the initial investment.

Method	*2008*	*2009*	*2010*	*2011*
SVR-1-STS	−0.01337	−0.02367	−0.01644	−0.01702
SVR-3-STS	−0.01725	−0.00659	−0.00416	−0.00433
SVR-5-STS	0.01644	0.00397	−0.00297	0.00242
MKL-3-STS	−0.00672	0.00308	−0.00232	0.00066
MKL-5-STS	0.00992	0.00417	0.00153	0.00201
SVR-1-GA	0.00103	0.00192	0.00260	0.00389
SVR-3-GA	0.00144	0.00173	0.00111	0.00299
SVR-5-GA	0.00411	0.00248	−0.00202	0.00157
MKL-3-GA	0.00754	0.01084	0.00762	0.00660
MKL-5-GA	0.00510	0.00376	0.00952	0.00648
Buy and Hold	−0.00432	−0.00221	−0.00429	−0.00194
Sell and Hold	0.00432	0.00221	0.00429	0.00194
ANN	−0.00680	0.00042	−0.00287	−0.00477-
ARIMA	−0.00467	−0.00106	−0.00307	−0.00364

Table 3-10. Standard deviation of the profit for each method during 250 trading hours (around 2 weeks) in each year (19 experiments per year).

Method	*2008*	*2009*	*2010*	*2011*
SVR-1-STS	0.02647	0.01635	0.01865	0.01792
SVR-3-STS	0.03431	0.01738	0.01838	0.02068
SVR-5-STS	0.02934	0.02938	0.01752	0.01422
MKL-3-STS	0.04304	0.02040	0.02014	0.01728
MKL-5-STS	0.03323	0.02477	0.02375	0.01976
SVR-1-GA	0.00750	0.01191	0.01173	0.01081
SVR-3-GA	0.00911	0.00573	0.00689	0.00857
SVR-5-GA	0.00896	0.00605	0.01610	0.01023
MKL-3-GA	0.01631	0.01299	0.01502	0.01631
MKL-5-GA	0.01152	0.00908	0.01522	0.01168
Buy and Hold	0.03257	0.02416	0.01876	0.01556
Sell and Hold	0.03257	0.02416	0.01876	0.01556

ANN	0.02234	0.01200	0.00872	0.01582
ARIMA	0.04038	0.02057	0.01881	0.02150

Note that there were 19 training and testing sequences in each year. We calculated the annual returns with each method, where the annual return was the sum of profits made in 19 experiments during a specific year. The returns were measured relative to the initial investment. **Table 3-9** shows the returns obtained with different methods. First, we focus on the methods based on the simple trading strategy (SVR-3-STS, SVR-5-STS, MKL-3-STS, and MKL-5-STS).

In some years, the methods based on the simple trading strategy obtained good returns, such as SVR-5-STS and MKL-5-STS in 2008. However, the methods with a simple trading strategy did not yield good returns stably, which is demonstrated by the results in **Table 3-9**. They also incurred huge losses in some years, such as SVR-1-STS from 2008 to 2011, or SVR-3-STS in 2008 and 2009. There were large differences in the returns during each experiment and the huge overall losses in some testing years were caused by very high losses in specific experiments. For example, **Fig. 3-4** shows the distribution of the returns with SVR-3-STS during the testing periods in 2011. There were 19 testing periods in each year. Ittime frames shows that there was a large overall loss in 2010 and it was attributable mainly to the huge losses in the first and tenth trading periods of 2011, although the number of periods that yielded profits was almost the same as the number of periods that incurred losses (9 and 10, respectively).

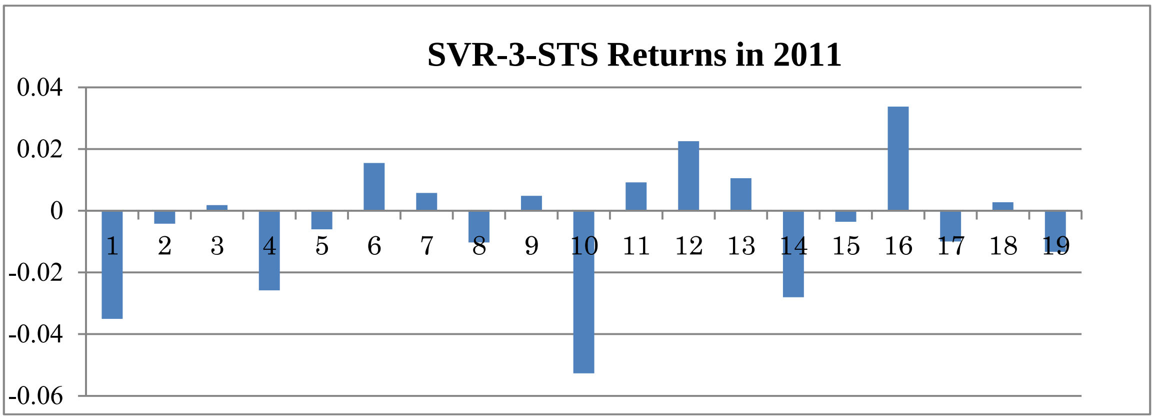

Fig.3-4. Distribution of returns in each trading period during 2010 for SVR-3-STS where the horizontal axis is the *n*-th testing period and vertical axis is the return in proportion to initial investment

Table 3-11. Profits with the proposed method and baseline methods between 2008 and 2011. The profits were measured relative to the initial investment per year

Method	*2008*	*2009*	*2010*	*2011*
SVR-1-STS	−0.2541	−0.4498	−0.3123	−0.3232
SVR-3-STS	−0.3278	−0.1251	−0.0790	−0.0821
SVR-5-STS	0.3123	0.0753	−0.0564	0.0459
MKL-3-STS	−0.1278	0.0585	−0.0441	0.0126

MKL-5-STS	0.1883	0.0791	0.0291	0.0381
SVR-1-GA	0.0195	0.0364	0.0495	0.0740
SVR-3-GA	0.0274	0.0328	0.0212	0.0568
SVR-5-GA	0.0780	0.0471	−0.0384	0.0298
MKL-3-GA	0.1433	0.2060	0.1447	0.1254
MKL-5-GA	0.0968	0.0715	0.1808	0.1231
Buy and Hold	−0.0822	−0.0419	−0.0815	−0.0368
Sell and Hold	0.0822	0.0419	0.0815	0.0368
ANN	−0.1291	0.0081	−0.0546	−0.0905
ARIMA	−0.0886	−0.0200	−0.0583	−0.0690

Based on the results with the SVR-GA hybrid method, which combined SVR with GA and was a simplified version of our proposed hybrid method, we found that the overall profits with our proposed methods were better in each year than those with SVR-GA. This may have been because of the relatively large errors (RMSE) of the SVR, as shown in **Table 3-7** and Table 3-8. We also found that ANN achieved losses in three years and a profit close to zero in 2009, while ARIMA suffered losses in four years.

The results with our proposed methods showed that MKL-3-GA and MKL-5-GA delivered returns that ranged from 7.1% to 18.08% per year and they incurred no losses for one year. A comparison of these methods over each period showed that our proposed method performed consistently better than the other methods. **Figures 3-5 to 3-8** show the accumulated profits (relative to the initial investments) in the experiments with proposed method MKL-3-GA method.

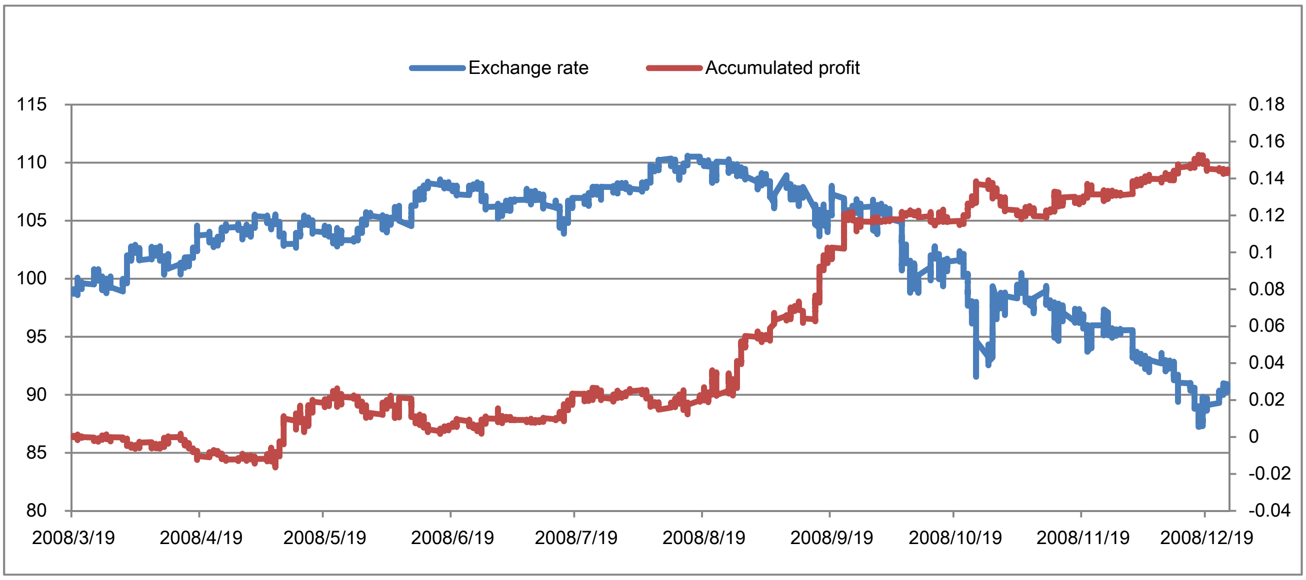

Fig.3-5. USD/JPY profit accumulated by MKL-3-GA from March 2008 to December 2008 where the horizontal axis is the testing period, the left-hand vertical axis is the value for FX rate, and the right-hand vertical axis is the return of proposed method in proportion to initial investment

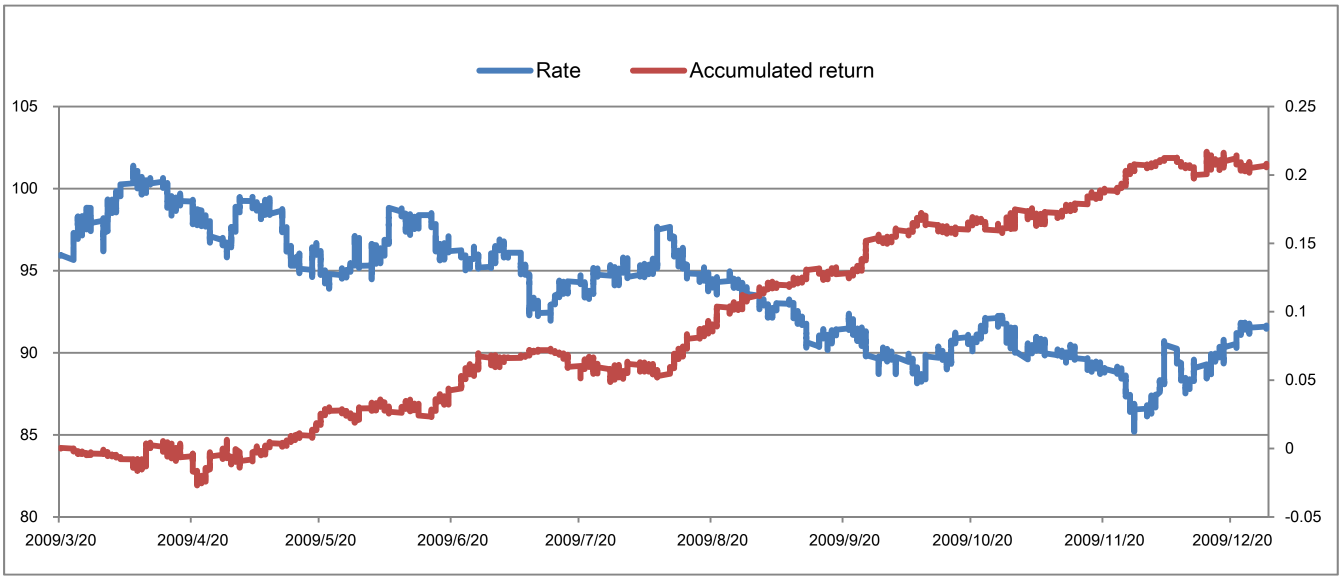

Fig.3-6. USD/JPY profit accumulated by MKL-3-GA from March 2009 to December 2009 where the horizontal axis is the testing period, the left-hand vertical axis is the value for FX rate, and the right-hand vertical axis is the return of proposed method in proportion to initial investment

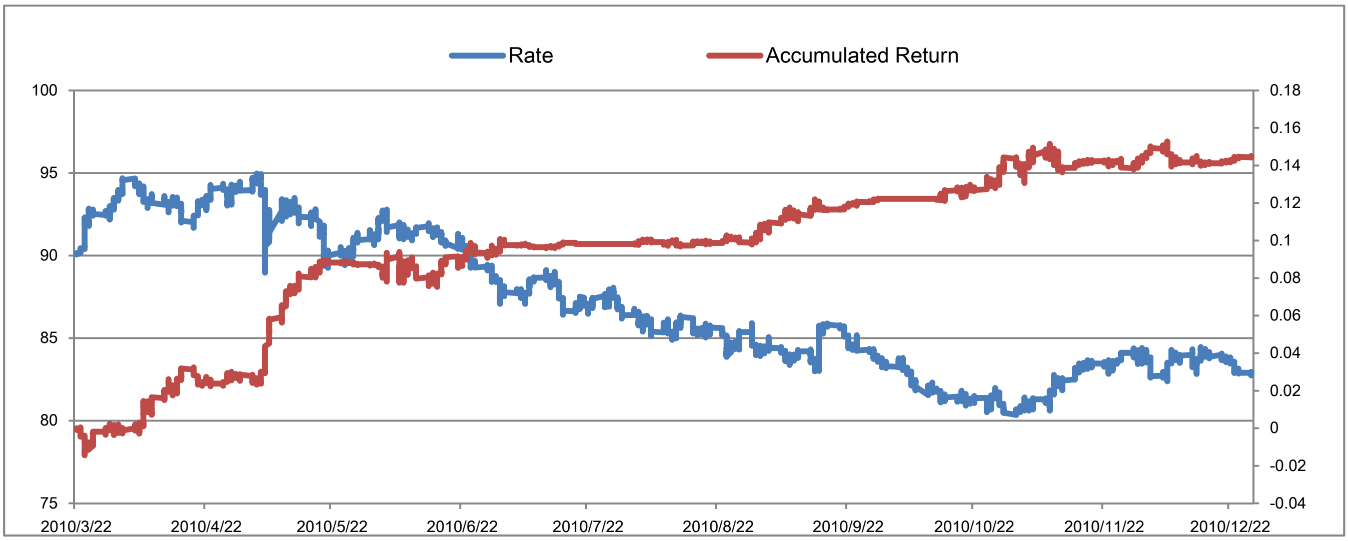

Fig.3-7. USD/JPY profit accumulated by MKL-3-GA from March 2010 to December 2010 where the horizontal axis is the testing period, the left-hand vertical axis is the value for FX rate, and the right-hand vertical axis is the return of proposed method in proportion to initial investment

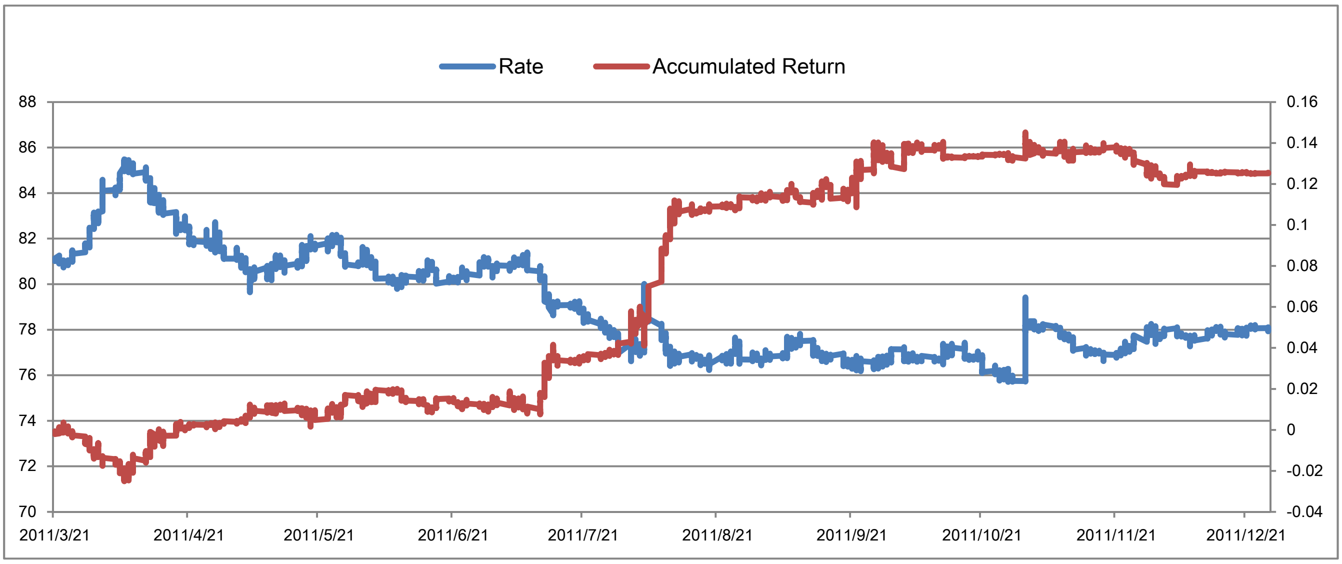

Fig.3-8. USD/JPY profit accumulated by MKL-3-GA from March 2011 to December 2011 where the horizontal axis is the testing period, the left-hand vertical axis is the value for FX rate, and the right-hand vertical axis is the return of proposed method in proportion to initial investment

Figures 3-5 to 3-8 show that how the large returns were made had interesting characteristics: in some years, most of the profits are made in the minority of the trading periods, e.g., most of the profits were made during August to September but in 2010, most of the profits were made during March to May, whereas most of the profits were made in July to August in 2011. However, in some years, profits were made in most of the trading periods. For example, profits were made in most of the year in 2009, almost from the start to the finish, and the final profit was 20%. Similar results were reported in a previous study **(Fletcher, 2010)**, where the model lacked high directional predictability but it yielded good profits because some transactions had enhanced profitability. We consider that high directional predictability or a capacity to make profits in the majority of transactions are important, but the fitness value of the GA we designed was the accumulated return, so we focused on the obtaining the best overall return in this study. Other important evaluation measures will be considered in our future research.

