Master Thesis in Statistics and Data Mining

Forecasting exchange rates using machine learning models with time-varying volatility

Ankita Garg

Abstract

This thesis is focused on investigating the predictability of exchange rate returns on monthly and daily frequency using models that have been mostly developed in the machine learning field. The forecasting performance of these models will be compared to the Random Walk, which is the benchmark model for financial returns, and the popular autoregressive process. The machine learning models that will be used are Regression trees, Random Forests, Support Vector Regression (SVR), Least Absolute Shrinkage and Selection Operator (LASSO) and Bayesian Additive Regression trees (BART). A characterizing feature of financial returns data is the presence of volatility clustering, i.e. the tendency of persistent periods of low or high variance in the time series. This is in disagreement with the machine learning models which implicitly assume a constant variance. We therefore extend these models with the most widely used model for volatility clustering, the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) process. This allows us to jointly estimate the time varying variance and the parameters of the machine learning using an iterative procedure. These GARCH-extended machine learning models are then applied to make one-step-ahead prediction by recursive estimation that the parameters estimated by this model are also updated with the new information. In order to predict returns, information related to the economic variables and the lagged variable will be used. This study is repeated on three different exchange rate returns: EUR/SEK, EUR/USD and USD/SEK in order to obtain robust results. Our result shows that machine learning models are capable of forecasting exchange returns both on daily and monthly frequency. The results were mixed, however. Overall, it was GARCH-extended SVR that shows great potential for improving the predictive performance of the forecasting of exchange rate returns.

Acknowledgement

It gives me a great pleasure in acknowledging my supervisor Prof. Mattias Villani, Linköping University**,** for his guidance, patience and encouragement in completion of this thesis. His support from initial to final stage has enabled me to have better understanding of this topic.

I would also like to thank Marianna Blix Grimaldi, Sveriges Riksbank, for introducing such an interesting topic and providing me opportunity to do it. Her comments and suggestions were helpful in improving this study.

Table of Contents

1. Introduction

1.1 Background

Financial forecasting is truly a challenging task and remains a very active research area. Over the last decades, many efforts to forecast time series from different aspect have been studied. In the modeling of financial time series, key characteristics that need to be considered are: heteroscedasticity, noise and leptokurtosis (Cont, 2001). The heteroscedasticity manifests itself by sustained periods of varying volatility. This leads to gradual changes in the dependency between the input and the output variable in the modeling of financial time series. Thus, it becomes difficult for a single model to capture this dynamic relationship. The noise characteristic means that there is incomplete information about the behavior of the series and the information that is missing is known as noise. This could lead to underfitting and overfitting problem degrading the performance of the model. Leptokurtosis describes a series with high kurtosis, i.e. one associated with probability distribution function having high peaks and heavy tails simultaneously. Financial markets are considered to be unpredictable and thus any improvement in forecasting over the random walk process will be of great interest.

After two decades of work, if economists and researchers have reached anything regarding predictability of exchange rates that may be taken as consensus is that they are difficult to forecast. Many explanations have been given. One central conjecture is that exchange rate models fails to predict future currency changes because of the presence of time-varying parameters. Another is that while models have become increasingly complex there is still a lot of misunderstanding about the driving factors behind exchange rates which is always ever changing. The well known but still controversial statement is that *the market is unpredictable*. This is reflected by the Random Walk Hypothesis and Efficient market Hypothesis. According to the Random Walk Hypothesis market prices evolve according to random walk, i.e. changes in the series are random and unpredictable. The efficient market hypothesis states that financial markets are *informationally efficient.*

The famous study by Meese and Rogoff (1983) showed that it is difficult to outperform Random Walk model of the exchange rate model using macroeconomic fundamentals. However, recent findings by Anaraki (2007) clarified the role of fundamentals in explaining the exchange rate behavior. In this study we do not take a stance about which factors contribute towards giving better performance of a given exchange rate model. We take instead a pragmatic approach and we select a number of key driving factors that have been recognized by market participants. Notably, these factors are related to the perhaps most well-known exchange rate models, that is models that are based on the Interest Rate Parity, models that are based on the Purchasing Power Parity and models based on the Balassa-Samuelson effects (Hauner, Lee and Takizawa, 2011). A brief description of these models is given in Section 2.2. To these fundamental factors we add some *non-fundamental* factors such as the volatility index (VIX) and that are intended to capture movements in exchange rates that are more closely related to sentiment and risk appetite. *Fundamental* and *non–fundamentals* factors should then be viewed as complement and not as competing factors.

