

# Supervised Support Vector Machine in Predicting Foreign Exchange Trading

**Thuy Nguyen Thi Thu**

ThuongMai University, HaNoi, Vietnam  
E-mail: NguyenTthuthuy@gmail.com, NguyenTthuthuy@tmu.edu.vn

**Vuong Dang Xuan**

SonLa Gifted High School, Sonla, Vietnam

Received: 23 March 2018; Accepted: 23 May 2018; Published: 08 September 2018

**Abstract**—Trends of currency rates can be predicted with supporting from supervised machine learning in the transaction systems such as support vector machine (SVM). By assumption of binary classification problems, the SVM can predict foreign exchange transaction as uptrend or downtrend. The prediction is performed basing on collected historical data. Alternative SVM models have been used to vote the best one, which is deployed detail in Expert Advisor (Robotics). This is to show that support vector machine models might help investors to automatically make transaction decisions of Bid/Ask in Foreign Exchange Market. For comparison, the transactions without using SVM model also are performed. The results of experimental transactions show the advantages of using SVM model compared to the transactions without using SVM model.

**Index Terms**—Foreign Exchange Market, Exchange Rates, Machine Learning, Support Vector Machine (SVM), Prediction.

## I. INTRODUCTION

Machine learning methods using computer science play important role in many areas such as business decision making, handling or monitoring factory chain, predicting or forecasting stock, and so on. The main task in machine learning is to teach computer systems abilities of learning to classify or cluster from given data set. In business area, machine learning especially supervised learning methods can effectively help users to analyze the big data collected from alternative resources to extract useful knowledge. For example, given currencies exchanging rate data from online transaction, supervised learning can predict the rise or fall in short or long period based on learning from historical data.

Supervised learning is one of machine learning methods to automatically build a predictive function  $f: x \rightarrow y$  that maps  $x$  to a prediction  $y$  [13]. This means the given data set has a structure of  $\{x, y\}$  where  $x$  can be seen as the raw data and  $y$  will be the target label. The task of supervised learning technique is to predict the

expected output based on the historical data.

Almost supervised learning techniques are used for classification and regression tasks. It is also so-called Intelligent Systems as it has been developed based on Artificial Intelligence system such as Logical/Symbolic techniques, Perceptron-based techniques and Statistics (Bayesian Networks, Instance-based techniques) [22].

The task is called classification if the target variable ( $y$ ) is categorical (nominal or discrete) and called regression when  $y$  is continuous (real-valued) [13]. For simple, many classification problems are treated as a binary classification task. On the other words, the output can be seen as binary attribute.

Support Vector Machines (SVMs) are the supervised machine learning techniques [26]. SVMs revolve around the notion of a “margin”—either side of a hyperplane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyper-plane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error [22]. Therefore, SVMs can be used to solve binary problems such as predicting Foreign Exchange (Forex) trend (Uptrend or Downtrend) based on historical data of “Open”, “Close”, “Low” and “High” in alternative time periods.

This paper has been organized in 5 sections. Section 1 shows the introduction of machine learning technique and Forex problem. The related works can be seen in section 2. Detail support vector machine (SVM) method and Forex problem are introduced in section 3. The section 3 also shows the detail deployment of Expert Advisor (Robotics) with using SVM model for FoRex transactions with demo account. This is considered as a novel contribution for the paper as not many previous researches have done before. Section 4 shows the experiments of using support vector machine model in predicting the Forex trend and their results. The results of using Robotics with and without SVM for FoRex transactions also are represented in this section in order to show the advantage of using SVM model. The conclusion is represented in section 5.

## II. RELATED WORKS

The Forex prediction problems have usually be seen as the binary classification ones because the expected output is one of either rise or fall foreign currencies rate trends. Box and Jenkins' AutoRegressive Integrated Moving Average (ARIMA) technique has been widely used for time series forecasting. However, ARIMA is a general unvaried model and it is developed based on the assumption that the time series being forecasted are linear and stationary [1]. According to [1], exchange rates prediction is one of the most challenging applications of modern time series forecasting. This is because of the inherently noisy or non-stationary rates [9]. In this case, the historical data is the major factor during the prediction process. The researches in [1, 8- 9] have used Radial Basic Function (RBF) and Multilayer Perceptron Network (MLP) models for the FoRex classification whereas the one in [18] used gene models. The research in [1] has used a data set of 1825 instances including a training of 1600 and testing set of 225 observations. It seems that there was a small volume of historical data set in order to train the model. The ANN model included multilayer perceptron network with a structure of 3 input nodes (price, weight, and fluctuation of prices), 2 hidden nodes and 1 output node (changed or unchanged). The model weights were calculated by gene algorithm with 8 genes in alternative chromosomes. The disadvantage of this model is simple as well as using of small data set for the experiments so that the quick regression of model might be mistaken. The mining data in Rumania stock using neural network models for price prediction also can be seen in [6]. The neural network such as MLP (multilayer perceptron) is not only successful in classifying of electroencephalogram (EEG) [12, 21] but also it has been used to predict the future price of stock code in financial field. The results have shown the advantages of using supervised learning techniques in predicting classification problems. Other research on predicting trends in stock market can be seen in [3, 5, 23, 28-29]. Their researches have reported using artificial neural networks or decision tree models. However, many of these papers reported that they are able to work in real-time settings and according to [19], there remains a lack of published research using high frequency data. For example, in [23], neural network results shown quite good accuracy (less than 5%) in short-term prediction of exchange rates (EUR/USD, GBP/USD and USD/JPY) in daily, monthly and quarterly. However, the size for the training, testing sets were very small (less than 83 observations in total). Therefore, the results might not be the general representative solution for FoRex prediction problems or the quick regression might be taken. Moreover, the research in [23] used one-step mode in neural network method. This simple mode might not deal with complicated FoRex problems such as many input variables, big size of transactions etc.