3.5.3 Effect of single information source on the returns of proposed method

In order to figure out which part of the multiple information sources is utilized by the proposed algorithm (MKR-3-GA) to obtain profits shown in **Table 3-11**, we use a random sequence in place of one of the multiple information sources (USD/JPY, EUR/USD or GBP/USD), for getting to know which time series data source(s) was (were) in fact in fact contributed or not to the return results.. **Table 3-12** shows a list of the benchmark methods used in the additional experiments.

Table 3-12. List of the benchmark methods used in additional experiments

No	*Method*	*Description*
1	MKR-GA-noUJ	The same as the proposed method MKR-3-GA except that the USD/JPY time series data is replaced by a random sequence
2	MKR-GA-noEU	The same as the proposed method MKR-3-GA, except that the EUR/USD time series data is replaced by a random sequence
3	MKR-GA-noGU	The same as the proposed method MKR-3-GA, except that the GBP/USD time series data is replaced by a random sequence

The returns of three additional benchmark methods and that of the proposed method are shown in **Table 3-13**. For the year 2008, MKR-GA-noUJ, MKR-GA-noEU, and MKR-GA-noGU yielded returns 0.12904, 0.04591 and 0.12454, respectively, compared with 0.14338 yielded by the proposed method MKR-3-GA. It indicates that in the year 2008 the information sources USD/JPY and GBP/USD contributed about 10%, while the information source EUR/USD contributed about 68% of the returns by the proposed method. For the year 2009, MKR-GA-noUJ, MKR-GA-noEU and MKR-GA-noGU yielded returns 0.08041, 0.15605 and 0.11557, respectively, all of which are smaller than that of the proposed method MKR-3-GA (0.20603). It indicates that all of the three FX information sources (USD/JPY, EUR/USD and GBP/USD) contributed more than 24% of the returns by the proposed method. For the year 2010, MKR-GA-noEU and MKR-GA-noGU yielded returns 0.1494 and 0.1487, which are very close to the return of the proposed method MKR-3-GA (0.1447), while MKR-GA-noUJ yielded returns about only 0.06257. It indicates that for the returns of the proposed method in year 2010, the information source USD/JPY contributed more than 50% of the profit by the proposed method while EUR/USD and GBP/USD contributed none of the returns of the proposed method. For the year 2011, return of MKR-GA-noUJ (0.12014) is very close to that of the proposed method MKR-3-GA (0.12541), while MKR-GA-noEU and MKR-GA-noGU yielded only 0.05807 and 0.09225, respectively. It indicates that in the year 2011 the information EUR/USD and GBP/USD contributed more than 25% of the returns by the proposed method but USD/JPY contributed less than 4% of the return.

Table 3-13. Returns of the proposed method and additional benchmark methods from 2008 to 2011. The returns were measured relative to the initial investment per year

Model	*2008*	*2009*	*2010*	*2011*
MKR-3-GA (proposed)	0.14338	0.20603	0.14478	0.12541
MKR-GA-noUJ	0.12904	0.08041	0.06257	0.12014
MKR-GA-noEU	0.04591	0.15605	0.14942	0.05807
MKR-GA-noGU	0.12454	0.11557	0.14872	0.09225

3.5.4 Sharpe ratios values with the proposed and benchmark methods

We evaluated the Sharpe ratio for our proposed method and the other baseline methods over the testing periods. The

3-month Japan Government Bond (http://www.mof.go.jp/english/jgbs/topics/t_bill/index.htm) return was used as risk-free return. We used “Yield at the Average Price” to calculate the average of 3-month risk-free return of each year, which was 0.49% (in average) in 2008, 0.16% (in average) in 2009, 0.12% (in average) in 2010, and 0.10% (in average) in 2011. We calculated the average returns of 2008 to 2011 and the average risk-free return of each testing year (converted to ten months return in average) was 0.725%. **Table 3-14** shows the average returns, standard deviations, and Sharpe ratios for each method.

Table 3-14. Sharpe ratios with the proposed method and the baseline methods

Model	*Average Return per Year (2008-2011)*	*Std. Dev.*	*Sharpe Ratio*
SVR-1-STS	－0.33488	0.08239	-4.15257
SVR-3-STS	－0.15354	0.11809	-1.36159
SVR-5-STS	－0.09430	0.15599	-0.651
MKL-3-STS	－0.02518	0.08031	-0.40381
MKL-5-STS	0.08373	0.07309	1.046381
SVR-1-GA	0.04490	0.02297	1.639094
SVR-3-GA	0.03462	0.01558	1.756739
SVR-5-GA	0.01033	0.00422	0.729858
MKL-3-GA	**0.15490**	0.03521	**4.193411**
MKL-5-GA	**0.11808**	0.04684	**2.36614**
Buy and Hold	－0.06064	0.02462	-2.75751
Sell and Hold	0.06064	0.02462	2.168562
ANN	－0.06665	0.05833	-1.26693
ARIMA	－0.05903	0.02882	-2.29979

A higher Sharpe ratio indicates a higher return or lower volatility. In some testing periods, several baseline methods obtained higher returns than our proposed method, but in the longer term our proposed method had a significantly higher Sharpe ratio than the baseline methods throughout the testing periods. We might expect improvements of the returns and the Sharpe ratios with the other methods if we introduced the additional tools used in our proposed methods. For example, we did not implement risk management in the simple trading strategy (STS) (i.e., we did not use stop-loss and take-profit orders). Indeed, the huge losses with SVR-1-STS in 2008, SVR-3-STS in 2009, and MKL-3-STS in 2010 could have been reduced if we had implemented risk management. However, we could not estimate the appropriate levels for the take-profit and stop-loss points in percentage terms without learning.

3.5.5 MKL coefficients results

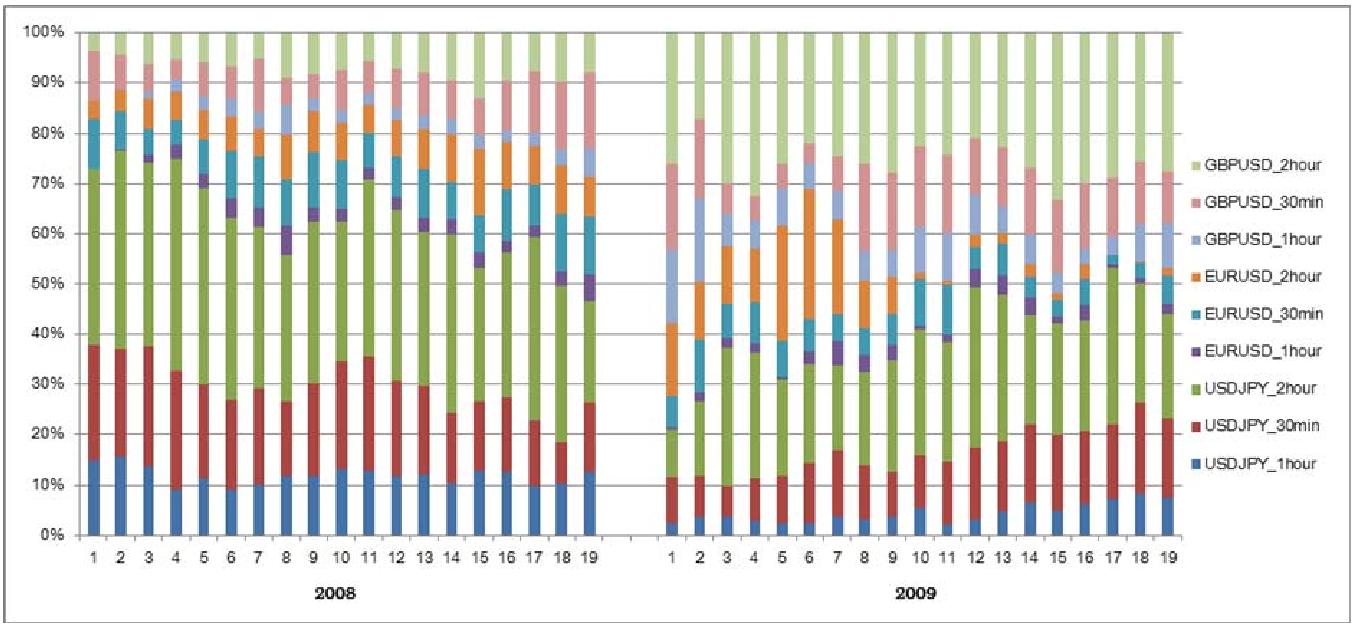

Fig.3-9. Coefficients of MK regression (three currency pairs) for 2008 and 2009

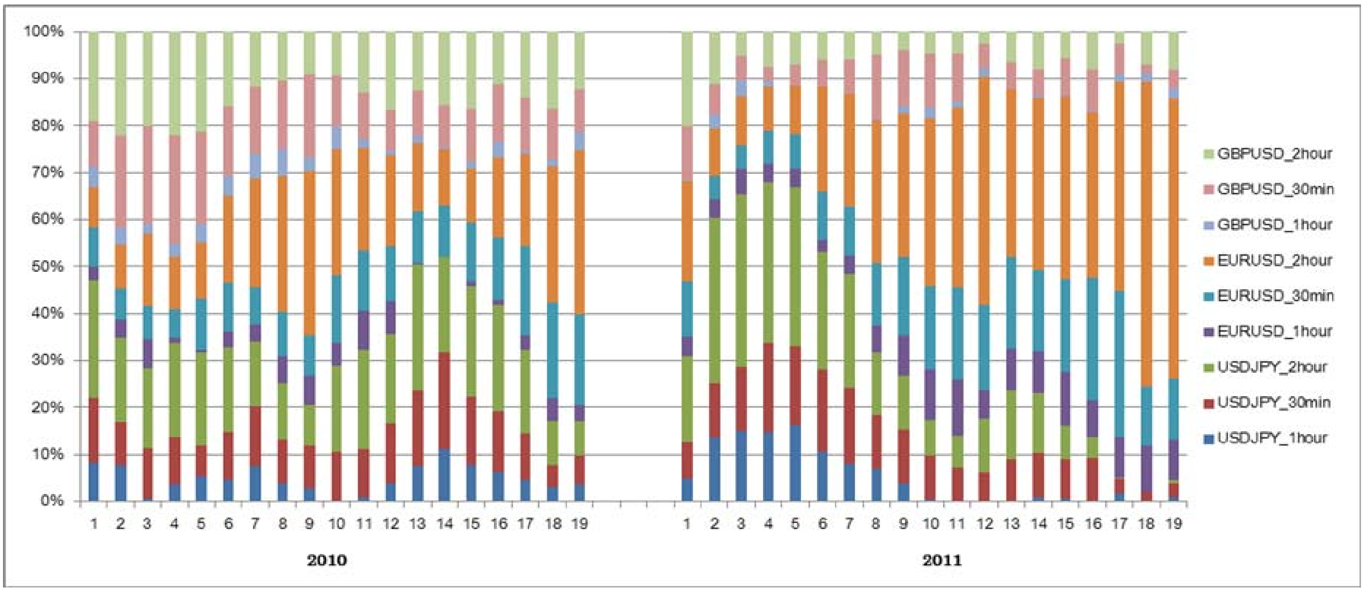

Fig.3-10. Coefficients of MK regression (three currency pairs) for 2010 and 2011

The *x*-axis values (1–19) indicate the indexes of the testing periods, which were numbered as our rolling windows proceeded. We had 19 learning and testing pairs in each year. The *y*-axis represents the relative weights of the currency pairs and the time horizons.

Figures 3-9 and 3-10 show the weights for three currency pairs with three time horizons during the training periods of

2008 to 2011. These figures show that the relative weights of the 2-hour USD/JPY were higher than the others in most of the MKL training periods. Thus, the 2-hour USD/JPY could be important references for identifying the trends in our target USD/JPY with a 1-hour time horizon. In addition, the weight of the 30-min USD/JPY was more stable than the others in the 2008–2011 training periods.

However, some currency pairs with specific time horizons had weights that were consistently smaller than others during the training periods of 2008 to 2011. For example, the weight of the 1-hour EUR/USD ranged from 0% to 2.5% from 2008 to 2010. **Table 3-13** provides a summary of the properties of the weights for specific currency pairs and their time horizons.

Table 3-13. Summary of the properties of weightings for specific currency pairs and time horizons (three currency pairs)

Year	*Consistently Small Weight*	*Consistently Large Weight*	*Weight Changes Rapidly*
2008	1-hour GBP/USD, 1-hour EUR/USD	2-hour USD/JPY	
2009	1-hour EUR/USD	2-hour GBP/USD	2-hour EUR/USD
2010	1-hour GBP/USD, 1-hour EUR/USD	2-hour USD/JPY	
2011	1-hour GBP/USD	2-hour EUR/USD	2-hour USD/JPY