The aim of this study is to apply models from the machine learning literature to make inferences and predictions about future currency changes. The models used are Regression trees, Random Forest, Support Vector Regression (SVR), Bayesian Additive Regression trees (BART) and least absolute shrinkage and selection operator (LASSO). The advantages of these models are that they are flexible models that make relatively few assumptions about the functional form of the relationship between the exchange rate and its potential explanatory variables. A successful application of these tools is based upon the fact that the data is generated according to a distribution with constant variance. But financial time series are characterized by volatility clustering exhibiting periods of low and high volatility. We therefore extend the machine learning models with the GARCH model for the variance of the series, see section 3.9 for details. We propose an iterative approach to jointly estimate the machine learning models' parameters and the parameters of the GARCH process.

1.2 Objective

The aim of this study is to investigate the predictability of monthly and daily exchange returns using flexible machine learning models such as Regression trees, Random Forest, Support Vector Regression, Bayesian Additive Regression Trees (BART) and Least absolute shrinkage and selection operator (LASSO) using the economic and the lagged variables. The performance of these methods will be compared with benchmark models that are frequently used in the financial literature, such as the random walk and the autoregressive process.

2. Data

2.1 Data Source

The data sources for both monthly and daily returns are EcoWin Reuters and Bloomberg. EcoWin Reuters is a provider of worldwide financial and economical data. It is based in Gothenberg, Sweden. Bloomberg enables financial professionals to have access to real-time financial market data.

2.2 Raw Data

In this study, we will use data from January, 2000 to December, 2011 for forecasting both monthly and daily returns. We will be using three currency pairs: EUR/SEK, EUR/USD and USD/SEK. The screenshots of a part of data that will be used for predicting monthly and daily returns is shown in Figure 1 and Figure 2 respectively. It consists of domestic and foreign short-term and long-term interest rate, money supply, risk appetite measure, equity index, GDP change, inflation and confidence variable.

Date	Eur_SEK	EURIBOR	STIBOR	LIRswe2	LIRswe5	LIRger2	LIRger5	Risk_App_	STOXX	OMSX_Inc	USD_SEK	EUR_USD	swgdpaqq	EUGNEMU	SWCPMOI	ECCPEMUI	SWETSURV	GRZEEUEX	MoneySup	MoneySupply2
1/31/2000	8.59678	3.34314	3.70286	5.19905	5.6769	4.30933	4.98214	23.0333	5900.1	328.189	8.48504	1.01306	0.6	1.3	-0.8	0	115.2	89.9	4137.8	975471.1956
2/29/2000	8.50051	3.53676	4.10214	5.36429	5.75524	4.42029	5.08457	23.71	6370.18	368.886	8.63239	0.98355	0.6	1.3	0.5	0.4	115.4	86.7	4133.7	968803.4033
3/31/2000	8.37735	3.74704	4.15365	5.17348	5.43522	4.45178	4.96196	22.7183	6656.12	394.718	8.68116	0.96474	0.6	1.3	0.5	0.3	118	81.5	4143.9	986845.2212
4/28/2000	8.26021	3.92905	4.1294	5.045	5.3615	4.44975	4.89245	27.0985	6449.84	361.199	8.73706	0.94563	2	0.8	-0.1	0.1	116.2	80.6	4186	1002483.403
5/31/2000	8.24418	4.35039	4.08974	5.01783	5.34065	4.85357	5.16087	26.2904	6500.79	364.963	9.0757	0.90638	2	0.8	0.5	0.1	115.7	78.2	4177.6	983642.3655
6/30/2000	8.3102	4.50173	4.03268	4.94727	5.17023	4.90832	5.00064	21.54	6645.97	360.223	8.74437	0.95054	2	0.8	0	0.4	115	74.9	4186.4	980882.6214
7/31/2000	8.40878	4.5829	4.18767	5.0531	5.31857	5.07848	5.13905	19.89	6595.85	361.915	8.9502	0.93987	0.6	0.4	-0.5	0.1	115.3	70.1	4184.9	958386.0305
8/31/2000	8.39052	4.77709	4.11022	4.91565	5.25065	5.17148	5.1717	18.0887	6506.58	345.83	9.27376	0.90448	0.6	0.4	0.1	0.1	116.2	62.8	4176.9	986213.5967
9/29/2000	8.41534	4.85281	4.06838	4.68762	5.08333	5.11062	5.1239	19.5848	6484.99	350.347	9.66287	0.87207	0.6	0.4	0.7	0.5	113.6	37.2	4182.5	990896.075
10/31/2000	8.52474	5.04127	4.02955	4.56386	5.01341	5.03564	5.05018	25.2	6182.61	317.425	9.9812	0.85425	2	0.6	0.2	0	117.1	17.6	4187.3	986960.695
11/30/2000	8.62938	5.09195	4.02573	4.505	4.89909	4.99141	5.01814	26.4432	6278.16	306.646	10.0831	0.85526	2	0.6	0.1	0.2	115.4	6.1	4210.8	998470.5791
12/29/2000	8.68405	4.93343	4.14162	4.37857	4.59143	4.65205	4.64986	26.579	6050.74	295.221	9.6394	0.89973	2	0.6	-0.1	0.4	111	-0.3	4299.6	992458.7895
1/31/2001	8.8921	4.77439	4.14348	4.21609	4.55609	4.40535	4.49696	24.987	5979.16	292.048	9.46988	0.93805	-0.4	0.9	-0.3	-0.5	104.2	-4.6	4348.6	1006137.214
2/28/2001	8.9736	4.7558	4.12235	4.145	4.50525	4.44745	4.53865	23.347	5727.04	280.227	9.74401	0.92087	-0.4	0.9	0.4	0.3	102.5	-7	4355.6	996586.4535