Finding solution in financial problem also has been interested in [11]. In there, the stochastic differential equation, which is usually used determining boundary in

financial problems such as asset pricing or Asian option to buy the asset at the future time, etc., has been used with Euler-Maruyama method. However, the assumption of linear equation seem to be hard to apply in non-linear hyperplan (non-linear dataset) in FoRex predicting problems.

Many publications have shown that SVMs were extended to solve nonlinear regression estimation problems and have been successfully used to solve forecasting or classification problems in various fields [16, 24, 25]. However, according to [14], there is not many studies for the application of SVM in financial time-series forecasting, although SVM has some advantages [7, 17]. Recently, the research in [2] has shown an integration of Genetic Algorithms and Support Vector Machine to trade in the FoRex market. In there, SVM is used to identify and classify the market and Genetic Algorithm is used to optimize the trading rules which are based on the best SVM classified indicators.

For above reasons, in this paper, SVM model is chosen to predict the rise or fall of currencies rates in FoRex market. Moreover, it is installed into Expert Advisor (Robotics) for making transactions. Another reason of choosing SVM is because of no parameters to tune except the upper bound C [10], and SVM avoids over-fitting problems like other Artificial Neural Networks models [13].

## III. SUPERVISED SUPPORT VECTOR MACHINES AND FOREIGN EXCHANGE PREDICTION

### A. Supervised Support Vector Machine

Support Vector Machines (SVMs) are so-called as linear or non-linear SVMs depending on linear or non-linear hyperplane lines separating the training data set. In linear hyperplane, the formula of linear line as follows:

$$H: w \cdot x + b = 0.$$

Where  $w = (w_1, w_2, \dots, w_d) \in \mathbb{R}^d$  and  $x = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$  then  $w \cdot x = \sum_{i=1}^d w_i \cdot x_i$ .

Assume that  $y_i \in \{+1, -1\}$ ,  $i = \overline{1, n}$ .  $y_i$  is labelled output class. Then outputs will be separated by:

$$y_i = \begin{cases} 1 & \text{if } w \cdot x_i + b \geq 0 \\ -1 & \text{if } w \cdot x_i + b < 0 \end{cases}$$

The hyperplane H has maximum margins to classes. That means SVMs need to build two others lines as:  $H_1: w \cdot x + b = 1$ ; and  $H_2: w \cdot x + b = -1$  where there is no existing  $x_i$  in  $H_1$  and  $H_2$ , and the margin between in  $H_1$  and  $H_2$  is maximum. If the margin is maximum, there are some data points having positive labels and negative labels in the line of  $H_1$ , and  $H_2$  respectively. These data points can be called as support vectors in hyperplane of H which can separates the data into two regions of positive and negative labels.

The distance of x in  $H_1$  and  $H_2$  to H can be calculated as:

$$\frac{|wx+b\pm 1|}{\|w\|} = \frac{1}{\|w\|} \quad (1)$$

Therefore, the margin can be calculated as:  $\frac{2}{\|w\|}$ , where

$$\|w\| = \sqrt{w^2} = \sqrt{w_1^2 + w_2^2 + \dots + w_d^2}$$

On the other words, the maximum margin will be referred to minimum of  $\frac{w^2}{2}$  (2)

with the subject to:  $y_i(w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n$ .  
To sum up, the problem is referred to

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ where } y_i \cdot (w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n \quad (3)$$

This is a quadratic convex optimization problem.

The problem in (3) can be replaced as dual problem as follows:

By using Lagrange factors  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \geq 0$ , we have:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^2 - \sum_{i=1}^n \alpha_i \cdot y_i (w \cdot x_i + b) + \sum_{i=1}^n \alpha_i \cdot \max_{\alpha} \mathcal{L}(w, b, \alpha) \quad (4)$$

With constrains as:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^n \alpha_i \cdot y_i \cdot x_i = 0 \quad (5)$$

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{i=1}^n \alpha_i \cdot y_i = 0 \quad (6)$$

And  $\alpha \geq 0$ .