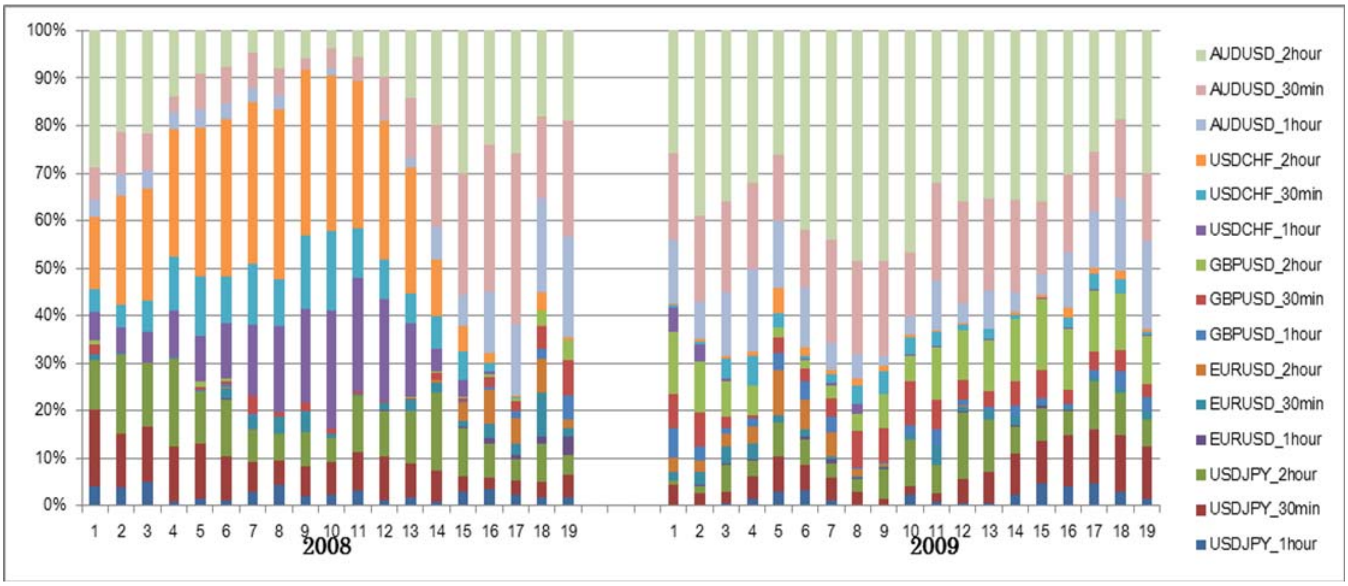

Fig.3-11. Coefficients of MK regression (five currency pairs) for 2008 and 2009

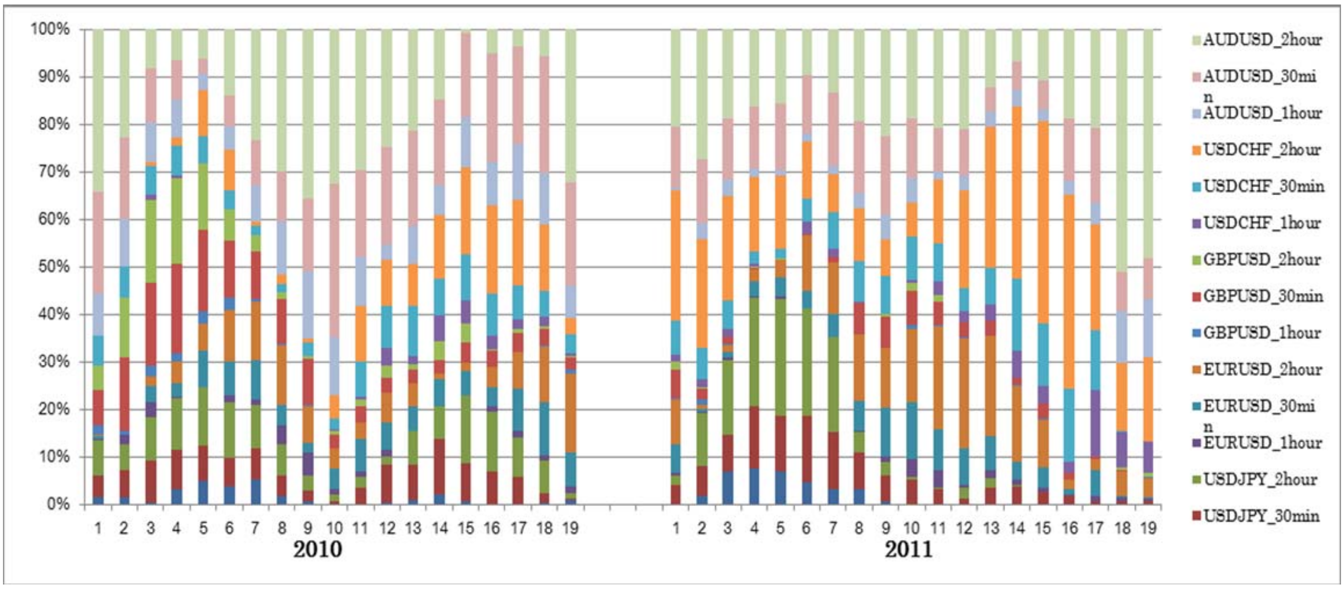

Fig.3-12. Coefficients of MK regression (five currency pairs) for 2010 and 2011

Figures 3-11 and 3-12 show the weights for five currency pairs with three time horizons in the training periods of 2008 to 2011. These weighting results show that the 2-hour USD/CHF and 2-hour EUR/USD changed greatly over time. For example, the 2-hour USD/CHF had a relatively high weight in 2011 but a very low weight in 2009. In addition, the relative weight of the 2-hour AUD/USD was larger than the others in most of the MKL training periods from 2008 to 2011. Thus, the 2-hour AUD/USD could be an important reference for identifying the trend in our target USD/JPY with a 1-hour time horizon.

However, some currency pairs with specific time horizons had consistently lower weights than others during the training periods of 2008 to 2011. For example, the weight of the 1-hour EUR/USD ranged from 0% to 2%. **Table 3-14** provides a summary of the properties of the weights of specific currency pairs with their time horizons.

Table 3-14. Summary of the properties of weightings for specific currency pairs and time horizons (five currency pairs)

Year	*Consistently Small Weight*	*Consistently Large Weight*	*Weight Changes Rapidly*
2008	30-min, 1-hour, 2-hour GBP/USD; 30-min, 1-hour, 2-hour EUR/USD; 1-hour USD/JPY		1-hour AUD/USD; 2-hour USD/JPY
2009	30-min, 1-hour, 2-hour EUR/USD; 30-min, 1-hour, 2-hour USD/CHF; 1-hour GBP/USD	30-min, 2-hour AUD/USD	
2010	1-hour USD/CHF; 1-hour GBP/USD; 1-hour EUR/USD		2-hour AUD/USD; 2-hour USD/CHF
2011	30-min, 1-hour, 2-hour GBP/USD;	2-hour USD/CHF	30-min, 1-hour USD/CHF;

	1-hour EUR/USD	2-hour AUD/USD; 2-hour EUR/USD

Initially, we expected that GBP/USD and EUR/USD would be more important than USD/CHF and AUD/USD for our target USD/JPY prediction. However, we found that the weights of three time horizons were consistently smaller for EUR/USD and GBP/USD over the years than for USD/CHF and AUD/USD, which contradicted our expectation. However, we also found that the weights of AUD/USD and USD/CHF usually changed rapidly, which may suggest that these are important for our predictions, although the importance might not be stable over different periods.

3.5.6 MKL coefficients results when using a random sequence

In **Section 3.5.3**, we used a random sequence as a source among the multiple information sources (USD/JPY, EUR/USD or GBP/USD), for getting to know which time series data source(s) in fact contributed or not to the return results. From these three additional experiments, we obtained the MKL coefficients for the three information sources (two FX information sources and one random sequence).

Table 3-15 to **Table 3-17** show the average of coefficients (each coefficient is the sum of the coefficients of the kernels for all three time frames) for the three additional experiments. We obtained the coefficients of the first additional experiment (which used a random sequence in place of USD/JPY) and the average coefficient of each information source as shown in Table 3-15. The coefficients from the year 2008 to 2011 (in average of 19 periods in each year) of a random sequence used in placed of USD/JPY contribute only about 5% to 8% to the total of the coefficients, which is much smaller than that of EUR/USD (about 38% to 58%) and GBP/USD (about 34% to 57%). From results of the second additional experiment (which used a random sequence in place of EUR/USD) and the third additional experiment (which used a random sequence in place of GBP/USD) shown in **Tables 3-16** and **3-17,** we got similar conclusion that MKL distributed relatively small coefficient (under 10%, in average) to the random sequence, with comparing to the coefficients of the real foreign exchange rate sources. They indicate that MKL reduces the coefficients of the kernel for a random sequence, because MKL found an information source that has virtually no related information.

Table 3-15. Average Coefficients (of 19 training periods) of MK regression when USD/JPY is replaced by a random sequence

Year	Random Sequence	EUR/USD	GBP/USD
2008	0.0803	0.4379	0.4816
2009	0.0431	0.3848	0.5720
2010	0.0552	0.5092	0.4355
2011	0.0643	0.5887	0.3469

Table 3-16. Average Coefficients (of 19 training periods) of MK regression when EUR/USD is replaced by a random sequence

Year	Random Sequence	USD/JPY	GBP/USD
2008	0.0376	0.4610	0.5012
2009	0.0163	0.4094	0.5741
2010	0.0525	0.4142	0.5332
2011	0.0722	0.4088	0.5189

Table 3-17. Average Coefficients (of 19 training periods) of MK regression when GBP/USD is replaced by a random sequence

Year	Random Sequence	USD/JPY	EUR/USD
2008	0.0419	0.4869	0.4711
2009	0.0296	0.5088	0.4614
2010	0.0454	0.4063	0.5481
2011	0.0666	0.3125	0.6208

3.6 Conclusion

In this section, we developed a hybrid method, which combined MK regression with a GA, where MK regression was used to build a prediction model and the GA was used to formulate a trading strategy. The MACDs of the main (three or five) currency pairs and three different short time horizons were used to make predictions. We used one Gaussian kernel and one linear kernel for each currency pair, and we used the MK method to combine the features of different currency pairs.

First, we applied MK regression to FX data for a specific training period to estimate the optimal parameters and weights. The results showed that the FX rate changes predicted by MK regression were much better than those predicted by SVR during our testing period, in terms of the RMSEs. Next, we used the GA to optimize the trading strategy using the predicted FX rate changes and the overbought/oversold indicators for the training period.

We traded USD/JPY based on the trading strategies generated using the GA, and calculating the returns and Sharpe ratios for the testing periods from March to December between 2008 and 2011. Each testing period was 250 hours (around 2 weeks). In some testing periods, several baseline methods outperformed our proposed methods in terms of profit, but our proposed methods obtained consistently good profits and Sharpe ratios without experiencing losses in any year. The profits ranged from 7.1% to 20.60% per year from 2008 to 2011, and the Sharpe ratios for MKL-3-GA and MKL-5-GA were 4.39 and 2.52, respectively. The average profit obtained during the testing period was positive with a statistically high confidence (higher than 99% confidence). In short, our proposed method obtained consistently favorable returns with low volatility over a four-year period.

The weights obtained using multiple kernel learning, which were applied to the regression function to determine the FX rate changes with different currency pairs and different time horizons, showed that there was a possible correlation between our target pair (USD/JPY), its target trading time interval (one hour), and other currency pairs, or other time horizons. The relative weights of the kernels calculated from the results of multiple kernel learning could be utilized by traders to identify possible correlations between reference currency pairs with reference time horizons and the target trading currency pair with the target time horizon.

The problems that remain include the determination of the best time horizon lengths for use as the references in our target trading time horizon. The selection of a very long or very short time horizon may also have negative effect on the predictions.

We only determined the relative weights among currency pairs with certain time horizons, because we did not consider all of the possible time horizons for all of the different pairs during MKL training. In addition, changing the input features might produce different weights for different currency pairs and different time horizons (we used the MACD indicator as the feature for MK regression). Changing the input features (using other indicators or other transforms from raw data) will be a future direction in our research.

4. Stock price prediction by MKL-GA method

4.1 Introduction

Many business practitioners and researchers have developed various kinds of models to predict and analyze stock prices. For example, there are many studies that estimate and predict the stock prices and stock volatility by using historical stock prices or volumes data. Researchers such as **Murphy (1999) and Neely (1998)** showed that technical analysis is one possible way to predict the stock markets and foreign exchange markets successfully. **Deng and Sakurai (2013)** used a single technical indicator from multiple time frames to generate trading rules.

In the last decade, numerous researchers have used machine learning technologies, such as artificial neural network (ANN), support vector machine (SVM), genetic algorithm (GA) or some integrated models of these to predict stock price or exchange rate changes or to find trading rules for the stock or foreign exchange rate trading, by mining historical price or transaction volume data. For example, Hann and Steurer (1996) and **Chen and Leung (2004)** applied ANNs to foreign exchange rate prediction and found that the ANN-based model outperformed the linear models. **Hadavandi et al. (2010)** proposed an ANNs-based integrated system for stock price forecasting. **Kwok (2000) and Cao (2003)** applied an SVM to predict the stock price and obtained good results. **Shioda et al. (2011)** predicted the foreign exchange market states with SVM. **Fuente et al. (2006)** and **Allen and Karjalainen (1999)** applied GA to generate trading rules. **Chien and Chen (2010)** applied a GA-based model to mine associative classification rules with stock trading data. **Hirabayashi et al. (2009)** applied GA to finding trading rules for foreign exchange intraday trading by mining features from several technical indicators. **Deng et al. (2013)** forecasted short term foreign exchange rates by a GA-based hybrid model. Their proposed models obtained better performance than that of some conventional models; however, they utilized the features extracted from only the historical prices or transaction volumes.

Other than stock time series data such as stock prices and transaction volumes, human factors have been considered recently as having significant impacts on the movements of stock prices. We hypothesize that these impacts could be quantified by looking at the Internet, since the advent of the digital information age has led many organizations and people to post news and their comments on the news on well-known social networks such as Twitter, Facebook, or Engadget. Therefore, by analyzing the dynamics of news items or user comments about relevant companies on the Internet, we may mine interesting possible correlations between social network activities with stock price movements. Previous researchers, such as **Gruhl et al. (2005)**, found correlations between sales rank and blog mentions. **Mondria et al. (2010)** used internet to search query data. **Smith (2012)** used Google internet search activity to predict volatility in the foreign currency. Choudhury et al. (2007) and Li et al. (2007) modeled dynamics in social networks. **Choudhury et al. (2008)** identified several contextual properties of communication and described dynamics in user comments and used an SVM framework to learn and to predict stock prices, while **Bollen et al. (2011)** used Twitter tweets to gauge the mood of stock markets and predict the stock market. Good performances on out-of-sample data of certain stocks showed that mining information from the Internet could be an alternative or complementary approach to prediction of changes in stock prices.

In this research, since the stock price movements are accumulation of individual behaviors which may appear as activities in social network service (SNS), we model stock price change rates as a function of quantitative features of news and comments in SNS and features relating to technical indicators of stock prices and trading volumes. To incorporate different types of features such as those for news and for prices into a regression model, a method called multiple kernel learning (MKL) by **Bach (2004)** is a very promising way. A strong point of MKL is that it allows us to combine different kernels when the job at hand requires one to use different kernels for different input features. In addition, MKL mitigates the risk of erroneous kernel selection to some degree, by taking a set of kernels and deriving a weight for each kernel, such that predictions are made based on the weighted sum of the kernels. Some researchers have applied MKL to predict foreign exchange rate or stock prices. For example, **Fletcher et al. (2010)** applied MKL to the limit order book for predicting the movement and trading of the EUR/USD currency pair. **Luss and d'Aspremont (2012)** applied MKL towards the prediction of abnormal returns from historical stock prices data and news. **Lee et al. (2010)** applied MKL to the prediction of prices in Taiwan's stock market, obtaining results that surpassed those of some conventional methods. Good performances in these literatures inspired us to use MKL to utilize the information from different sources.

Evaluation measures are very important to evaluate the performances of models. The root mean square error (RMSE) is a measure which is often used for evaluating prediction results. However, given that people will sell or buy stocks when they can predict the stock price, the goodness of the predictions cannot be provided by differences of prediction and real values alone; a proper measure should be the trading profits based on the prediction. In addition, beyond the accumulated returns, most investors also pay close attention to the variability of returns. In other words, they hope the proposed model can increase profits as well as decrease associated risks while doing so. Therefore, to evaluate the appropriateness of prediction, we should not confine ourselves to RMSE and we should also use accumulated returns and returns to variability ratio or Sharpe ratio that further considers risk free profits.

To evaluate appropriateness of predictions by the accumulated returns and returns to variability ratio (or Sharpe ratio), simulated trading should be conducted based on the prediction, which requires us to define a trading rule. The trading rule is a set of rules that specify under what condition, an action "sell", "buy", or "no trade" should be taken. Since the conditions include some parameters which should be set according current market situations, they should be learned from data. We did not consider transaction costs in this paper since the purpose of this research is to evaluate the prediction performance of the proposed model, not trading. There would be a case that a model could attain higher than 50% hit ratio of predicting directions, however, it is possible that this model's prediction is correct for smaller magnitude of movements but incorrect for larger magnitude of movements, which is not favorable for application of the model to real financial markets. We want to show that our proposed model yields positive returns and therefore the case does not happen in our proposed model.

For learning the trading rules, we adopted GA with the resulting accumulated profits as the fitness value of chromosomes, because the profit is generated through transactions that are discrete events. As demonstrated in **Allen and Karjalainen (1999)**, GA is a good method for finding good trading rules.

In summary, our study makes three main contributions. First, we extract features from both the time series data source and a social network source, in contrast to previous studies **(e.g., Choudhury et al. (2008))** which considered properties of social network to predict stock price movements, and Luss and d'Aspremont (2012) used text data of news for the prediction of abnormal returns. These literatures inspired us to attempt the prediction of stock prices by extracting features from time series data and a social network.

Second, we use the multiple kernel regression (MKR) framework to optimally combine the features of time series data, news, and user comments, in contrast to other works (e.g., **Choudhury et al. (2008))** which used a single kernel for the SVM. Results from **Hann and Steurer (1996), Chen and Leung (2004), Kwok (2000), and Cao (2002)** prove that ANNs and SVMs (especially the latter) are good models to predict stock price change rates. However, given that the input features extracted from the time series of historical stock price change rates and those from a social network have different properties, we should consider using different kernels for input feature sets from different types of sources. However, it is not easy to assign good kernels manually. Therefore, we use MKL to solve this problem.