Figure 1. A subset of the monthly dataset

Date	Eur_SEK	EURIBOR	STIBOR	LIRswe2	LIRswe5	LIRger2	LIRger5	Risk_App_	STOXX	OMSX_Inc	USD_SEK	EUR_USD	swgdpaqq	EUGNEMU	SWCPMOI	ECCPEMUI	SWETSURV	GRZEEUEX	MoneySup	MoneySupply2
1/3/2000	8.5832	3.338	3.633	5.18	5.6	4.343	4.895	24.21	6067.89	328.38	8.3643	1.00776	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/4/2000	8.6265	3.343	3.667	5.195	5.65	4.327	4.926	27.01	5828.4	319.81	8.3685	1.03189	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/5/2000	8.6305	3.341	3.65	5.19	5.645	4.306	4.918	26.41	5683.15	307.09	8.3648	1.03606	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/6/2000	8.6293	3.331	3.65	5.19	5.645	4.356	4.959	25.73	5631.77	307.09	8.3656	1.03445	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/7/2000	8.656	3.322	3.618	5.165	5.625	4.257	4.868	21.72	5816.44	312.98	8.4097	1.02987	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/10/2000	8.6655	3.317	3.61	5.1	5.555	4.163	4.785	21.71	5899.13	322.08	8.4521	1.0248	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/11/2000	8.6755	3.315	3.605	5.18	5.655	4.318	4.926	22.5	5845.9	323.08	8.3921	1.02849	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/12/2000	8.6525	3.322	3.617	5.18	5.675	4.303	4.961	22.84	5818.12	323.1	8.3979	1.02934	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/13/2000	8.631	3.322	3.608	5.155	5.635	4.281	4.935	21.71	5867.31	325.07	8.4156	1.0268	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/14/2000	8.577	3.321	3.6	5.11	5.58	4.281	4.942	19.66	6043.79	331.65	8.4615	1.01802	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/17/2000	8.5603	3.316	3.612	5.155	5.66	4.287	4.962	19.66	6112.79	339.3	8.458	1.00857	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/18/2000	8.573	3.313	3.65	5.215	5.735	4.321	5.042	21.5	5958.23	332.39	8.455	1.00929	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/19/2000	8.5825	3.308	3.717	5.19	5.705	4.282	5.024	21.72	5967.46	332.35	8.4755	1.0101	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/20/2000	8.5825	3.31	3.74	5.19	5.72	4.283	5.065	21.75	5980.4	333.86	8.439	1.00929	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/21/2000	8.58	3.31	3.767	5.23	5.745	4.294	5.066	20.82	5906.04	334.77	8.4995	1.00675	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/24/2000	8.545	3.317	3.82	5.23	5.745	4.3	5.059	24.07	5905.37	339.05	8.5202	1.0019	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8
1/25/2000	8.5325	3.322	3.785	5.21	5.695	4.289	5.011	23.02	5823.84	334.61	8.5113	1.00371	0.6	1.3	-0.8	0	115.2	89.9	975471	4137.8