From (5) and (6) we have:

$$w = \sum_{i=1}^n \alpha_i \cdot y_i \cdot x_i \quad \text{and} \quad \sum_{i=1}^n \alpha_i \cdot y_i = 0.$$

Therefore, the expression of  $\mathcal{L}(w, b, \alpha)$  will be:

$$\mathcal{L}_D = \mathcal{L}(w, b, \alpha) \equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot x_i \cdot x_j \quad (7)$$

With  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$  and given  $b$  we have:

$$w = \sum_{i=1}^n \alpha_i \cdot y_i \cdot x_i.$$

For each  $x \in \mathbb{R}^d$  will be classified by:

$$\begin{aligned} f(x) &= \text{sign}(w \cdot x + b) \\ &= \text{sign} \left( \left( \sum_{i=1}^n \alpha_i \cdot y_i \cdot x_i \right) \cdot x + b \right) \\ &= \text{sign} \left( \sum_{i=1}^n \alpha_i \cdot y_i \cdot (x_i \cdot x) + b \right) \end{aligned}$$

Where:  $\text{sign}(x) = \begin{cases} 1 & \text{v} \acute{o}i \ x \geq 0 \\ -1 & \text{v} \acute{o}i \ x < 0 \end{cases}$

For non- linear hyperplane in non-linear training data set, a solution for the inseparability problem is mapping data onto a higher dimensional space (the transformed feature space), where a linear hyperplane can be used there.

$$\begin{aligned} \Phi: \mathbb{R}^d &\rightarrow \mathbb{R}^{d'} \\ x &\mapsto \Phi(x) \end{aligned} \quad (3)$$

The Equation of (7) will be:

$$\mathcal{L}_D \equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot \Phi(x_i) \cdot \Phi(x_j) \quad (8)$$

Where  $\Phi(x_i) \cdot \Phi(x_j) = k(x_i, x_j)$  is so-called as ‘kernel function’ [4], this kernel function is used to map new points into the feature space for classification whenever a hyperplane has been created.

In some cases, SVM may not be able to find any separating hyperplane as if the data contains misclassified instances. In this situation, SVM used a soft margin that accepts some misclassifications of the training instances [15].

In other words, the D set might contain noisy data. At this time, the constraint of no existing data point between  $H_1$  and  $H_2$  is not guaranteed. However, we can put positive variables of  $\xi_i$  to change the conditions of  $H_1$  and  $H_2$  as:

$$\begin{aligned} wx_i + b &\geq 1 - \xi_i & \text{where } y_i &= +1 \\ wx_i + b &\leq -1 + \xi_i & \text{v} \text{where } y_i &= -1 \\ \xi_i &\geq 0, & i &= 1, 2, \dots, n \end{aligned}$$

We define a coefficient of fine C (positive value) to the objective function for the noisy data points as:

$$\min_{w,b,\xi} \left( \frac{1}{2} w^2 + C \cdot \left( \sum_{i=1}^n \xi_i \right)^m \right) \quad (9)$$

Basically,  $m$  equals as 1, therefore, the optimization problem can be seen as:

$$\min_{w,b,\xi} \left( \frac{1}{2} w^2 + C \cdot \sum_{i=1}^n \xi_i \right) \quad (10)$$

With the constraints:

$$\begin{aligned} y_i(w \cdot x_i + b) + \xi_i - 1 &\geq 0, \quad i = 1, 2, \dots, n \\ \xi_i &\geq 0, \quad i = 1, 2, \dots, n \end{aligned}$$

In Dual problem there are no variables  $\xi_i$ , the constraints  $\alpha_i$  have a super limitation of  $C$ . We have:

$$\max_{\alpha} \mathcal{L}_D \equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (11)$$

With constrains:

$$0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0$$

*Karush-Kuhn-Tucker (KKT) conditions* [27]

The condition of existing optimum values for optimization problem or dual problem as

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (12a)$$

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{i=1}^n \alpha_i y_i = 0 \quad (12b)$$

$$\frac{\partial \mathcal{L}}{\partial \xi} = 0 \quad (12c)$$

$$\alpha_i (y_i (w \cdot x_i + b) + \xi_i - 1) = 0, \quad i = 1, 2, \dots, n \quad (12d)$$

$$\beta_i \xi_i = 0, \quad i = 1, 2, \dots, n \quad (12e)$$

Depending on  $\alpha_i \in [0, C]$ , we have:

1. If  $\alpha_i = 0$  then  $\beta_i = C - \alpha_i = C > 0$ .

From (12e) we have  $\xi_i = 0$ .