Third, for generating trading signals, we use not only the predicted stock price change rates from the MKL model, but also three well-known overbought and oversold technical indicators. In addition, we consider thresholds over which the difference between the predicted value and current value should prompt an action: we buy if the combined decision value is greater than the buying threshold, and we sell if the combined decision value is less than the selling threshold. The best values of both thresholds are learned in the GA learning periods.

4.2 Features used in proposed method

Features used for proposed model are from three sources: historical traded prices and volumes of stocks, news in social network service, and user comments on the news.

4.2.1 Features from historical prices and volumes

The ROC, SMA and MACD are often used to understand the present trend by traders. Therefore, we use these technical indicators features for historical prices and transaction volumes, which are shown in **Table 4-1**.

Table 4-1. List of features from historical traded prices and volumes

Number	***Features based on historical prices and volumes***
1	ROC for historical prices
2	ROC for historical transaction volumes
3	SMA for historical prices
4	SMA for historical transaction volumes
5	MACD for historical prices
6	MACD signal for historical prices

4.2.2 Features from news and user comments

Table 4-2 shows a list of numerical features from news items. The greater the number of news items about a certain company, the more the impact of the event described by that news item on a future event. Hence, the first numerical feature is the number of news per day f_t^c, where *t* is the date, and *c* is the stock described in the news. In addition, the SMA is used to understand the present trend, thus SMA of frequency of news (FNSMA) is used as feature. We set the parameter *n* for SMA to be 7 because we assume that the return of a company stock on a certain weekday depend on its stock prices and communication activity pertaining to it in the preceding 7 days.

Table 4-2. List of numerical features of news items

Number	*Numerical features based on news*	*Calculation*
1	Frequency of news	f_t^c
2	SMA of frequency of news	$FNSMA(t) = (\sum_{k=t-n+1}^{t} f_k^c)/n$ n=7

Table 4-3 shows a list of numerical features from user comments which are defined similarly to news

Table 4-3. List of features based on user comments

Number	*Features based on user comments*	*Calculation*
1	Frequency of user comments	F_t^c
2	SMA of frequency of user comments	$FCSMA(t) = (\sum_{k=t-n+1}^{t} F_k^c)/n$ n=7
3	Average and standard deviation of comment length	$a_t^c = \frac{1}{m}\sum_{k=1}^{m} l_k^c$ $b_t^c = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(l_k^c - a_t^c)^2}$

The number of comments in all the posts per day F_t^c is extracted from user comments, where *t* is the date, and *c* is the stock of the company mentioned in the comments. The definition of SMA of frequency of comments (FCSMA) is shown in **Table 4-3**.

In addition, the longer the comment on a news item pertaining to a certain company, the more attention users are bound to focus on that company. Hence, we calculate the average and standard deviation of the length of user comments. Their definitions are shown in **Table 4-3**, where a_t^c is the average length of comments about stock *c* on date *t*, and b_t^c is the standard deviation of the length of comments on date *t*. The total number of comments on date *t* is *m* and l_k^c is the length of a comment about stock *c*.

4.3 proposed method

Our model first makes prediction of stock price change rates by multiply kernelized linear function and then infers trading position to take, i.e., sell, buy, or retreat for a while, by a linear function of the predicted value and technical indicators. The former regression function is learnt by MKR and the latter prediction function is learnt by GA. The model is expressed by the following:

$$pred(x_1, x_2, y_1) = \; < u, \phi(x_1, x_2, y_1) > -b \tag{4-1}$$

$$TDV(x_1, x_2, y_1, y_2, y_3, y_4) = \langle w, (pred(x_1, x_2, y_1), y_2, y_3, y_4) \rangle \tag{4-2}$$

$$S(x_1, x_2, y_1, y_2, y_3, y_4) = \begin{cases} +1 & \text{if } TDV(x_1, x_2, y_1, y_2, y_3, y_4) > \theta_{buy} \\ -1 & \text{if } TDV(x_1, x_2, y_1, y_2, y_3, y_4) < \theta_{sell} \\ 0 & \text{otherwise} \end{cases} \tag{4-3}$$

where u and w are weight vectors, b is an offset. x_1, x_2, and y_1 are vectors of features from news, user comments, historical trading data respectively. $(pred(x_1, x_2, y_1), y_2, y_3, y_4)$ is a vector of four elements: prediction of stock price change rate, and RSI, BIAS, and WPR of historical prices. $\langle \; \rangle$ means the dot product, and $\phi(\;)$ is the function that maps inputs to higher dimensional feature space and that accompanies a kernel function.

Note that $pred$ in **equation (4-1)** is a regression function for prediction. TDV, which stands for trading decision value, is a function to combine values of MKR prediction and values of three overbought and oversold indicators. S is the function that outputs trading signal. As a trading signal, $S(x_1, x_2, y_1, y_2, y_3, y_4) = +1$ designates "buy", $S(x_1, x_2, y_1, y_2, y_3, y_4) = -1$ designates "sell", while $S(x_1, x_2, y_1, y_2, y_3, y_4) = 0$ designates "no trade". The features x_1 and x_2 are those from SNS, y_1 to y_4 are from historical trading data. The threshold value θ_{buy} and θ_{sell} will be described in **Section 4.3.3**.

The structure of the proposed model is shown in **Fig.4-1**. It is composed of three parts:

- Raw data preprocessing
- Features extraction
- Trading signal generation

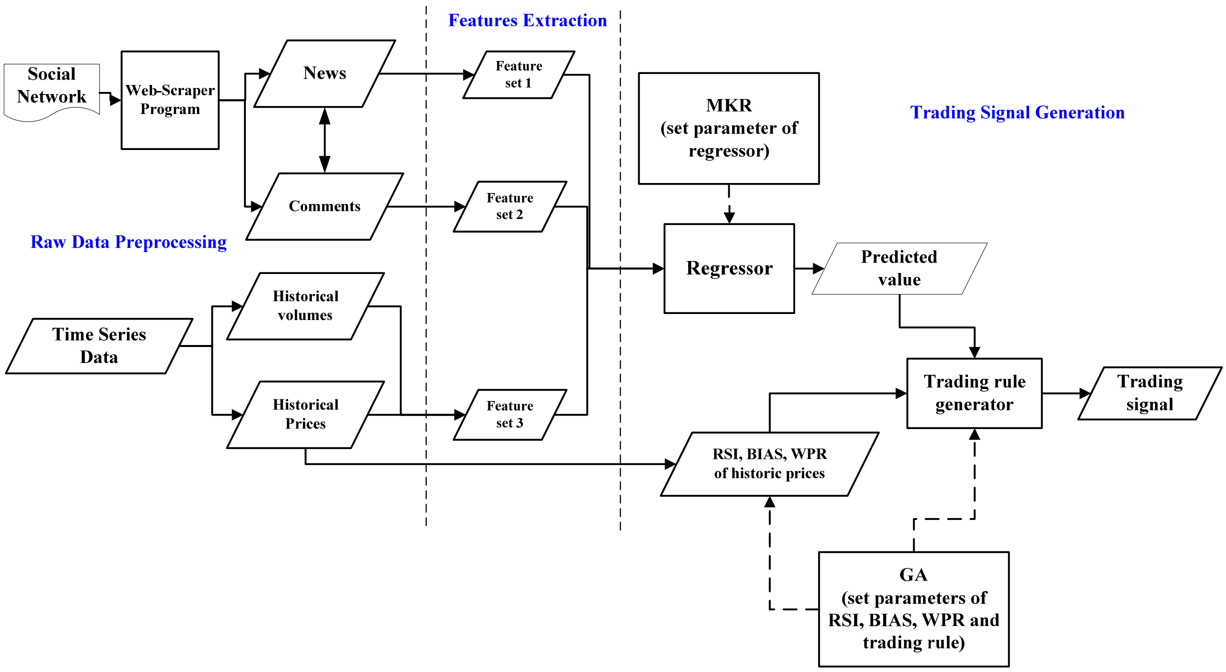

Fig. 4-1. Diagram of the proposed model

4.3.1 Raw data preprocessing (RDP)

The RDP part processes the raw data to be used for experiments. For the social network source, the RDP part downloads the news and user comments, including their contents and date of posting, for each day. The frequency of news and user comments and the comment length are calculated by a JAVA program. For the time series data source, the RDP part downloads the historical daily prices (opening, highest, lowest, and closing) and transaction volumes data from the Google Finance website.

4.3.2 Features extraction (FE)

The FE part extracts the features we need from the data downloaded using the RDP part. For the time series source, the FE part extracts the daily historical opening, highest, lowest, and closing prices and transaction volumes for three famous companies, Amazon, Sony and Sharp, dealing information technology (IT) products and services. For the social network source, the FE part extracts features of news and user comments which are downloaded using the RDP part.

4.3.3 Trading signal generation (TSG)

For the purposes of our study, input feature sets for the proposed model are composed of three parts: 1) features from historical prices and volumes, 2) features from news, and 3) features from user comments. In spite of extensive SVR applications in financial forecasting, SVR models did not address the challenges posed by multiple data sources or multiple representations. In contrast, MKR considers the linear combination of kernels, solves the convex optimization problem of linear combination of kernels, and is guaranteed to achieve the global optimal solution. Hence, theoretically, MKR models perform better than SVR models.

In TSG part, the MKR framework is used to learn a regression function based on these three feature sets and then, to predict the stock returns for the next trading day. In experiments, we use one linear kernel and one Gaussian kernel for each input feature set, and we simply set the default values for the parameters of the Gaussian kernel.

The output $PC(t)$ of the MKR in the testing period is the predicted stock price change rates at time t, which is:

$$PC(t) = \frac{Predict(t+1) - Price(t)}{Price(t)} \tag{4-4}$$

where $Predict(t+1)$ is the prediction for stock price for time $t+1$

Although the predicted stock price change rates can be used for simulated trading, in our preliminary experiments, the accumulated profits based on just the stock price change rate predictions were not good enough; the same was true for accumulated profits based on using just technical indicators (RSI, WPR, and BIAS). Considering the prediction and the technical indicators might have complementary components, we propose to combine them to get the trading signal.

After obtaining predictions of the stock price change rate, the proposed method uses a linear function to fuse the predicted stock price change rates and overbought and oversold technical indicators. Parameters of the linear function and technical indicators are learnt by GA. In the GA chromosome design, the accumulated return in the trading period is the fitness function. The TSG part finds a trading rule and executes it to generate a trading signal for the next trading.

The final trading decision value *TDV* is a linear combination of the overbought/oversold indicators and the stock price change rates predicted by MKR:

$$TDV = \sum_{i=1}^{N} w_i e_i \tag{4-5}$$

where w_i are the weights learned by the GA, and e_i are the values of the MKR as well as the values of the overbought/oversold technical indicators under consideration (RSI, WPR and BIAS). Note that the indicators here are expressed as a ratio. We use RSI/100, BIAS and WPR/100 from Table 1. Further, note that the MKR outputs are stock price change rates. By these conventions, the e_i in **equation (4-5)** is dimensionless and therefore, is consistent.

Once the weights and the other parameters are learned by the GA, we can obtain the decision values *TDV* on the days for the testing period. In the meanwhile, the threshold for buying (θ_{buy}) and the threshold for selling (θ_{sell}) are also learned by the GA. Then, based on the *TDV* and the threshold values for buying and selling, the trading rule of the proposed method is expressed in **Table 4-4**. The trading rule in Table 4-4 is equivalent to **equation (4-3)**. In addition, because the target prediction horizon is one day, our trading rule will simply close the position one day after we open it.

Table 4-4. Trading rule design of proposed model

Trading rule	***Rule in details***
Proposed model	If $TDV > \theta_{buy}$, the next trading position is "buy", else if $TDV < \theta_{sell}$, the next trading position is "sell", else (i.e., $\theta_{sell} \leq TDV \leq \theta_{buy}$) the next trading position is "no trade".

Procedures for the learning of trading rules in TSG part are as follows:

1) Obtain the one-day ahead stock price change rate prediction that was obtained by a regressor trained by MKR.
2) Create chromosomes randomly as the first generation. For every chromosome, apply the trading rules to the training data at every specified time in the training period, by calculating the value of technical indicators, computing the decision value *TDV*, and making decisions.
3) Compute the accumulated profits during the trading period as the fitness value. Reserve the top 10% of the chromosomes (those that make the top 10% in profit) directly for the next generation. Create new chromosomes using a crossover operation on the chromosomes selected from the current generation (with selection probability based on the fitness score of each chromosome). Repeat the crossover until a new generation is generated. Mutate or flip some bits of the chromosomes randomly.
4) Repeat steps 2 and 3 until the maximum number (100) of generations are generated or until the fitness of the best individual does not improve for 10 successive generations. Then choose the best chromosome as the one to represent the optimized trading rule. Calculate the return by applying the resulting trading rule on the testing data.

4.4 Experiment design

4.4.1 Data sources

Our data for training and testing are from two sources: Google Finance and Engadget. The historical time series data of the daily stock price (opening, closing, highest, and lowest) and daily transaction volumes were obtained from the Google Finance website. We downloaded the attributes pertaining to news items and user comments from the Engadget website.

In the experiment, we selected three companies' stocks: Amazon, Sony, and Sharp, which are important companies in US stock markets. Since Engadget is a blog network with daily coverage of gadgets and consumer electronics, it is ideally suited for our purpose.

Because we make daily predictions, the comments and news we use must be published by no later than 9:00 am of day (T + 1) to enable us to make a prediction for a said stock. For example, if a news item pertaining to Amazon is published on May 1, then there may be some comments on that news items published by users after 9:00 am on May 2. We cannot utilize such user comments since we attempt a one-day ahead prediction. We use news and user comments data for the three companies from January 1, 2006 to August 15, 2008.

4.4.2 Inputs for MKL

We use data of the preceding week (7 days, or (T – 6) to T) as the input features to predict stock price change rates on the next day, where T is the current date or one day before the predicted date. In other words, we assume that the stock price change rates of a company on a certain weekday depend on its stock price and communication activity pertaining to it in the preceding 7 days. We have three sets of input features: features from time series data (historical prices and volumes), features from news, and features from user comments.

The first feature set concerns the technical analysis of time series data. We would like to learn to predict changes in stock prices with the MKR framework by using the SMA, MACD, and ROC of the stock price and volume. The details of feature set 1 are shown in **Table 4-5**.

Table 4-5. Features from stock prices and volumes

Number	***Indicator***	***On***	***Description***
1	ROC	Closing price	ROC for historical prices, from day (T − 6) to T
2	ROC	Volume	ROC for historical volumes, from day (T − 6) to T
3	SMA	Closing price	SMA for historical prices, from day (T − 6) to T
4	SMA	Volume	SMA for historical volumes, from day (T − 6) to T
5	MACD	Closing price	MACD for historical prices, from day (T − 6) to T
6	MACD signal	Closing price	MACD signal for historical prices, from day (T − 6) to T

Feature set 2 and 3 consist of features from news dynamics and user comment dynamics, respectively. In this research, we used only numerical features and not, for example, text data. In **Table 4-6**, Feature Numbers 1 and 2 are from news and 3 to 5 are from user comments.

Table 4-6. Features from news dynamics and user comment dynamics

Number	***Feature***	***On***	***Description***
1	Frequency	News	Frequency of news, from (T − 6) to T
2	SMA	News	SMA of news, from (T − 6) to T
3	Frequency	User comments	Frequency of user comments from (T − 6) to T
4	SMA	User comments	SMA of user comments from (T − 6) to T
5	MA/Standard deviation	User comments	Average and standard deviation of comment length from (T − 6) to T

4.4.3 Chromosome design of the GA

Based on the trading rule design of proposed method, we design the chromosomes (shown in **Table 4-7**) for the trading rules that combine signals.

Table 4-7. Chromosome design of the GA model

Number	*Length*	*Value range*	*Meaning*
1	5 bits	−1 to 1	RSI weight
2	5 bits	−1 to 1	WPR weight
3	5 bits	−1 to 1	BIAS weight
4	5 bits	−1 to 1	MKR weight
5	5 bits	−1 to 1	Threshold value for buying
6	5 bits	−1 to 1	Threshold value for selling
7	4 bits	2 to 17	Parameter of RSI
8	4 bits	2 to 17	Parameter of WPR
9	4 bits	2 to 17	Parameter of BIAS

The representations of the genes are as follows:

1) Numbers 1 to 4 (20 bits in total) represent the weights for the three technical indicators and the MKR results. The weight values range from −1 to +1, where the least significant bit represents 2/32=0.0625.
2) Numbers 5 and 6 (10 bits in total) represent the threshold values for buying and selling. The range for each threshold is −1 to +1. The least significant bit represents 2/32=0.0625.
3) Numbers 7 to 9 (12 bits in total) represent the parameters of RSI, WPR, and BIAS. The values range from 2 to 17.