Figure 2. A subset of the daily dataset

These factors can be categorized into fundamental and non fundamental factors.

- **Fundamental Factors**

The *fundamental* factors can be organized into three main categories:

a) Interest rate parity factors

According to the uncovered interest rate parity, a currency with a higher interest rate is expected to depreciate by the amount of the interest rate difference so that the returns on the investment in the two currencies are the same. For example, if the interest rate in the euro area is 1% and 1.5% in Sweden, then the Swedish krona is expected to depreciate by about 0.5%.

b) Purchasing Power parity factor

Under the purchasing power parity condition, a currency with a higher inflation rate is expected to depreciate vis-à-vis currency with a lower inflation rate. The inflation rate and currency have an inverse relationship. The theory of purchasing power parity is another form of the law of one price, which states that with unimpeded trade, identical goods will be sold at the same price.

c) Balassa-Samuelson factors

According to the Balassa-Samuelson hypothesis the real exchange rate between each pair of countries increases with the tradable sector productivities ratio between these countries and decreases with their non-tradable sector productivities ratio.

In addition to this, a money supply variable is added, which is also expected to affect the exchange rate. When the money supply of a country exceeds its demand, the currency value depreciates, whereas when the demand exceeds the supply, the foreign currency depreciates. These approaches should not lead to the belief that whenever the specified factor moves in a specific direction, the currency will also move accordingly. It should be kept in mind that these are one of the many factors that influence exchange rates.

- **Non-fundamental Factors**

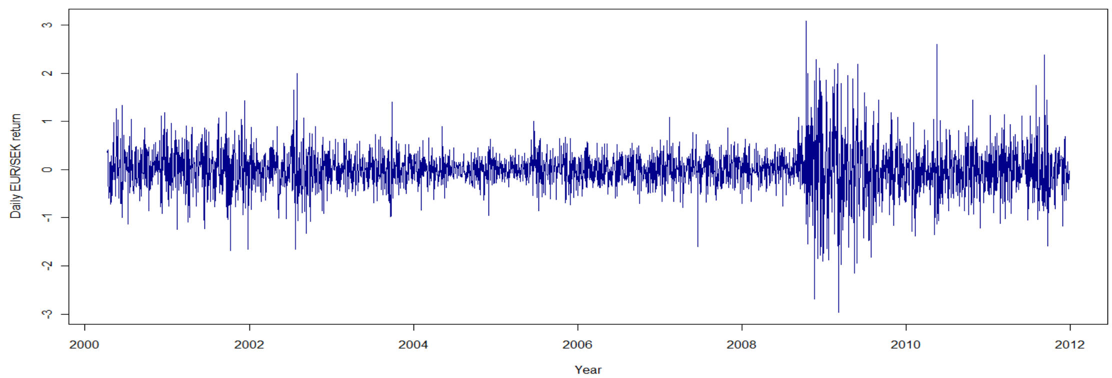
In addition to *fundamental* factors, exchange rates respond to other factors such as changes in the perception of risk and risk appetite, which are called non-fundamental factors. Therefore we add to our dataset measures of general risk aversion such as the VIX as well as the Confidence Indicator which are based on Swedish data and thus capture changes in risk perception from a domestic investor perspective.