Therefore:

$$y_i (w \cdot x_i + b) - 1 \geq 0.$$

2. If  $0 < \alpha_i < C$  then from (12d) we have:

$$y_i (w \cdot x_i + b) + \xi_i - 1 = 0.$$

Moreover,  $\beta_i = C - \alpha_i > 0$  so  $\xi_i = 0$ .

Therefore:  $y_i (w \cdot x_i + b) - 1 = 0$ .

3. If  $\alpha_i = C$  then from (12d) we have:  $y_i (w \cdot x_i + b) + \xi_i - 1 = 0$ .

Moreover,  $\beta_i = C - \alpha_i = 0$  so  $\xi_i \geq 0$ .

Therefore:  $y_i (w \cdot x_i + b) - 1 \leq 0$ .

All above cases will be used to recalculate the Lagrange values during training models process.

*Training SVM models algorithm*

The popular algorithm used for SVM models is Sequential minimal optimization – SMO [20]. The SMO solved the dual optimization problem as:

$$\max_{\alpha} \mathcal{L}_D \equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j.$$

with conditions  $0 \leq \alpha_i \leq C, \forall i$  and  $\sum_{i=1}^n \alpha_i y_i = 0$ .

**SMO algorithm**

**Optimality Algorithm**

**Input:** Training data, parameter settings

**Output:**  $\alpha_i$  as optimal

Optimality = 0, and  $\alpha_i = 0$

Repeat the following steps until reaching optimality

**Step 1:** Choose an adequate working set

**Step 2:** Calculate new  $\alpha_i$  corresponding to QP sub-problem

**Step 3:** Update  $\alpha_i$

**Step 4:** Check a stopping criterion for optimality.

Assume that  $E(x, y)$  is prediction error for each pair of  $(x, y)$ . Therefore,

$$E(x, y) = \sum_{i=1}^n \alpha_i y_i x_i \cdot x + b - y.$$

According to [20], the selection of  $\alpha_i, \alpha_j$  will be performed in two heuristic loops so that  $|E_i - E_j|$  will be maximum. The Lagrange multipliers  $\alpha_i$  are used in the process to select a  $\alpha_i$  so that  $0 < \alpha_i < C$  and it is not satisfied Karush-Kuhn-Tucker conditions [27], and  $\alpha_j$  ( $0 < \alpha_j < C$ ) so that  $|E_i - E_j|$  will be maximum.

*B. Foreign Exchange Prediction*

The Forex market can be seen as a special one in financial markets with its characteristics of high benefits, maintaining capital in the case of inflation, and its transaction can be performed all the time at any location in the world. The investors can buy/sell the pairs of money such as EURUSD (Euro vs US Dollar), USDJPY (US Dollar vs Japanese Yen), GBPUSD (Great Britain Pound vs US Dollar) etc. to get benefits from different exchange rates between these pairs. In general, the following parameters define the trend of up or down of the rates:

- Target Profit (TP): is positive to indicate the target profit.
- Stop Loss (SL): is positive number to show that the loss is acceptable.

- Time Trend Detection (T): is positive to indicate the forward time starting from current point.

Therefore, given the current rate of (P), the outputs of prediction as:

- The trend of exchange rate is up if the rate is increased in term of (P+TP) instead of decreasing an amount of (P-SL) in the duration of T.
- The trend of exchange rate is down if the rate is decreased in term of (P-TP) instead of increasing an amount of P+SL in the duration of T.

The parameters of SL, TP are used with unit of “PIP” (to measure the profit or loss in the market).

#### Frame work of FOREX transaction

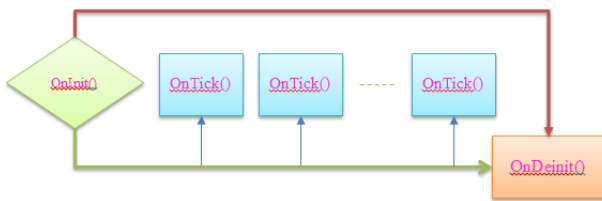


Fig.1. Frame works for Robotics in FoRex Transaction.

```

int OnInit()
{
  //Robotics initializations such as
  checking market information (minLot,
  maxLot, etc.), Pip,StopLoss, TakeProfit,
  Using SVM model or not, etc.

  //return INIT_FAILED or INIT_SUCCEEDED
  return (INIT_SUCCEEDED);
}

void OnTick()
{
  //Manage the Open, Close current demands
  ManagerOpenedPosition();
  //Analyse the history data to predict of
  opening or closing
  CheckSignal(...);
  ProcessSignal(...);
  //Repeat until there is no change in the
  active FoRex pair.
}

void OnDeinit(const int reason)
{
  //Release memory for Robotics
}
  
```

Fig.2. Initial Setting for Robotics Transaction.

The robotics performs transactions by using OnInit() function. If any change of current money pair has happened, OnTick() function will be performed to manage the opened or close demands as well as analyse the history of demands. If SVM model is used, the SVM models with alternative kernel functions and their parameters such as number of support vectors, C,  $\gamma$ , etc. will be used (see Fig. 3a, Fig. 3b).