Before executing the GA training steps, the population size and maximum number of generations is set to 150 and 100, respectively. Individuals are initialized with random chromosomes following the gene structure shown in **Table 4-7**. The fitness value is the profit accumulated during the GA learning. In order to retain high-fitness individuals, the elite 10% (the top 10% of individuals in terms of fitness) were reserved automatically at every generation. Therefore, at every generation, the individuals that obtained top 10% of the profits will be reserved for the next generation.

4.4.4 **Rolling window method for training and testing**

To separate the training and testing periods, we use a rolling window method. We execute the MKR on data of 240 trading days (around 12 months) and obtain the predicted values for the following 160 trading days (around 8 months). The predictions of the first 80 trading days (around 4 months) are used for the GA training and of the remaining 80 trading days are used for the GA testing, i.e., to test the whole MKR-GA procedure (see **Table 4-8** and **Fig.4-2**). Then, for each subsequent experiment, we move both the training and testing periods forward by 80 trading days (around 4 months). There are, in total, five training and testing periods for each stock from January 1, 2006 to August 15, 2008.

Table 4-8. Training and testing period

Period	*Process*	*Length of period*
A	MKR training	240 trading days (around 12 months)
B	MKR testing (prediction)	160 trading days (around 8 months)
C	GA training	80 trading days (around 4 months)
D	GA testing (trading)	80 trading days (around 4 months)

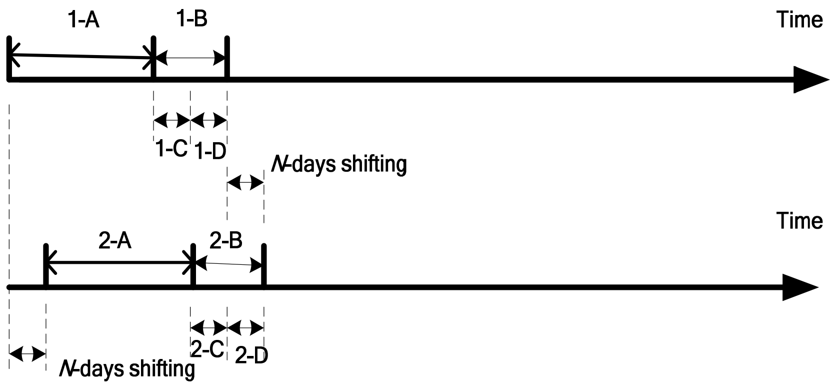

Fig. 4-2. Rolling window method for training and forecasting. (For example, 1-A means Period A (see Table 9) in 1st period)

4.4.5 Methods for comparison

We use the term baseline model to designate a simple model implemented easily. **Table 4-9** lists our proposed model MKR-all-GA model, and other models for comparison in the experiments including the baseline models. All models (except "Buy and Hold" and "Sell and Hold") are supposed to output −1, 0, or 1 as trading signal, when the predicted stock price change rate is negative, none, or positive, respectively. If the outputs are predicted stock price change rate, sign function is applied, i.e., the function that outputs −1, 0, or 1 if the argument is negative, 0, or positive values. The trading rule is, then, common to these models and simple: when prediction is −1, 0, or 1, a trading agent will sell the stock, wait, or buy the stock, respectively, at the start of predefined time frame, a trading day in the experiments and close the position at the end of the time frame.

Table 4-9. A list of models for stock change prediction and trading signal generation

	change rate prediction		*trading signal generation*		
Model	Features	Learning Method	Features	Learning Method	***Description***
MKR-all-GA (proposed)	news/comments/historical data	multiple kernel (MKR)	prediction/RSI/WPR/BIAS	GA	Integration of MKR and GA. Three feature sets are used in MKR
SVR-ts-GA	historical data	single kernel (SVR)	prediction/RSI/WPR/BIAS	GA	Integration of SVR and GA. Stock prices and transaction volumes are

					used in SVR
SVR-news-GA	news	single kernel (SVR)	prediction/RSI/WPR/BIAS	GA	Integration of SVR and GA. Features from news are used in SVR
SVR-coms-GA	user comments	single kernel (SVR)	prediction/RSI/WPR/BIAS	GA	Integration of SVR and GA. Features from user comments are used in SVR
SVR-all-GA	news/comments/historical data	single kernel (SVR)	prediction/RSI/WPR/BIAS	GA	Integration of SVR and GA. Three feature sets are used in SVR
SVR-ts-STS	historical data	single kernel (SVR)		no learning; sign function	SVR and a simple trading strategy. Features from stock prices and transaction volumes are used in SVR
SVR-news-STS	news	single kernel (SVR)		no learning; sign function	SVR and a simple trading strategy. Features from news are used in SVR
SVR-coms-STS	user comments	single kernel (SVR)		no learning; sign function	SVR and a simple trading strategy. Features from user comments are used in SVR
SVR-all-STS	news/comments/historical data	single kernel (SVR)		no learning; sign function	SVR and a simple trading strategy. Three feature sets are used in SVR
MKR-all-STS	news/comments/historical data	multiple kernel (MKR)		no learning; sign function	MKR and a simple trading strategy. Three feature sets are used in MKR
ANN-all-STS	news/comments/historical data	Artificial neural network (ANN)		no learning; sign function	ANN and a simple trading strategy. Three feature sets are used in ANN.
Buy and Hold	-	no learning		no learning	Buy, then hold the position, until the end of the testing period
Sell and Hold*	-	no learning		no learning	Sell, then hold the position, until the end of the testing period

*Note for "Sell and Hold", we do margin transaction

In **Table 4-9**, compared to the proposed model MKR-all-GA, models SVR-ts-GA, SVR-news-GA, SVR-coms-GA use the SVR as learner with only one input feature set. Model SVR-all-GA uses the same inputs as the proposed model, but it uses SVR as a learner instead of MKR. The following six models (SVR-ts-STS, SVR-news-STS, SVR-coms-STS, SVR-all-STS, MKR-all-STS, ANN-all-STS) are similar to the first five models, but their trading signal generation rule is simple and fixed, and, therefore, is named simple trading strategy (STS): if predicted price change rate is negative, 0, or positive, then −1, 0, or +1 is supposed to output respectively, which means the trading signal "sell", "no trade" or "buy" respectively. The two models at the bottom of Table 10 use the same rule with fixed trading position, i.e., simply buy or sell the target stocks and wait until the end of the testing period.

4.5 Experimental results

4.5.1 RMSE results for stock price change predictions

Every model except for "Buy and Hold" and "Sell and Hold" to be tested has prediction step. In this subsection we evaluate the accuracy of prediction results. Note that the models such as SVR-ts-GA and SVR-ts-STS have the same prediction method so that in this section these two models are summarized as SVR-ts.

Based on the average of RMSE results for the out-of-samples data of the stock price change rate of the three companies (see **Table 4-10**), it is obvious that in the testing periods, the average of RMSEs of the MKR-all for Sharp (0.03117), Amazon (0.01729), and Sony (0.01873) were smaller than that of the other models. It indicates that MKR-based model outperformed SVR-based model in predicting price change rates of stocks. Additionally, at times, the prediction results of SVR-all are not as good as those of methods with just one input: for Amazon, SVR-news (0.03890) and SVR-coms (0.03984) outperform SVR-all (0.04075). Similarly, for Sharp, SVR-news (0.02029) and SVR-coms (0.02251) outperform SVR-all (0.02307); and for Sony, SVR-news (0.02124) outperforms SVR-all (0.02371). SVR-all also utilizes the same information as MKR-based one, but their prediction results are even worse than some SVR-based models which utilize information from just one source. It indicates that if we use just one kernel, information from different kinds of sources could be harmful although SVM and SVR are thought to be robust to many relevant and irrelevant features.

Table 4-10. Average of RMSEs in the five out-of-sample testing periods for three stock price changes

Models	*Sharp*	*Amazon*	*Sony*
SVR-ts	0.04756	0.03295	0.03212
SVR-news	0.03890	0.02029	0.02124
SVR-coms	0.03984	0.02251	0.02560
SVR-all	0.04075	0.02307	0.02371
MKR-all	0.03117	0.01729	0.01873

4.5.2 Profit and loss results

Table 4-11 shows the average returns made by different models in the five testing periods.

Table 4-11. Average returns in the five out-of-sample testing periods (of four months each) in the ratio of the initial investment

Models	*Sharp*	*Amazon*	*Sony*
MKR-all-GA (proposed)	0.02143	0.14811	0.03797
SVR-ts-GA	0.01307	0.12478	-0.01702

SVR-news-GA	0.01425	0.08226	-0.02630
SVR-coms-GA	0.01343	0.13386	0.00116
SVR-all-GA	0.01358	0.02097	-0.02479
SVR-ts-STS	0.00399	-0.00258	-0.07979
SVR-news-STS	0.02224	-0.12041	-0.03325
SVR-coms-STS	0.16217	0.13640	-0.07926
SVR-all-STS	0.04633	-0.13022	-0.09349
MKR-all-STS	-0.03845	-0.25684	0.01630
ANN-all-STS	-0.01677	0.17738	0.02611
Buy and Hold	-0.03205	0.15849	0.00543
Sell and Hold	0.03205	-0.15849	-0.00544

First, we note that the results for the models with GA-optimized trading rule i.e., SVR-ts-GA, SVR-news-GA, SVR-coms-GA, SVR-all-GA, and MKR-all-GA, show that most of the average returns (12 of 15 periods) are positive. In **Table 4-11**, among the GA-optimized models, the proposed model (MKR-all-GA) yields the best returns for all three stocks, which may be attributed to the fact that MKR-all performed the best in the change rate prediction (in **Table 4-10**, we find MKR-all obtains best RMSE results than SVR-ts, SVR-news, SVR-coms, SVR-all for all three stocks). A comparison of these results evaluated by returns reveals only proposed model, MKR-all-GA, and SVR-coms-GA yielded profit for all three stocks, and if we compare each profit of them, proposed model performed better than the SVR-coms-GA for all three stocks.

Then, we focus on the models with simple trading strategy (SVR-ts-STS, SVR-news-STS, SVR-coms-STS, SVR-all-STS, MKR-all-STS, and ANN-all-STS). We found that while models using the STS could yield good profits for a stock, they also suffered huge losses for other stocks. For example, although SVR-coms-STS yielded an average profit of 16.21% for Sharp, it also suffered an average loss of −7.92% for Sony. In addition, none of the models with the STS made profits for all three stocks.

Finally, we focus on the “Buy and Hold” and the “Sell and Hold” models. For Amazon, “Buy and Hold” yielded an average profit of 15.84%, because its stock price rose significantly during the testing periods. “Sell and Hold” for Sharp yielded an average profit of 3.205%, because its stock price fell a little during the testing periods. However, whether the stock price goes up or goes down is hardly known, which is a starting point of our research on prediction. Neither “Buy and Hold” nor “Sell and Hold” made profits for all three stocks.

4.5.3 Effect of single information source on the returns of proposed method

In order to figure out which of the multiple information sources is utilized by the proposed algorithm MKR-all-GA to obtain returns than other methods did, we use a random sequence in place of one of the multiple information sources (time series, news or user comments), for getting to know which information source(s) was (were) in fact contributed or not to the return results yielded by the proposed method MKR-all-GA. **Table 4-12** shows a list of the benchmark methods used in additional experiments.

Table 4-12. List of the benchmark methods used in additional experiments

No	*Method*	*Description*
1	MKR-GA-noTS	The same as the proposed method MKR-all-GA, except that the time series data is replaced by a random sequence
2	MKR-GA-noNews	The same as the proposed method MKR-all-GA, except that the news data is replaced by a random sequence
3	MKR-GA-noComs	The same as the proposed method MKR-all-GA, except that the comments data is replaced by a random sequence

The results (average returns in five testing periods) of three additional benchmark methods and that of the proposed method MKR-all-GA are shown in **Table 4-13**. For Sharp, MKR-GA-noTS, MKR-GA-noNews and MKR-GA-noComs yielded an average return of 0.00567, 0.02092, and 0.00456, compared with 0.02143 of the proposed method. It indicates that the news information contributed only about 2.4%, while the information sources stock time series and user comments contributed about 75% of the trading results of the proposed method. For Amazon, the return 0.14477 of MKR-GA-noNews is very close to 0.14811 of the proposed method; therefore the news information contributed none of the returns of the proposed method. In contrast to it, the MKR-GA-noTS and MKR-GA-noComs yielded average returns 0.12705 and 0.08217, respectively. It indicates that comments and stock time series data contributed about 14% and 45%, respectively, to the returns of the proposed method. For Sony, MKR-GA-noTS and MKR-GA-noNews yielded average returns 0.03590 and 0.03072, compared with 0.03797 of the proposed MKR-all-GA, while MKR-GA-noComs obtained a loss return of −0.04034. It indicates that the comments information contributed about 100% while news information and stock time series information contributed none for the returns of Sony yielded by the proposed method

Table 4-13. Average returns in the five out-of-sample testing periods (of four months each) in the ratio of the initial investment

Model	*Sharp*	*Amazon*	*Sony*
MKR-all-GA (proposed)	0.02143	0.14811	0.03797
MKR-GA-noTS	0.00567	0.12705	0.03590
MKR-GA-noNews	0.02092	0.14477	0.03072
MKR-GA-noComs	0.00456	0.08217	−0.04034

4.5.4 Sharpe ratio results

We evaluated the Sharpe ratio values of the models for the five testing periods. The yield of 3-month United States treasury bills in secondary market (http://www.federalreserve.gov/releases/h15/data.htm) was used as the risk-free return, which was 1.18% (in average) per 3-month in 2006, 1.08% (in average) per 3-month in 2007, and 0.34% (in average) per 3-month in 2008. We calculated the average return of 2006 to 2008 and the average risk-free return of each testing data set (covert to four months return in average) was 1.16%. **Table 4-12** shows the Sharpe ratios of the returns for each model in the five testing periods.

Table 4-12. Sharpe ratios in the five testing data sets for the three stocks trading

Models	*Sharp*	*Amazon*	*Sony*
MKR-all-GA (proposed)	0.486201	0.396327	0.293529
SVR-ts-GA	0.01702	0.273714	-0.24266
SVR-news-GA	0.01907	0.199747	-0.16851
SVR-coms-GA	0.01992	0.355082	-0.01481
SVR-all-GA	0.01464	0.47166	-0.16203
SVR-ts-STS	-0.06608	-0.0738	-1.00542
SVR-news-STS	0.177327	-0.669	-0.17258
SVR-coms-STS	0.78592	0.250027	-0.42945
SVR-all-STS	0.553535	-0.47117	-0.93764
MKR-all-STS	-0.22801	-1.17413	0.02049
ANN-all-STS	-0.08673	0.580096	0.22462
Buy and Hold	-0.32521	0.409668	-0.016035
Sell and Hold	0.152361	-0.47437	-0.044259

A higher Sharpe ratio indicates a higher ratio between net return (asset return minus the risk-free return) and volatility. From **Table 4-12**, we find that the proposed model obtained positive Sharpe ratio values for all three stocks. Some models yielded better profits than our proposed model for some stocks (such as "Sell and Hold", SVR-coms-STS, and SVR-all-STS for Sharp). However, the Sharpe ratio values of "Sell and Hold" for Amazon and Sony, of SVR-coms-STS for Sony, and of SVR-all-STS for Amazon and Sony are negative, thus indicating that their average return is less than risk free return. In addition, SVR-coms-GA and the proposed model (MKR-all-GA) are the only two models with positive Sharpe ratios for all three stocks. Furthermore, on comparing the Sharpe ratios of these two models, we find that the proposed model has a higher Sharpe ratio than SVR-coms-GA for all three stocks. From the results for the average return and the Sharpe ratios, we confirm that the proposed MKL-all-GA model outperforms the baseline and other models in terms of return as well as the Sharpe ratio.

4.6 Conclusion

In this section, we proposed a model to generate heuristically optimized trading rules by utilizing social network activities and historical traded prices and transaction volumes. The proposed model extracts three kinds of features from multiple sources. Then it predicts the stock price change rates based on the MKR framework. Finally, GA finds trading rules based on the stock price change rate prediction and three overbought and oversold indicators. We evaluated the prediction and trading performances of the experimental results by RMSEs, accumulated returns, and Sharpe ratio. Experimental results indicate that our proposed model outperforms baseline and other models in stock price change rate prediction, accumulated returns, and Sharpe ratios for three technology companies.