2.3 Data Transformations

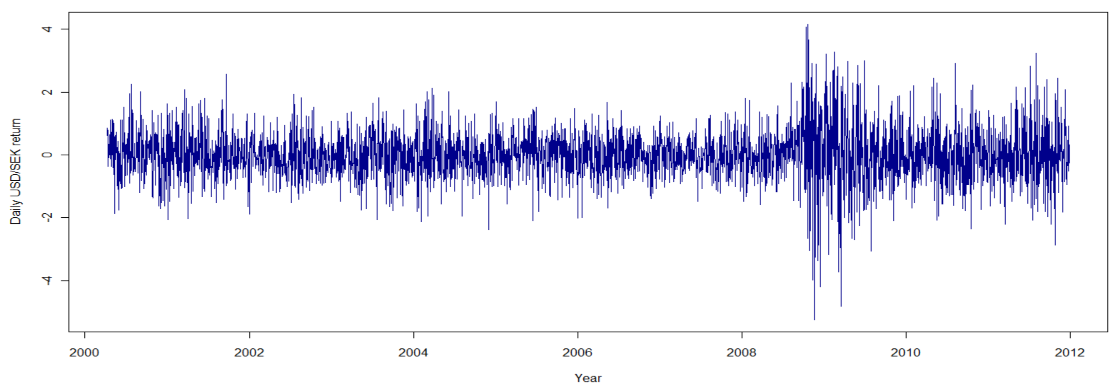
Let E_t denote the exchange rate at time t and define the percentage change in return as:

$$r_t = 100 * \ln\left(\frac{E_t}{E_{t-1}}\right)$$

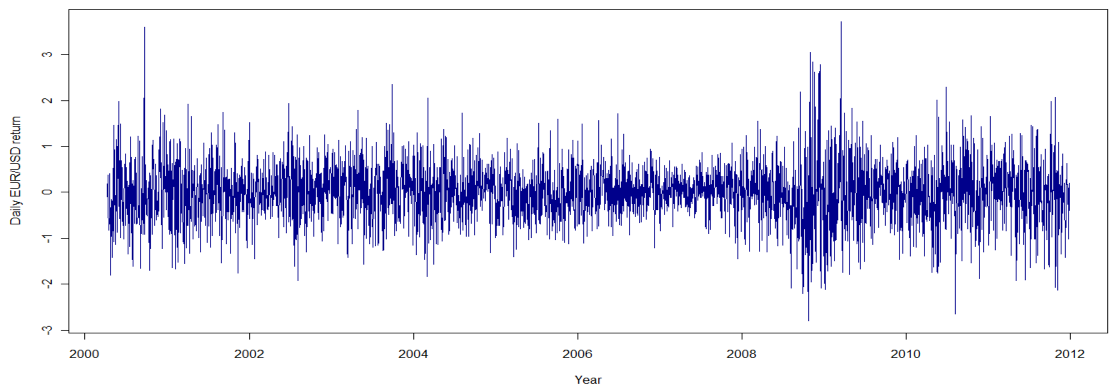
A display of daily exchange returns for all currency pairs is shown in Figure 3. The features of volatility clustering, i.e. periods of high volatility and low volatility can be seen. Following the exchange rate forecasting literature, we will use r_t as our response variable.



(a)



(b)



(c)

Figure 3. Daily returns for (a) EUR/SEK (b) USD/SEK (c) EUR/USD

The economic variables that will be used and shown in Figure 1 and Figure 2 are transformed in order to make one-step-ahead forecasts. The interest rate differential (IRD) is calculated as the difference between the interest rate of the two currencies. It is computed for both short term and long term interest rates. The variables money stock and equity index are transformed by taking the first difference of their logarithms using *ln*(money stock at time *t*) – *ln*(money stock at time *t-1*) and *ln*(equity index at time *t*) – *ln*(equity index at time *t-1*) respectively. The remaining variables are transformed in the same manner. In addition to these variables, lagged variables are also added. Table 1 shows the list of regressors used in the forecasting model of returns of a particular currency pair.

Table 1. Regressors

Exchange Returns	**Regressors**
EUR/SEK	Short term IRD, Long term IRD for 2 years and 5 years, Transformed money supply, Transformed Equity Index, Confidence variable, CPI Inflation, GDP change, Risk Appetite measure, EUR/USD returns, USD/SEK returns, lagged variables
EUR/USD	Short term IRD, Long term IRD for 2 years and 5 years, Transformed money supply, Transformed Equity Index, Confidence variable, CPI Inflation, GDP change, Risk Appetite measure, EUR/SEK returns, USD/SEK returns, lagged variables
USD/SEK	Short term IRD, Long term IRD for 2 years and 5 years, Transformed money supply, Transformed Equity Index, Confidence variable, CPI Inflation, GDP change, Risk Appetite measure, EUR/USD returns, EUR/SEK returns, lagged variables

3. Methods

The methods used in this study are described in this section. The estimation procedure is discussed in Section 3.9. For carrying out the analysis, R software has been used. The description of the packages used for the corresponding method and the particular choices of tuning parameters for the algorithm are also provided.