The Ontick () function will be used iteratively until end of user transaction session time. At this time, the OnDeinit() will be used to release the robotics.

```

double SVM::KernelFunc(int idx, double
&inVector[])
{
  double s = 0.0;
  int d = MathMin(nDim,
  ArraySize(inVector));
  int i;

  switch (eKernel)
  {
    // Gaussian RBF
    case klGsRBF:
      //calculate SupportVectors
      for (i = 0; i < d; i++)
        s +=(SupportVectors[idx][i]-
        inVector[i])*(SupportVectors[idx][i]-
        inVector[i]);
      return MathExp(-gamma * s);
    //Polynomial
    case klPolynomial:
      // calculate SupportVectors
      for (i = 0; i < d; i++)
        s += SupportVectors[idx][i] *
        inVector[i];
      return MathExp(MathLog(coeflin * s
      + coefconst) * poly_Degree);
    //Sigmoid Tanhyperbolic
    case klTanh:
      // calculate SupportVectors
      for (i = 0; i < d; i++)
        s += SupportVectors[idx][i] *
        inVector[i];
      return Tanh(coeflin * s +
      coefconst);
  }
}
  
```

Fig.3a. SVM code installed in Robotics transaction.

```

double SVM::RealPredict(double &inVector[])
{
  double out = 0;
  int i;
  //Linear model
  if (eKernel == klLinear)
  {
    int d = MathMin(nDim,
    ArraySize(inVector));
    //Calculate w * inVector
    for (i = 0; i < d; i++)
      out += w[i] * inVector[i];
  }
  //Nonlinear model
  else
  {
    for (i = 0; i < nSupportVector; i++)
      out += alpha[i] * outs[i] *
      KernelFunc(i, inVector);
  }

  return (out - b);
}
  
```

Fig.3b. SVM code installed in Robotics transaction.

## IV. EXPERIMENTS

## A. Data Configuration and Model preparation

## Data

The FoRex data is collected by using MetaTrader 4 [30] to download from 01/01/2013 to 30/09/2016. In this paper, the pair of EUR/USD has been selected as it is very popular one in the market. The training data set D included the instances in a period of 01/01/2013 to 31/12/2015, and the test data set D' contained the instances in a period 1/01/2016 to 30/09/2016.

## Features

The selection of experimental features is effected to the model results. In this paper, the features are selected basing on the objectives of profits about 10 – 15 Pips for each transaction. The observed exchange rates are the "Open", "Close", "Low" and "High" for the candles in the window time of M1 (1 minute), M5 (5 minutes), M15 (15 minutes), H1 (1 hour) and D1 (1 day). The features are also based on the RSI (Relative Strength Index), MA (Moving Average), Bollinger (Bollinger Bands), and Custom Indicators. The features are selected in the given data set can be seen in Table 1.

Table 1. 137 Selected Feature

Feature#	Data	N0	Note
#1-16	O, H, L, C on Timeframe M1	16	4 values x 4 candlesticks
#17-32	O, H, L, C on Timeframe M5	16	4 values x 4 candlesticks
#33-44	O, H, L, C on Timeframe M15	12	4 values x 3 candlesticks
#45-56	O, H, L, C on Timeframe H1	12	4 values x 3 candlesticks
#57-72	O, H, L, C on Timeframe D1	16	4 values x 4 candlesticks
#73-77	Time data	5	Data Attributes
#78-81	RSI(7) on Timeframe M5, M15	4	1 values x 4 Timeframe
#82-85	RSI(14) on Timeframe M5, M15	4	1 values x 4 Timeframe
#86-101	MA(9), MA(12), MA(100), MA(200)	16	4 values x 4 Timeframe
#102-113	Custom Indicator	12	4 PAX + 8 MKC
#114-119	Bollinger Bands	6	3 Lines x 2 Timeframe
#120-125	Average True Range	6	2 values x 3 Timeframe
#126-137	Highest, Lowest M1, M5	12	4 values x 3 Timeframe

## Model configurations

The machine learning technique in the experiments is SVM model. Its model is used with alternative of Gaussian RBF, and Polynomial kernel functions. In detail, the model parameters are:

- RBF:  $k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma}} = e^{-\gamma\|x-y\|^2}$  with  $\gamma \in [0; 5]$
- Polynomial:  $k(x,y) = (x \cdot y + \theta)^d$ ,  $\theta \in \mathbb{R}$ ,  $d \in \mathbb{N}^*$  with  $d = 2,3,4$  and  $\theta \in [0; 1]$ .

- The parameter of C has been used with  $C \in [1; 10]$ . According [7] the upper bound C and the kernel parameter in kernel functions play an important role in the performance of SVMs. Therefore, after many performed experiments, the parameters have been used are follows as if they produce acceptable results.