This research is the first of its kind to apply MKR on time series data of prediction target and social network data for a training period. The results show that the prediction of the stock price change rates by MKR was better than that by SVR for all three stocks in terms of RMSEs. We then applied the GA to optimize a change direction predictor that uses the predicted stock price change rate and overbought/oversold indicators in the training periods. We conducted simulated trading of our target stocks and evaluated the results by accumulated returns and Sharpe ratios in the testing periods. From the results in **Tables 4-11 and 4-12**, it is clear that the proposed model outperforms baseline models ("Buy and Hold", "Sell and Hold"), and other models such as SVR with time series data, and SVR with time series data and GA-optimized change direction predictor. Although baseline models and other models outperformed the proposed model in some testing periods, e.g., SVR-coms-STS yielded 16.21% for Sharp, only our proposed method obtained good profits (2% to 14.8% profit per testing data set of around 4 months) and consistently positive Sharpe ratios (i.e., 0.29 to 0.48), which is better than SVR-coms-GA which also attained positive Sharpe ratios for the three stocks. In short, our proposed model obtained favorable returns with low volatility over all five testing periods for all stocks in our experiments. This indicates that the proposed model can be used as an effective approach to automatic trading.

5. Crude oil price forecasting based on multiple markets and time frames

5.1 Introduction

Crude oil is the world's most actively traded commodity, accounting for over 10% of total world trade **(Verleger, 1994)**. The reason for this large volume of trade in crude oil is two-fold: its key role in the world economy and the worldwide dependence on crude oil for meeting energy demands.

West Texas Intermediate (WTI) and Brent Crude oil market are two of the world's most important crude oil markets. While Brent Crude oil is sourced from the North Sea and is primarily used in Europe, WTI crude oil is refined mostly in the Midwest and Gulf Coast regions in the United States of America, and is mainly supplied to the North American market. Although crude oil prices in these two markets have a significant interrelationship, for instance, price fluctuations in one market impacts prices in the other price movements in these markets are not always similar because of differing crude oil quality characteristics and the diverse locations they cater to.

A fluctuation in crude oil prices may significantly impact a nation's economy. Forecasts assist in minimizing such risks arising from the uncertainty surrounding future crude oil prices. To this end, it is critical to engage in prediction exercises modeled for forecasting crude oil prices. Although many business practitioners and researchers have attempted to develop various forecasting methods to predict crude oil prices, it is extremely difficult to design a model that captures the various dimensions affecting future crude oil prices. Crude oil prices are strongly influenced by several factors, including gross domestic product (GDP) growth, political events, conflicts and wars, and financial policies relating to the US dollar (since crude oil is priced in US dollar), among others. Additionally, since crude oil sourced from different locations have varying qualities and transport costs at different rates are involved in shipping crude oil from one location to another, crude oil prices vary in different parts of the world. All these factors together contribute to strong fluctuations in the world market for crude oil, which has subsequently acquired the characteristics of complex nonlinearity, dynamic variation, and high irregularity.

Technical analysis is a way to forecast market prices of securities such as stocks based solely on the past prices and traded volumes, and technical indicators are usually used to do technical analysis. In the last few decades, numerous researchers **(Taylor and Allen, 1994; Brock et al., 1992; Wong et al., 2003)** have estimated and predicted movements of stock prices and foreign exchange rates based on technical indicators by using historical time series data. Several well-known technical indicators can be employed for finding trends in time series movements, the most famous being Moving Average (MA) trend indicator, on which many other technical indicators are based. Many researchers have applied the technical analysis method to identify trends in the time series of crude oil prices. For example, Park and Irwin **(Park and Irwin, 2010)** applied several technical indicators in order to generate trading rules for crude oil prediction and trading. Aldea **(1997)** successfully developed a trading system for crude oil prediction based on technical analysis. Successful studies such as these indicate that technical indicators are useful tools for mining and identifying useful patterns in original time series data.

Other researchers have used econometric models or traditional time series analysis methods such as co-integration analysis and autoregressive integrated moving average (ARIMA) for forecasting prices. For example, Huntington **(Huntington, 1994)** applied a sophisticated econometric model to predict crude oil prices. Gulen **(1998)** used co-integration analysis to predict the price of West Texas Intermediate (WTI) crude oil. Contreras et al. **(2003)** applied the ARIMA model to predict electricity price by analyzing time series data. These models register good prediction performance when the price series under study is linear or nearly linear, and may not be appropriate for forecasting future fluctuations in crude oil markets and prices, marked by nonlinearity and irregularity **(Wang and Yang, 2010)**.

Since these models based on the linearity assumptions are not suitable for approximation of nonlinear patterns hidden in crude oil price series, this study has applied nonlinear models to predict crude oil prices. Some machine learning methods such as artificial neural networks (ANNs) and support vector machines (SVM) were proposed to solve the nonlinearity problems of time series and gave better results than conventional methods. For example, many researchers applied ANN based models **(Haidai et al., 2008; Yu et al., 2008; Wang and Wan, 2008; Jammaizi and Aloui, 2012)**. Pierdzioch et al. **(2010, 2013)** forecasted oil price under asymmetric loss and found new evidence of anti-herding of oil price forecasters. Xie et al. **(2006)** proposed a new method for crude oil price prediction based on an SVM model and Xiao-lin and Hai-wei **(2012)** applied SVM to predict crude oil prices. Although these methods have been observed to provide better solutions in predicting nonlinear crude oil price movements, they suffer from some limitations. The ANNs model often suffers from local minima and overfitting problems, while the models based on SVM, in spite of extensive applications in crude oil price forecasting, do not address the challenge of learning from multiple sources or different representations of the same source. SVM models have difficulty in fusing information and features drawn from different sources (e.g., different crude oil markets) or different representations (e.g., different time frames) of the target source, often with varying properties, thus making prediction problematic.

In recent years, some researchers have applied the multiple kernel learning (MKL) **(Bach et al., 2004)** method to address the problem of selecting suitable kernels for different feature sets from different data sources. An advantage of using the MKL method is that it allows combination of different kernels for different input features. Additionally, MKL mitigates the risk of erroneous kernel selection to some extent by taking a set of kernels and assigning a weight to each kernel, ensuring that predictions are based on the derived weighted sum of the kernels. MKL further solves the convex optimization problem of linear combination of single kernels and is guaranteed to achieve global optima. Hence, the MKL models theoretically show better performance than SVM. Moreover, MKL learns the coefficients of kernels from different data sources, and the relationships among them are learned in the meanwhile. Some researchers have applied MKL to predictions in foreign exchange (FX) or stock markets. For example, Fletcher et al. **(2010)** applied MKL to the limit order book for predicting and trading on the currency pair of EUR/USD. Luss and d'Aspremont **(2012)** applied MKL for predicting abnormal returns from historical stock prices data and news. Deng et al. **(2011)** used MKL to fuse information from stock data and social networks for stock price prediction. Yeh et al. **(2011)** applied MKL to predict stock prices of the Taiwan stock market and obtained better results than outcomes from conventional methods. Although MKL models have shown great potential, no research has been undertaken to study the application of MKL in prediction of crude oil prices. Owing to the several advantages of MKL and its good prediction performance in the studies mentioned above, applying MKL for crude oil price forecasting with multiple data sources and different representations holds great promise.

For the purpose of forecasting crude oil prices by considering features from different sources and different representations, we propose to extract and use the features from two main crude oil spot markets and three different time frames. The two markets in this context are WTI and Brent Crude oil markets, the two largest crude oil markets in the world. Although WTI crude oil is mainly supplied to North America and Brent Crude oil is mainly used in Europe, some interrelationship between these two markets cannot be ruled out, given the interdependence of worldwide oil markets in the highly integrated contemporary global economic system. For instance, the fluctuations in one market do not go unnoticed in the other market. Therefore, there is a strong case for referring to price movements in the other market for predicting crude oil prices in a particular market. In addition to extracting features from two different crude oil markets, the features of different time frames are also considered as useful information for prediction.

In order to predict crude oil prices (WTI or Brent) in the target market, this study uses features from other crude oil markets besides features of the target market, and examines features from two time horizons other than the target time frame. Features from different sources or features of different time representations may have different properties and quality characteristics. Given its efficient prediction performance observed in studies mentioned earlier, the MKL model has been used in our study to address the problem of fusing information from different crude oil markets and time frames.

5.2 Proposed method

The proposed model is composed of three components as shown in **Fig. 5-1**:

- Feature extraction (FE) component
- Multiple kernel regression/prediction (MKRP) component
- Performance evaluation (PE) component

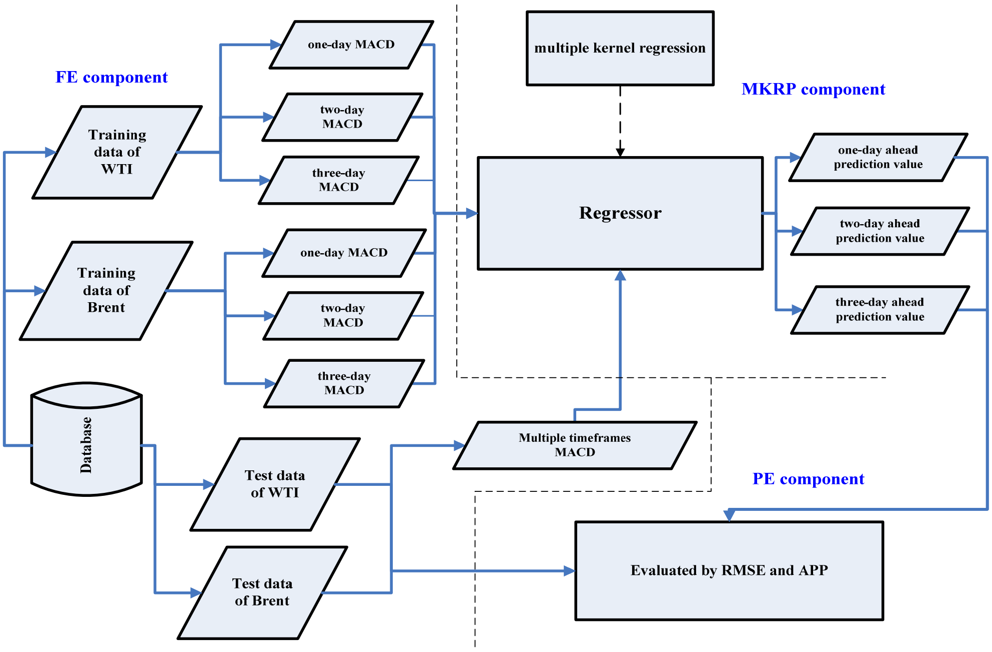

Fig.5-1. Structure of proposed model

The FE component first transforms crude oil spot price to MACD and MACD signals, following which it extracts features (historical n-days MACD features, see **Table 5-1**) from the two main crude oil markets and three different time frames.

The MKRP component then predicts the crude oil price by fusing information from the two crude oil markets and three different time frames. In this research, we have tested the forecasting ability of the proposed model on the basis of one-day, two-day, and three-day ahead predictions, while previous studies usually focused only on one-day ahead prediction. Since the trend of different time frames might be overlapped with each other, we used the features of multiple time frames for prediction.

For MKR, the input features are extracted from two different sources: WTI and Brent Crude oil prices. Since WTI and Brent Crude oil are the two biggest crude oil markets in the world, we selected these two sources. We transformed the original spot prices to MACD and MACD signal. For each kernel, the inputs are 4-period MACD values and MACD signals calculated from different time frames. Details is presented in **Table 5-1** and **Fig. 5-1**.

Table 5-1. Features for each kernel in the MKL framework

No.	*Feature*	*No.*	*Feature*
1	MACD value at time t	5	MACD value at time $(t-2)$
2	MACD signal at time t	6	MACD signal at time $(t-3)$
3	MACD value at time $(t-1)$	7	MACD value at time $(t-3)$
4	MACD signal at time $(t-1)$	8	MACD signal at time $(t-3)$

Finally, the PE component evaluates the prediction and trading results based on the two evaluation criteria.

5.3 Experiment design

5.3.1 Research data and experiment platform

There is a number of crude oil price series in the world. Of these various series, two main crude oil price series, WTI crude oil spot price and Brent Crude oil spot price, are chosen as experimental samples. There are two primary reasons why these two are chosen as crude oil price sources for our study. First, these two crude oil markets have maximum impact on the world economy; hence, these forecasts would be useful for many countries in the world. Second, since fluctuations in one market could be an important reference for the other, both these markets have been used for the experiment. This study uses daily spot prices obtained from the energy information administration (EIA) website of the US Department of Energy (DOE) (http://www. eia.doe.gov/). Note that this data includes only the spot price data for each working day. The WTI crude oil spot price information we obtained from the website is from January 2, 1986 through January 2, 2012, and the Brent Crude oil spot price information is from January 2, 1987 to January 2, 2012. The difference in duration for which data was collected is because of reasons pertaining to data availability - the EIA website provides Brent data from May 20, 1987. Moreover, since this study uses information from both sources for prediction of price series, for reasons of convenience, we have considered data for the period ranging from January 2, 1990 to December 31, 2011, in both cases.

The platform for experiment is Ubuntu, R language. MKL shogun package **(Sonnenburg et al., 2010)** is installed for MKL experiments.

5.3.2 Data sets and multiple step-ahead predictions

We used a rolling window method to separate the training and testing period. Since we want to have about 10 to 20 pairs of training and testing data and about one year for testing in the experiments, we decided to perform regression on data relating to 2048 trading days (around 8 years) and obtained predicted values for 256 trading days (around 1 year). Further, for each subsequent experiment, we moved both the training and testing period forward by 256 trading days (around 1 year). There is a total of 14 training and testing periods for WTI and Brent crude oil price prediction from the beginning of 1990 to the end of 2011. The training and testing period and their relations are shown in **Fig. 5-2**.

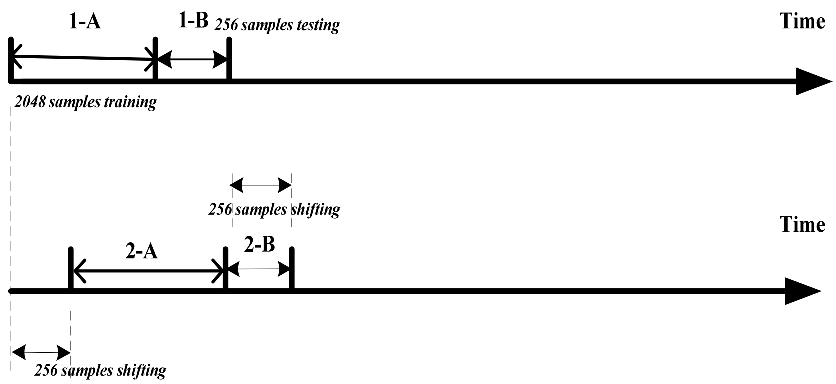

Fig. 5-2. Rolling window method for training and forecasting (*n*-A means the *n*th training period and *n*-B means the *n*th testing period)

Additionally, we tested the prediction ability of the proposed model by conducting forecasts for three different days: one-day, two-day, and three-day ahead predictions. For example, for testing two-day ahead prediction, we predict the crude oil price two days later; for trading, we hold the trading position for two days and close the position after two days.

5.3.3 Proposed and benchmark methods

A list of proposed and benchmark methods is shown in **Table 5-2**. The "M" in the method name means "using both the Brent and WTI crude oil price data" while the "S" means "using only the target market price data". We use the term benchmark method to indicate high performance in case of a commonly used method. In the following list, MKR-M-3 and MKR-S-3 are our proposed methods and others are benchmark methods. Method 1 uses SVR as the learner, but only one source is used (the features from Brent are used to predict Brent Crude oil price) and only one time frame is considered (to predict price two-day ahead, features of two-day time frame are used); Method 2, SVR-S-3, uses features from three different time frames of the target crude oil source. Method 3 uses features from two crude oil markets and from all the three time frames. Method 4 uses the MKR framework and only one data source, that is, the target crude oil market, and three different time frames. Method 5 represents another application of our proposed model and uses the same features as Method 3, but applies a MKR framework to all the information. Method 4 can be compared with Method 5 to test whether the additional features from other relevant markets are useful for prediction or not. Since MKR-S-1 uses only target market and only the target time frame, it is the same as method SVR-S-1.

Table 5-2. A list of proposed and benchmark methods

No.	*Method name*	*Data source*	*Time frames*	*Prediction method*
1	SVR-S-1	Only the target market	Only the target time frame	Support vector regression (SVR)
2	SVR-S-3	Only the target market	Three time frames	Support vector regression (SVR)
3	SVR-M-3	Both the markets	Three time frames	Support vector regression (SVR)
4	MKR-S-3	Only the target market	Three time frames	Multiple kernel regression (MKR)
5	MKR-M-3	Both the markets	Three time frames	Multiple kernel regression (MKR)

5.4 Experiment results

5.4.1 Prediction results for Brent and WTI crude oil

Tables 5-3 and 5-4 show the average RMSE/mean and standard deviation of RMSE/mean results for Brent crude oil price prediction. Similarly, **Tables 5-5 and 5-6** show the average RMSE/mean and standard deviation of RMSE/mean results for WTI.