- RBF:  $\gamma \in [1; 2]$ .
- Polynomial:  $d = 3$  and  $\theta = 1$ .
- The parameter of C  $\in [1; 2]$

Table 2. Experimental Model Configurations

Models	Poly1	Poly2	Poly3	GsRBF1	GsRBF2	GSRBF3
C	1.0	1.0	1.0	2.0	1.0	2.0
Kernel	Poly	Poly	Poly	RBF	RBF	RBF
Parameters	Power 2	Power 3	Power 3	$\gamma=5.0$	$\gamma=4.0$	$\gamma=5.0$
No of vectors	3000	3000	2000	3000	4500	5000

## B. Results

The data is used with the cross validation method. The D data set is divided in to  $D_{pos}$  (positive output); and  $D_{neg}$  (negative output). They are split to  $D_{pos}^1, D_{pos}^2, \dots, D_{pos}^k$  and  $D_{neg}^1, D_{neg}^2, \dots, D_{neg}^k$  sub-sets (each sub-sets has  $\frac{1}{k} |D_{pos/neg}|$  samples. D\_Test will be each

sub-set and k-1 remaining sub-sets will be used for D\_Train. In this paper experiment, k is chosen of 5. The results can be seen in Table 3 and Table 4 for 6 above models. The Accuracy Rate, Precision of Positive, Negative, Micro-Average and Macro-Average results are used to compare the performance between alternative models.

Table 3. RBF Configuration Models Results

Kernel: Gaussian RBF Features: 137		GsRBF1		GsRBF2		GSRBF3	
		D_Train	D_Test	D_Train	D_Test	D_Train	D_Test
<b>Accurate</b>		82.41%	58.11%	86.08%	58.57%	87.28%	58.36%
<b>Precisio-n</b>	<b>Positive</b>	81.87%	57.61%	85.67%	57.78%	86.83%	57.59%
	<b>Negative</b>	82.96%	58.57%	86.50%	59.38%	87.75%	59.13%
	<b>Micro-Average</b>	82.41%	58.11%	86.08%	58.57%	87.28%	58.36%
	<b>Macro-Average</b>	82.42%	58.09%	86.09%	58.58%	87.29%	58.36%

Table 4. Poly Configuration Models Results

Kernel: Polynomial Features: 137		Poly1		Poly2		Poly3	
		D_Train	D_Test	D_Train	D_Test	D_Train	D_Test
<b>Accurate</b>		76.78%	74.20%	82.46%	74.67%	82.52%	74.60%
<b>Precisio-n</b>	<b>Positive</b>	76.38%	74.69%	82.98%	78.40%	83.07%	78.50%
	<b>Negative</b>	77.19%	73.76%	81.95%	71.95%	81.98%	71.80%
	<b>Micro-Average</b>	76.78%	74.20%	82.46%	74.67%	82.52%	74.60%
	<b>Macro-Average</b>	76.78%	74.22%	82.47%	75.18%	82.53%	75.15%

C. Discussions

Table 3 and Table 4 show the results of alternative models using alternative kernel functions of Gaussian RBF and Polynomial respectively. The results in Table 3 show that the performance of positive and negative prediction for Test sets has big gap compared to the Training sets (about 53% and 82.5 % respectively). In contrast, there is a little different between performance rates of the training and testing in Table 4.

The accuracy rates taken from two kernel functions of

Gaussian RBF and Polynomial are represented in the chart to show the high performance of the use of Polynomial one in general. In Polynomial functions group (Poly1, Poly2, Poly3), there is no difference of accuracy rates for the Test sets (about 74.5%) whereas Poly2 and Poly 3 models have better results for the training set (82.5%). These results show that the use of C=1, polynomial power of 3 and alternative support vectors (see in Table 2) does not change the experiment performance.

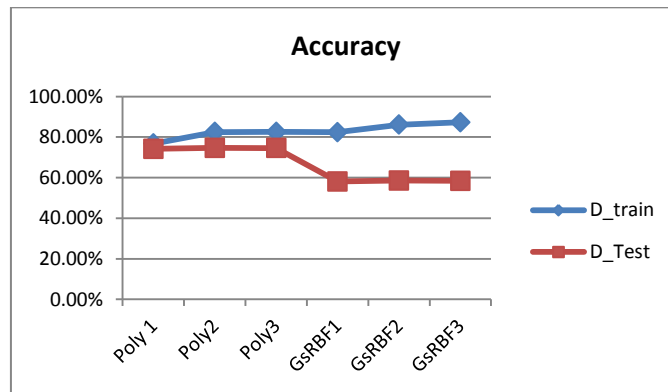


Fig.4. Accuracy rates of Experimental Model Configurations

D. Expert Advisor in FoRex Transactions

The Expert Advisor (EA) is a plug-in script implemented in MetaTrader 4 [30]. This script collects historical data, technical indicators as well as doing the transaction of buy or sell the pair of currencies. In this paper, the advisor used the trained SVM model taken from the previous stage. This is because the training might take much time to calculate whereas the decision of buying or selling is made in some seconds. Moreover, the changes in currency rates usually is happened in seconds or some blocks of second.

For comparison, the experiment of Robotics (expert advisor) without using support of SVM also has been used. The conditions for experiments as follows:

- Use Backtest function
- Broker: ICMarkets
- Currency pair: EUR/USD
- Time: 9 months
- Initial Margin: 1000USD
- Margin Level: 1:500
- Spread: 0.8 – 1.0 pip
- Commission: 7USD



The experiment results can be seen in Fig. 5 whereas the details of indicators' results of No\_SVM and SVM used for EA can be seen in Table 5.

Table 5 shows the advantage of using SVM in the transaction Robotics, in overall. For example, the Volume using SVM is a half of No\_SVM (96 and 189 respectively) whereas the Profit rate is triple (3.03) compared to 0.77. The Fair Drawdown in the use of SVM is a little bit high (34.69%). But it is still a half of the one using without SVM implementation.

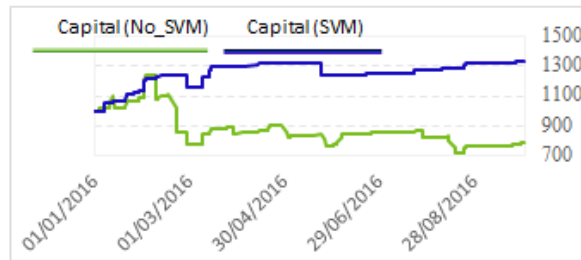


Fig.5. Experiment Results.

Table 5. Indicators' Results

Index	Indicators	No_SVM		SVM	
1	Net Profits		<b>-210.46</b>		<b>337.94</b>
2	Gross Profits		718.37		504.7
3	Gross Loss		-928.83		-166.77
4	Profit rate		<b>0.77</b>		<b>3.03</b>
5	Growth		<b>-21.0%</b>		<b>33.8%</b>
6	Average Growth (month)		<b>-2.59%</b>		<b>3.29%</b>
7	Strict Drawdown		371.72		151.5
8	Fair Drawdown	645.36	<b>50.67%</b>	450.67	<b>34.69%</b>
9	Volume		189		96
10	Profit Transactions	147	77.8%	85	88.5%
11	Loss Transactions	42	22.2%	11	11.5%
12	Ask Profit Transactions	111	91.0%	71	100.0%
13	Bid Profit Transactions	78	59.0%	25	56.0%

## V. CONCLUSIONS

The supervised machine learning technique such as support vector machine can help to solve the FoRex problem as it can be represented in binary classification task. The power of using artificial intelligence in learning models helps to deal with a huge, complex and time series data to extract knowledge for users' purpose. For example, in this paper, an amount of experimental data set of more than 13500 instances has been performed. By using Gaussian and Polynomial kernel functions, the alternative models with alternative parameters have been performed. The results of using Polynomial Kernel have shown the better results compared to the models using Gaussian in general. To show the advantage of using SVM in the actual

application, a best Poly model is chosen to deploy in Robotics for doing automatically transactions in FoRex market with demo account. Its results show the advantageous use compared to the traditional one (without use of SVM).

The Robotics transaction in this paper is used with demo account, therefore, further research will focus on continuing to use transactional Robotics in the real foreign exchange market