Table 5-3. Average RMSE/mean results for Brent Crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.02796	1.99121	2.29711
SVR-S-3	0.05688	3.34386	4.46337
SVR-M-3	0.06629	5.17562	6.16460
MKR-S-3	0.02316	1.60254	1.97642
MKR-M-3	0.02316	1.60211	1.97584

Table 5-4. Standard deviation of RMSE/mean for Brent Crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.01011	1.61699	1.76454
SVR-S-3	0.06187	2.68442	3.99079
SVR-M-3	0.02494	3.71164	4.43922
MKR-S-3	0.00447	0.95943	1.18787
MKR-M-3	0.00447	0.95845	1.18666

From average RMSE/mean results of the total 14 experiments conducted ("mean" is the average crude oil price in the corresponding testing periods), the results of experiments based on the MKR model (MKR-S-3 and MKR-M-3) showed the best prediction results (the lower the value, better the score in RMSE). Additionally, we found that although the SVR-S-3 and SVR-M-3 use more features from more sources than SVR-S-1, forecasts of these methods is less accurate than SVR-S-1. In fact, SVR-S-3 has additional features of two more time frames and SVR-M-3 has additional features of another crude oil market and two more time frames. The reason for this could be that SVR has failed to fuse the information from different sources and/or different time frames. On the contrary, the MKR based methods (MKR-S-3 and MKR-M-3) yield better results than SVR- based methods (SVR-S-1, SVR-S-3, and SVR-M-3), indicating that information from different sources and different representations are useful for predictions. Moreover, this also indicates that, the MKR-based prediction methods fused the entire information more effectively than SVR-based methods.

Furthermore, since average RMSE/mean of MKR-M-3 and MKR-S-3 are not very different, we can conclude that the additional data source other than the target is not very useful. Moreover, on observing the standard deviation results, we find that the results of MKR-S-3 and MKR-M-3 are smaller than that of SVR-S-1, SVR-S-3, and SVR-M-3 for different prediction days (one-day, two-day, and three-day ahead). This indicates that the MKR-based model not only outperforms the SVR based model in terms of the magnitude of prediction, but it also attains low volatility. Similar conclusions could be derived for WTI crude oil price prediction from results shown in **Tables 5-5 and 5-6.**

Table 5-5. Average RMSE/mean results for WTI Crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.02856	2.02755	2.33543
SVR-S-3	0.04904	3.49764	4.14796
SVR-M-3	0.07788	5.83853	6.71285
MKR-S-3	0.02450	1.70738	2.06678
MKR-M-3	0.02450	1.70724	2.06662

Table 5-6. Standard deviation of RMSE/mean for WTI Crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.01050	1.47208	1.57962
SVR-S-3	0.03186	3.20480	3.45841
SVR-M-3	0.03384	4.58162	5.72177
MKR-S-3	0.00510	1.05566	1.24878
MKR-M-3	0.00510	1.05577	1.24895

5.4.2 APP results for Brent and WTI crude oil

Tables 5-7 and 5-8 show the APP results (the definition of APP is shown in section 2.2) for Brent and WTI prediction. First, we focus on the APP results of SVR based methods (SVR-S-1, SVR-S-3, and SVR-M-3). For Brent, SVR-S-1 method yields average returns at about 0.24% per day for one-day ahead prediction, about 0.21% per day for two-day ahead prediction, and about 0.14% per day for three-day ahead prediction, which indicates that although SVR used features only from the target crude oil market and target time frame, it is a promising method for making profits in crude oil trading. SVR-S-3 and SVR-M-3 used more information (features from another crude oil market or other time frames, or both) than SVR-S-1, but they presented worse results than SVR-S-1. The reason for this could be that the SVR was unable to effectively fuse information from different sources or different representations. Similar conclusion could be derived from the APP results of WTI shown in **Table 5-8**.

Table 5-7. Average APP results for Brent crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.00242	0.00207	0.00138
SVR-S-3	−0.00132	−0.00057	−0.00036
SVR-M-3	−0.00154	−0.00071	−0.00040
MKR-S-3	0.01038	0.00469	0.00282
MKR-M-3	0.01054	0.00464	0.00280

Table 5-8. Average APP results for WTI crude oil price prediction (total 14 experiments)

Method	*one-day ahead*	*two-day ahead*	*three-day ahead*
SVR-S-1	0.00367	0.00176	0.00117
SVR-S-3	0.00057	−0.00005	0.00009
SVR-M-3	−0.00143	−0.00088	−0.00044
MKR-S-3	0.01390	0.00546	0.00344
MKR-M-3	0.01309	0.00556	0.00348

We now focus on the results for varying time horizons. From **Tables 5-7 and 5-8**, we find that for trading from three different time horizons, proposed method (MKR-S-3 and MKR-M-3) yields the best average APP results (14 experiments overall). For example, from APP results of WTI, MKR-S-3 and MKR-M-3 yield about 1.3% for one-day ahead prediction, 0.54% for two-day ahead prediction, and 0.34% for three-day ahead prediction. Since the results of MKR-S-3 and MKR-M-3 are very close, we can say that the data source other than the target is not useful.

Next, we focus on the results of different trading horizons for the same method. Note that since APP refers to average percentage profit per day. APP results for different time horizons of the same method are comparable. For WTI crude oil, one-day ahead prediction yields best average profit per day for SVR-S-1, SVR-S-3, MKR-S-3, and MKR-M-3, and worst results for SVR-M-3. We can draw similar conclusions on observing the APP results for Brent Crude oil: across all methods, one-day ahead prediction yields the best average profit per day than two-day or three-day ahead prediction yield. However, we found that for Brent and WTI, MKR-M-3 and MKR-S-3 yields very close profits, which indicates that features from another market do not necessarily improve trading performance.

Finally, we compare the results of our proposed method for WTI and Brent. For WTI, proposed method MKR-M-3 yields about 1.30%, 0.55%, and 0.34% per day for one-day, two-day, and three-day ahead predictions, respectively. For Brent, it yields about 1.05%, 0.46%, and 0.28% per day for one-day, two-day, and three-day ahead predictions, respectively. This indicates that for each prediction based on time horizon, the proposed method produced better results when applied to WTI spot price, rather than Brent spot price.

5.5 Conclusion

In this section, we have proposed an MKL-based crude oil prediction method, which includes three components: feature extraction (FE), multiple kernel regression for prediction (MKRP), and performance evaluation (PE). In this study, the FE component first extracts features as MACD indicator from two crude oil sources and three different time frames. Second, the MKRP component predicts the crude oil prices by employing MKR. Finally, the PE component evaluates the prediction results by using RMSE and APP. Experimental results based on data from WTI and Brent Crude oil market show that MKR-based methods outperform benchmark methods on one-day ahead, two-day ahead, and three-day ahead predictions.

Experimental results show that prediction method based on the MKR framework yields better results than those obtained from SVR. Our study also detected that in case information is extracted from more than one source and/or different representations, SVR fails to effectively fuse the information, resulting in even more inaccurate results than those produced by employing the SVR method that used information from only a single source, pertaining to a single time frame. On the contrary, methods based on the MKR framework effectively fused information from different sources and different representations, and produced better results than the benchmark methods, with the exception that the additional data source did not add to the effectiveness of the forecast. However, we first believed that the knowledge of another market price movements is beneficial for a trader (therefore we conducted experiments) but in fact, if the knowledge of one market price movement is highly utilized, the knowledge of another market price movement one day ago is not useful at least for the case we experimented. The reason might be that the two markets are correlated almost in real time.

The coefficients that we obtained from the MKL regression function for crude oil price prediction, using data from different crude oil markets and time frames, demonstrated a possible correlation between our target crude oil market (WTI or Brent) and its target prediction time horizons (one-day, two-day, or three-day ahead), with other crude markets or other time frames. The relative value of coefficients of the kernels in MKL results could be utilized to see possible correlations between reference crude oil markets with reference time frames and the target crude oil market with the target prediction time horizon. As the time horizon goes on extending, coefficients of each feature set become unstable and the average percentage profit (APP) results become weak, indicating the difficulty with predicting crude oil price in longer time frames.

Future work in this field may take several interesting directions. For example, other than crude oil prices, stock prices of USA, main European stock markets, exchange rates of EUR/USD and USD/JPY are considered as useful information for predicting crude oil prices. Besides exploring some of these determinants of crude oil prices, possibility of incorporating features from more than three time frames and including more stages in the step-ahead prediction model, could be investigated by future studies.

6. Conclusion and future work

We first applied the proposed hybrid model on short term FX rate prediction and simulation trading. First, we applied MK regression to FX data for a specific training period to estimate the optimal parameters and weights. The results showed that the FX rate changes predicted by MK regression were much better than those predicted by SVR during our testing period, in terms of the RMSEs. Next, we used the GA to optimize the trading strategy using the predicted FX rate changes and the overbought/oversold indicators for the training period. We traded USD/JPY based on the trading strategies generated using the GA, and calculating the returns and Sharpe ratios for the testing periods from March to December between 2008 and 2011. Each testing period was 250 hours (around 2 weeks). In some testing periods, several baseline methods outperformed our proposed methods in terms of profit, but our proposed methods obtained consistently good profits and Sharpe ratios without experiencing losses in any year. The profits ranged from 7.1% to 20.60% per year from 2008 to 2011, and the Sharpe ratios for MKL-3-GA and MKL-5-GA were 4.36 and 2.49, respectively. The average profit obtained during the testing period was positive with a statistically high confidence (higher than 99% confidence). In short, our proposed method obtained consistently favorable returns with low volatility over a four-year period.

We then applied the proposed hybrid model to generate heuristically optimized trading rules by utilizing social network activities and historical traded prices and transaction volumes. The proposed model extracts three kinds of features from multiple sources. Then it predicts the stock price change rates based on the MKR framework. Finally, GA finds trading rules based on the stock price change rate prediction and three overbought and oversold indicators. We evaluated the prediction and trading performances of the experimental results by RMSEs, accumulated returns, and Sharpe ratio. Experimental results indicate that our proposed model outperforms baseline and other models in stock price change rate prediction, accumulated returns, and Sharpe ratios for three technology companies. This research is the first of its kind to apply MKR on time series data of prediction target and social network data for a training period. The results show that the prediction of the stock price change rates by MKR was better than that by SVR for all three stocks in terms of RMSEs. We then applied the GA to optimize a change direction predictor that uses the predicted stock price change rate and overbought/oversold indicators in the training periods. We conducted simulated trading of our target stocks and evaluated the results by accumulated returns and Sharpe ratios in the testing periods. From the results, it is clear that the proposed model outperforms baseline models (“Buy and Hold”, “Sell and Hold”), and other models such as SVR with time series data, and SVR with time series data and GA-optimized change direction predictor. Although baseline models and other models outperformed the proposed model in some testing periods, e.g., SVR-coms-STS yielded 16.21% for Sharp, only our proposed method obtained good profits (2% to 14.8% profit per testing data set of around 4 months) and consistently positive Sharpe ratios (i.e., 0.23 to 0.38), which is better than SVR-coms-GA which also attained positive Sharpe ratios for the three stocks. In short, our proposed model obtained favorable returns with low volatility over all five testing periods for all stocks in our experiments. This indicates that the proposed model can be used as an effective approach to automatic trading.

Furthermore we applied MKL method on the prediction and simulation trading in crude oil market with multiple time horizons. Experimental results show that prediction method based on the MKR framework yields better results than those obtained from SVR. Our study also detected that in case information is extracted from more than one source and/or different representations, SVR fails to effectively fuse the information, resulting in even more inaccurate results than those produced by employing the SVR method that used information from only a single source, pertaining to a single time frame. On the contrary, methods based on the MKR framework effectively fused information from different sources and different representations, and produced better results than the benchmark methods, with the exception that the additional data source did not add to the effectiveness of the forecast. However, we first believed that the knowledge of another market price movements is beneficial for a trader (therefore we conducted experiments) but in fact, if the knowledge of one market price movement is highly utilized, the knowledge of another market price movement one day ago is not useful at least for the case we experimented. The reason might be that the two markets are correlated almost in real time. The coefficients that we obtained from the MKL regression function for crude oil price prediction, using data from different crude oil markets and time frames, demonstrated a possible correlation between our target crude oil market (WTI or Brent) and its target prediction time horizons (one-day, two-day, or three-day ahead), with other crude markets or other time frames. The relative value of coefficients of the kernels in MKL results could be utilized to see possible correlations between reference crude oil markets with reference time frames and the target crude oil market with the target prediction time horizon. As the time horizon goes on extending, coefficients of each feature set become unstable and the average percentage profit (APP) results become weak, indicating the difficulty with predicting crude oil price in longer time frames.

The experiments shown in the application of three different kinds of financial markets indicate that MKL-GA or MKL are good methods for prediction and simulation. However, there are still some problems that remain for future work. For example, in the case of FX rate prediction, there is an problem about determination of the best time frame lengths for use as the references in our target trading time time framesince the selection of a very long or very short time frame may also have negative effect on the prediction, the lengths of the time frame could be set as parameters to be optimized by GA or some other machine learning methods; We only determined the relative weights among currency pairs with certain time frames, because we did not consider all of the possible time frames for all of the different pairs during MKL training, there is research direction to solve this problem by implanting a new MKL method which can distribute coefficients to a large number of kernels (time frames) and discard the kernels (time frames) that have relatively very small coefficients after learning step, for selecting useful time frames automatically. In addition, changing the input features might produce different weights for different currency pairs and different time frames (we used the MACD indicator as the feature for MK regression). Changing the input features (using other indicators or other transforms from raw data) may have negative effect. A possible way to solve this problem could be using technical indicators as many as possible and design a new MKL method for useful features selection. In addition, the information sources we used for crude oil price, stock price, or FX rate predictions may not enough. A possible research direction is to propose new method to combine features from relevant information sources as many as possible. For example, for crude oil price prediction, stock prices of USA, main European stock markets, exchange rates of EUR/USD and USD/JPY are considered as useful information for predicting crude oil prices. For information from SNS, we may use information from more famous SNS which are used by a huge number of people such as Facebook or Twitter. Besides exploring some of these determinants of crude oil prices, possibility of incorporating features from more than three time frames and including more stages in the step-ahead prediction model, could be investigated by future studies.

Acknowledgements

First I want to thank my advisor Prof. Sakurai. I appreciate all his contributions of time and ideas to make my Ph.D. experience productive.

In addition, this research was supported by the fund "Keio Leading-edge Laboratory of Science & Technology Research Grant", "Global-COE program (Symbiotic, Safe and Secure System Design)", and "Graduate School Doctoral Student Grant-in-Aid Program" of Keio University, Japan. I would like to thank their funding supports.

Furthermore, I would like to thank ICAP, Forexite, Google Finance, Engadget, EIA for making their financial data (FX rates, stock prices/volumes, crude oil prices) and social network service data available for this research.

Lastly, I would like to show my sincere thanks to Prof. Yamaguchi, Prof. Hibiki, and Prof. Hagiwara for examining my doctor dissertation.

October 2014

References

Aldea, C. E. *Technical analysis-based futures trading system* (Unpublished doctoral dissertation). Simon Fraser University, **1997**.

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of financial Economics*, *51*(2), 245-271.

Azoff, E. M. (1994). *Neural networks time series forecasting of financial markets*. John Wiley and Sons.

Bach, F. R., Lanckriet, G. R., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning* (p. 6). ACM.

Badawy, F. A., Abdelazim, H. Y., Darwish, M. G. (2005). Genetic algorithms for predicting the Egyptian stock market. *Proceedings of ITI Third International Conference on Information and Communication Technology Enabling Technologies for the New Knowledge Society*, 109–122.

Banzhaf, W., Nordin, P., Keller, R. E. Francone, F. D. (1998). *Genetic Programming – An introduction: on the automatic evolution of computer programs and its applications*. Kaufmann, San Francisco.

Barnett, W. A., Gallant, A. R., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T., & Jensen, M. J. (1997). A single-blind controlled competition among tests for nonlinearity and chaos. *Journal of econometrics*, 82(1), 157-192.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1-8.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics*, 31, 307–327.

Bonilla, C. A., Romero-Meza , R., and Maquieira, C. (2011). Nonlinearities and GARCH inadequacy for modeling stock market returns: empirical evidence from Latin America, *Macroeconomic Dynamics*, 15(5), 713-724.

Box, G. E. P., Jenkins, G. M. (1994). *Time series analysis: Forecasting and control* (third edition). Englewood Cliffs: Prentice Hall.

Brock, W., Lakonishok, J., & LeBaron, B. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, **1992**; *47*(5), 1731-1764.

Broock, W. A., Scheinkman, J. A., Dechert, W. D., & LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*, 15(3), 197-235.

Cao, L. (2003). Support vector machines experts for time series forecasting. *Neurocomputing*, *51*, 321-339.

Champernowne, D. G. (1948). Sampling theory applied to autoregressive schemes. *Journal of the Royal Statistical Society: Series B*, 10, 204–231.

Chang Chien, Y. W., & Chen, Y. L. (2010). Mining associative classification rules with stock trading data–A GA-based method. *Knowledge-Based Systems*, *23*(6), 605-614.

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

Chu, S. C. W., Ng, H. S., Lam, K. P. (2000). Intelligent trading advisor. *Proceedings of 2000 IEEE international conference on management of innovation and technology*, 1, 53–58.

Contreras, J., Espinola, R., Nogales, F. J., & Conejo, A. J. ARIMA models to predict next-day electricity prices. *Power Systems, IEEE Transactions on*, **2003**; *18*(3), 1014-1020.

De Choudhury, M., Sundaram, H., John, A., & Seligmann, D. D. (2008). Can blog communication dynamics be correlated with stock market activity?. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia* (pp. 55-60). ACM.

De Choudhury, M., Sundaram, H., John, A., & Seligmann, D. D. (2007). Contextual prediction of communication flow in social networks. In *Proceedings of the IEEE/WIC/ACM international Conference on Web intelligence* (pp. 57-65). IEEE Computer Society.

De Faria, E. L., Albuquerque, M. P., Gonzalez, J. L., Cavalcante, J. T. P., Albuquerque, M. P. (2009). Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods. *Expert Systems with Applications*, 36 (10), 12506–12509.

Dempster, M. A. H., Jones, C. M. (2002). Can channel pattern trading be profitably automated? *The European Journal of Finance*, 8 (3), 275–301

Deng, S., & Sakurai, A. (2013). Foreign Exchange Trading Rules Using a Single Technical Indicator from Multiple Time frames. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on* (pp. 207-212). IEEE.

Deng, S., Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A. Multiple kernel learning on time series data and social networks for stock price prediction. In *Machine Learning and Applications and Workshops (ICMLA),* **2011** *10th International Conference on* (Vol. 2, pp. 228-234). IEEE.

Deng, S., Sakurai, A. (2011). Combining Multiple Kernel Learning and Genetic Algorithm for Forecasting Short Time Foreign Exchange Rate. *Proceedings of AIA 2011*. DOI: 10.2316/P.2011.717-111

Deng, S., Yoshiyama, K., Mitsubuchi, T., & Sakurai, A. (2013). Hybrid Method of Multiple Kernel Learning and Genetic Algorithm for Forecasting Short-Term Foreign Exchange Rates. *Computational Economics*, 1-41.

Drożdż, S., Górski, A.Z., Kwapień, J. (2007). World currency exchange rate cross-correlations, *The European Physical Journal B*, 58 (4), 499–502.

Dunis, C. L., Likothanassis, S. D., Karathanasopoulos, A. S., Sermpinis, G. S., & Theofilatos, K. A. (2013). A hybrid genetic algorithm–support vector machine approach in the task of forecasting and trading. *Journal of Asset Management*, *14*(1), 52-71.

Engadget. http://www.engadget.com/ (The website was last accessed on 2014-Feb-17th)

Fletcher, T., Hussain, Z., & Shawe-Taylor, J. (2010). Multiple Kernel Learning on the Limit Order Book. *Journal of Machine Learning Research-Proceedings Track*, *11*, 167-174.

Fletcher, T., Hussain, Z., & Shawe-Taylor, J. Multiple Kernel Learning on the Limit Order Book. *Journal of Machine Learning Research-Proceedings Track*, **2010**; *11*, 167-174.

Fletcher, T., Hussain, Z., Shawe-Taylor, J. (2010). Multiple kernel learning on the limit order book. *JMLR: Workshop and Conference Proceedings*, 11, 167–174.

Foresti, L., Tuia, D., Pozdnoukhov, A., Kanevski, M. (2009). Multiple Kernel Learning of Environmental Data. Case Study: Analysis and Mapping of Wind Fields. Artificial Neural Networks –ICANN, *Lecture Notes in Computer Science*, 5769, 933–943.

Frick, A., Herrmann, R., Kreidler, M., Narr, A., Seese, D. (1996). Genetic Based Trading Rules: A New Tool to Beat the Market With? *Proceedings of 6th International AFIR Colloquium*, 1/2, 997–1018.

Fuente, D., Garrido, A., Laviada, J., Gomez, A. (2006). Genetic algorithms to optimize the time to make stock market investment. In *GECCO 2006 Conference Proceedings*, 2, 1857–1858.

Gehrig, T., Menkhoff, L., 2006. Extended evidence on the use of technical analysis in foreign exchange, *International Journal of Finance and Economics*, 11(4), 327–338,

Gencay, R., Dacarogna, M., Olsen, R., Pictet, O. (2003). Foreign exchange trading models and market behavior. *Journal of Economic Dynamics and Control*, 27, 909–935.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley, Reading, MA.

Google Finance. http://finance.google.com/finance (The website was last accessed on 2014-Feb-17th)

Gruhl, D., Guha, R., Kumar, R., Novak, J., & Tomkins, A. (2005). The predictive power of online chatter. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 78-87). ACM.

Gülen, S. G. Efficiency in the crude oil futures market. *Journal of Energy Finance & Development*, **1998**; *3*(1), 13-21.

Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems*, *23*(8), 800-808.

Haidar, I., Kulkarni, S., & Pan, H. Forecasting model for crude oil prices based on artificial neural networks. In *Intelligent Sensors, Sensor Networks and Information Processing,* **2008**. *ISSNIP 2008. International Conference on* (pp. 103-108). IEEE.

Hann, T. H., & Steurer, E. (1996). Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. *Neurocomputing*, *10*(4), 323-339.

Hirabayashi, A., Aranha, C., & Iba, H. (2009). Optimization of the trading rule in foreign exchange using genetic algorithm. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (pp. 1529-1536). ACM.

Huntington, H. Oil price forecasting in the 1980s: what went wrong? *The Energy Journal*. **1994**, 15(2), 1-22.

Jammazi, R., & Aloui, C. Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modeling. *Energy Economics*, **2012**; *34*(3), 828-841.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning (ECML)*, 1398, 137–142.

Joutou, T., Yanai, K. (2009). A Food Image Recognition System with Multiple Kernel Learning. *Proceedings of International Conference on Image Processing*, 285–288.

Kodogiannis, V., Lolis, A. (2002). Forecasting financial time series using neural network and fuzzy system-based techniques, *Neural Computing and Applications*, 11, 90–102.

Koskela, T., Varsta, M., Heikkonen, J., Kaski, K. (1997). Time series prediction using recurrent SOM with local linear models. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 2, 60–68.

Kwapień, J., Gworek, S., Drożdż, S., Górski, A. (2009). Analysis of a network structure of the foreign currency exchange market. *Journal of Economic Interaction and Coordination*, 4 (1), 55–72

Kwok, J. T.-Y. (2000). The evidence framework applied to support vector machines. *IEEE Transactions on Neural*

Networks, 11 (5), 1162–1173.

Lee, M. (2003). Common trend and common currency: Australia and New Zealand. *International Journal of Business and Economics*, 2, 155–165.

Li, B., Xu, S., & Zhang, J. (2007). Enhancing clustering blog documents by utilizing author/reader comments. In *Proceedings of the 45th annual southeast regional conference* (pp. 94-99). ACM.

Luss, R., & d'Aspremont, A. (2012). Predicting abnormal returns from news using text classification. *Quantitative Finance*, DOI:10.1080/14697688.2012.672762

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

Mehdi, K., Mehdi, B. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing*, 11. 2664–2675.

Mizuno, T., Takayasu, H., Takayasu, M. (2006). Correlation networks among currencies. *Physica A*, 364, 336–342.

Mondria, J., Wu, T., & Zhang, Y. (2010). The determinants of international investment and attention allocation: Using internet search query data. *Journal of International Economics*, *82*(1), 85-95.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

Neely, C. J. (1998). Technical analysis and the profitability of US foreign exchange intervention. *Federal Reserve Bank of St. Louis Review*, *80*(July/August 1998).

Neely, C. J., Weller, P. A. (2003). Intraday technical trading in the foreign exchange market. *Journal of International Money and Finance*, 22, 223–237.

Olson, D. (2004). Have trading rule profits in the currency market declined over time? *Journal of Banking and Finance*, 28, 85–105.

Online Source 1. http://fxtrade.oanda.com/analysis/currency-correlation (last accessed on August 24th, 2012)

Online Source 2. http://forexautomaton.com/forex-correlation-analysis-reports-1h1m (last accessed on August 24th, 2012)

Online Source 3. http://fx.sauder.ubc.ca/CAD/analysis.html (last accessed on August 24th, 2012)

Online Source 4. http://www.federalreserve.gov/releases/h15/data.htm (last accessed on August 24th, 2012)

Pai, P.-F., Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega: The International Journal of Management Science*, 33 (6), 497–505.

Park, C-H and Irwin, S. H. (2004). The Profitability of Technical Analysis: A Review, AgMAS Project Research Report No. 2004-04

Park, C. H., & Irwin, S. H. A reality check on technical trading rule profits in the US futures markets. *Journal of Futures Markets*, **2010**; *30*(7), 633-659.

Pesaran, M. H., Allan T. (1995). Predictability of Stock Returns: Robustness and Economic Significance. *The Journal of Finance*, 50 (4), 1201–1228.

Pierdzioch, C., Rülke, J. C., & Stadtmann, G.. New evidence of anti-herding of oil-price forecasters. *Energy Economics*, **2010,** *32*(6), 1456-1459.

Pierdzioch, C., Rülke, J. C., & Stadtmann, G.. Oil price forecasting under asymmetric loss. *Applied Economics*, **2013**, *45*(17), 2371-2379.

Platt, J. C. Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods: Support vector machines, **1998**; *Cambridge, MA: MIT Press.*

Ravindran, R., Ganisen, S. and Roslan, M.R. (2008). Exchange Rate Forecasting. An Application of Higher Order ARIMA and GARCH models. *Annual London Conference on Money, Economy and Management.*

Riccardo, C., Charles, G., Dominique, G., Richard, P. (1997). Do technical trading rules generate profits? Conclusions from the intra-day foreign exchange market. *International Journal of Finance & Economics*, 2 (4), 267–280.

Schulmeister, S. (2007). The Profitability of Technical Stock Trading has Moved from Daily to Intraday Data. *WIFO working papers*.

Sermpinis, G., Jason, L., Andreas, K., Christian, L. D. (2012). Forecasting and trading the EUR/USD exchange rate with Gene Expression and Psi Sigma Neural Networks. *Expert Systems with Applications*, 39 (10), 8865–8877

Sharpe, W. F. (1994). The Sharpe Ratio. *Journal of Portfolio Management*, 21 (1), 49–58.

Shioda, K., Deng, S., & Sakurai, A. (2011). Prediction of Foreign Exchange Market States with Support Vector Machine. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on* (Vol. 1, pp. 327-332). IEEE.

Smith, G. P. (2012). Google Internet search activity and volatility prediction in the market for foreign currency. *Finance Research Letters*, *9*(2), 103-110.

Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., & Franc, V.. The SHOGUN machine learning toolbox. *The Journal of Machine Learning Research*, **2010**; *11*, 1799-1802.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *The Journal of Machine Learning Research*, *7*, 1531-1565.

Stawicki, S. P. (2007). Application of financial analysis techniques to vital sign data: a novel method of trend interpretation in the intensive care unit. *OPUS 12 Scientist*, 1 (1), 14–16.

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

Vapnik, V. The Nature of Statistical Learning Theory (Book), Springer, NY, USA, 1995

Verleger, P. K. *Adjusting to volatile energy prices* (Vol. 39). **1994**; Peterson Institute.

Wang, J., & Wan, W. Application of desirability function based on neural network for optimizing biohydrogen production process. *International Journal of Hydrogen Energy*, **2009**; *34*(3), 1253-1259.

Wang, T., & Yang, J. Nonlinearity and intraday efficiency tests on energy futures markets. Energy Economics, **2010**; 32(2), 496-503.

Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*. Trend Research, first edition.

Williams, L. (2005). *Trade Stocks Commodities with the Insiders: Secrets of the COT Report*. Wiley.

Wong, B. K., Selvi, Y. (1998). Neural network applications in finance: A review and analysis of literatures (1990-1996), *Information & Management*, 34, 129–139.

Wong, W. K., Manzur, M., and Chew, B. K. How rewarding is technical analysis? Evidence from Singapore stock market. *Applied Financial Economics*, **2003**; 13(7), 543-551

Wu, M.-Ch. (2007). Phase correlation of foreign exchange time series, *Physica A: Statistical Mechanics and its Applications*, 375 (2), 633–642.

Xiao-lin, Z. and Hai-wei, W. Crude Oil Prices Predictive Model Based on Support Vector Machine and Particle Swarm Optimization. In Software Engineering and Knowledge Engineering: Theory and Practice, **2012**; (pp. 645-650). *Springer Berlin Heidelberg.*

Xie,W., Yu, L., Xu, S.Y.,Wang, S.Y. A new method for crude oil price forecasting based on support vector machines. *Lecture Notes in Computer Science,* **2006**; 3994, 441–451.

Yeh, C.-Y., Huang, C.-W., Lee, S.-J. (2011). A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Systems with Applications*, 38 (3), 2177–2186.

Yu, L., Wang, S., & Lai, K. K. Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. Energy Economics, **2008**; 30(5), 2623-2635.

Zhao, Q.-Y., Zhao, X.-Y., Duan, F. (2009). Prediction Model of Stock Prices Based on Correlative Analysis and Neural Networks. *Proceedings of the 2009 Second International Conference on Information and Computing Science*, 3, 189–192.

de la Fuente, D., Garrido, A., Laviada, J., & Gómez, A. (2006). Genetic algorithms to optimise the time to make stock market investment. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 1857-1858). ACM.

Publication list

International Journal

[1] **Shangkun Deng**, Kazuki Yoshiyama, Takashi Mitsubuchi, Akito Sakurai. Hybrid Method of Multiple Kernel Learning and Genetic Algorithm for Forecasting Short-Term Foreign Exchange Rates. *Computational Economics*, 2013, 1-41. DOI: 10.1007/s10614-013-9407-6

[2] **Shangkun Deng**, Mitsubuchi, T., & Sakurai, A. Stock Price Change Rate Prediction by Utilizing Social Network Activities. *The Scientific World Journal*, 2014. (Published online)

[3] **Shangkun Deng**, Akito Sakurai. Crude oil spot price forecasting based on multiple crude oil markets and time frames. *Energies*, 7(5), pp. 2761-2779.

[4] **Shangkun Deng**, Akito Sakurai. Integrated Model of Multiple Kernel Learning and Differential Evolution for EUR/USD trading. *The Scientific World Journal*, 2014. (Published online)

International Conference

[1] **Shangkun Deng**, Akito Sakurai. Combining multiple kernel learning and genetic algorithm for forecasting short time foreign exchange rate. *Proceedings of the IASTED International Conference Artificial Intelligence and Applications* (AIA 2011) pp. 200-209

[2] **Shangkun Deng**, Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A.. Combining technical analysis with sentiment analysis for stock price prediction. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on* (pp. 800-807). IEEE.

[3] **Shangkun Deng**, Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A. Multiple kernel learning on time series data and social networks for stock price prediction. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on* (Vol. 2, pp. 228-234). IEEE.

[4] **Shangkun Deng**, Sun Yizhou, and Sakurai Akito. "Robustness Test of Genetic Algorithm on Generating Rules for Currency Trading." *Procedia Computer Science* 13 (2012): 171-183.

[5] **Shangkun Deng**, and Akito Sakurai. "Foreign Exchange Trading Rules Using a Single Technical Indicator from Multiple Time frames." *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. (pp. 207-212) IEEE, 2013.

[6] Shioda Kei, **Shangkun Deng**, and Akito Sakurai. "Prediction of Foreign Exchange Market States with Support Vector Machine." *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*. Vol. 1. (pp.327-332), IEEE, 2011.

[7] **Shangkun Deng**, Akito Sakurai. Short-term Foreign Exchange Trading Based on Support/Resistance Level of Ichimoku Kinkohyo (accepted by ISEEE)