## REFERENCES

- [1] A.a. Baasher, and W.F. Mohamed, "FOREX Trend Classification using Machine Learning Techniques", *Arab Academy for Science and Technology*, 2010, pp. 41-47.
- [2] B. J. Almeida, R. F. Neves, N. Horta, "Combining Support Vector Machine with Genetic Algorithms to optimize investments in Forex markets with high leverage", *Applied Soft Computing*, Vol. 64, 2018, pp. 596-613.
- [3] C. F. Tsai, and Y.C. Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches", *Decision Support Systems*, 50(1), 2010, pp. 258 - 269.
- [4] C. Scholkopf, J.C. Burges, and A. J. Smola, "Advances in Kernel Methods", *MIT Press*, 1999.
- [5] C.L. Huang, and C.Y. Tsai, "A hybrid SOFM-SVR with a filter based feature selection for stock market forecasting", *Expert Systems with Applications*, 36(2, Part 1), 2010, pp: 1529 - 1539.
- [6] D. m. Nemeş, & A. Butoi, "Data Mining on Romanian Stock Market Using Neural Networks for Price Prediction", *Informatica Economică*, vol. 17, No. 3, 2013, pp. 125-136.
- [7] F.E.H. Tay, L. Cao, "Application of support vector machines in Financial time series forecasting", *Omega* 29, 2001, pp. 309-317.
- [8] G.E.P. Box, and G. M. Jenkins, "Time Series Analysis: Forecasting and Control", *Holden- Day, San Francisco*, CA. 1970.
- [9] G.J. Deboeck, "Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets", *New York Wiley*, 1994.
- [10] H. Drucker, D. Wu, V.N. Vapnik, "Support vector machines for spam categorization", *IEEE Trans. Neural Networks* 10 (5), 1999, pp.1048-1054.
- [11] H. R.Erfanian, M. Hajimohammadi, M. J. Abdi, "Using the Euler-Maruyama Method for Finding a Solution to Stochastic Financial Problems" *I.J. Intelligent Systems and Applications*, 2016, No. 6, pp.48-55.
- [12] H.A.R. Akkar, F. B. Ali Jasim, "Intelligent Training Algorithm for Artificial Neural Network EEG Classifications", *I.J. Intelligent Systems and Applications*, 2018, No.5, pp. 33-41.
- [13] I.H. Witten, E. Frank, MA. Hall. "Data mining: practical machine learning tools and techniques". *Morgan Kaufmann Publishers*, 2011.
- [14] K. Kim, "Financial time series forecasting using Support Vector Machines", *Neurocomputing* 55, 2003, pp. 307-319.
- [15] K. Veropoulos, C. Campbell, & N. Cristianini, "Controlling the Sensitivity of Support Vector Machines". *In Proceedings of the International Joint Conference on Artificial Intelligence*, 1999. (IJCAI99).
- [16] K.R. Müller, A. Smola, G. Räsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with



- support vector machines", *In proceedings of the IEEE workshop on neural networks for signal processing 7*, 1997, pp.511 – 519.
- [17] L. Liu, and W. Wang, "Exchange Rates Forecasting with Least Squares SVM", *International Conference on Computer Science and Software Engineering*, 2008.
- [18] M. Punniyamoorthy, & J.J. Thoppan, "ANN-GA based model for stock market surveillance", *Journal of Financial Crime*, vol. 20, No. 1, 2013, pp. 52-66.
- [19] M. Reboredo, J.J. Rubio, "Nonlinearity in forecasting of high-frequency stock returns", *Computational Economics*, 40(3), 2012, pp. 245– 264.
- [20] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", *Microsoft Research Technical Report MSR-TR-98-14*, 1998.
- [21] P.Thakar, A. Mehta, "A Unified Model of Clustering and Classification to Improve Students' Employability Prediction", *I.J. Intelligent Systems and Applications*, 2017, No.9, pp. 10-18.
- [22] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica*: 31, 2007, pp. 249-268.
- [23] S. Galeshchuk, "Neural networks performance in exchange rate prediction", *Neurocomputing*, 2016, No. 172, pp. 446–452.
- [24] S. Mukherjee, E.Osuna, F. Girosio, "Nonlinear prediction of chaotic time series using SVM", *In proceedings of the IEEE workshop on neural networks for signal processing 7*, 1997, pp. 511 – 519.
- [25] T. Scaria, T. Christopher, "Microarray Gene Retrieval System Based on LFDA and SVM", *I.J. Intelligent Systems and Applications*, 2018, No.1, pp. 9-15.
- [26] V. Vapnik, "The Nature of Statistical Learning Theory", *Springer Verlag*, 1995.
- [27] W. H. Kuhn, W.A. Tucker, "Nonlinear programming", *Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press*, 1951, pp. 481–492.
- [28] W. Huang, Y. Nakamori, and S. Y. Wang, "Forecasting foreign exchange rates with artificial neural networks, a review", *International Journal of Information Technology & Decision Making*. Vol 3, No 1, 2004, pp. 145-165.
- [29] W. Shen, X. Guo, C. Wu, and D. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm", *Knowledge-Based Systems*, 24(3), 2011, pp. 378–385.
- [30] Metatrader 4, (2016). Website: <http://www.metaquotes.net/>.

**Vuong Dang Xuan** has been graduated MSc. in Computer Science in Hanoi University of Education in 2016. He is now working in SonLa Gifted High School in Son La province. His interests include Machine Learning, Maths and Data Mining.

**How to cite this paper:** Thuy Nguyen Thi Thu, Vuong Dang Xuan, "Supervised Support Vector Machine in Predicting Foreign Exchange Trading", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.10, No.9, pp.48-56, 2018. DOI: 10.5815/ijisa.2018.09.06

## Authors' Profiles



**Thuy T. T. Nguyen** was born in 1973 in Bacgiang, Vietnam. She graduated university in 1993 in Math. In 1999, she received MSc degree in Information Technology in Hanoi National University. She received PhD in Computer Science at The University of Hull, UK in 2011 respectively. From 2001 afterward, she joined to Vietnam University of Commerce, as a lecturer. Her interested research includes data mining, neural network, supervised/unsupervised learning techniques, information systems especially to management information systems. Many her publications also are concentrated to these areas.