3.1 Design of the forecast evaluations

In order to make prediction in time series, the data needs to be adjusted such that if return at time t (r_t) needs to be predicted then all the historical information available until time t is used. By doing so, the latest information related to the economic variable is used in estimating the forecasting model. The process is shown in Figure 4, where TS denotes the training sample and TO denote the test observation. It shows how the training sample is continuously updated at every one-step ahead forecast.

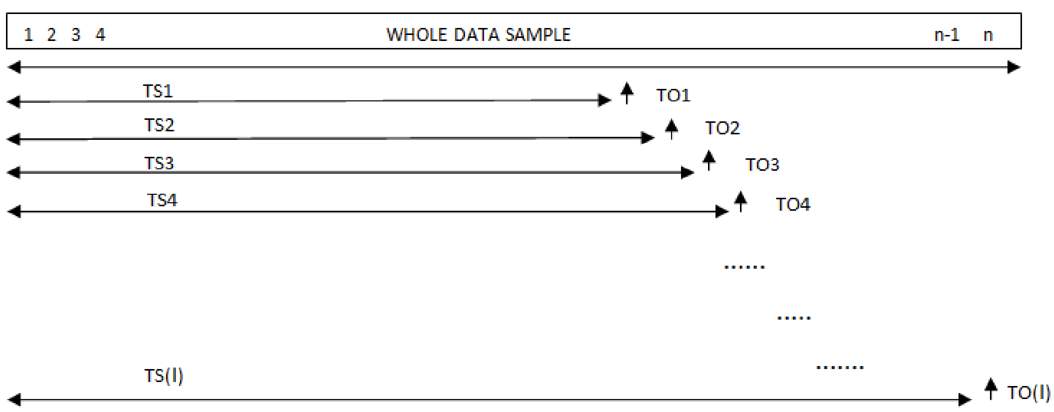


Figure 4. Design of the forecast evaluations

3.2 Autoregressive model

The autoregressive model attempts to predict the future output on the basis of linear formulation of the previous outputs. It is the simplest univariate time series model. The notion AR(p) indicates the autoregressive model of order p, defined as:

$$Y_t = c_t + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \varepsilon_t$$

where φ_i are the parameters of the model and ε_t is white noise, typically following a normal distribution.

The major limitation of this model is the pre-assumed linear form. The approximation of a complex real world problem by a linear model does not always give satisfactory results (Kumar and Thenmozhi, 2007). For simplicity, we model first-order case where p=1, i.e.

$$Y_t = c_t + \varphi Y_{t-1} + \varepsilon_t$$

The AR(1) process is sometimes called the Markov process, after the Russian A. A. Markov. For fitting an AR model, the R command arima() in the ts library has been used.

3.3 ARCH/GARCH

ARCH/GARH (Autoregressive conditional heteroscedasticity/generalized autoregressive conditional heteroscedasticity) is by far the most popular models used for analyzing volatility in financial context. With the adoption of these tools, the heteroscedasticity was modeled only up to a certain extent; however, they provided much better volatility forecasts as compared to traditional approaches. The ARCH model was introduced by Engle (1982). The ARCH process can be defined in a variety of contexts. Defining it in terms of the distribution of the errors of a dynamic linear regression model as in Bera (1993), the dependent variable Y_t, at time t, is generated by

$$Y_t = X_t' \xi + \varepsilon_t$$

where X_t defines the predictor vector of length k, which consists of lagged variables of the dependent variable, and ξ is a k-dimensional vector of regression parameters. The ARCH model characterizes the distribution of the stochastic error ε_t conditional on the realized values of the set of variables $\psi_{t-1} = [Y_{t-1}, X_{t-1}, Y_{t-2}, X_{t-2}, \dots]$. The original formulation of ARCH assumes

$$\varepsilon_t | \psi_{t-1} \sim N(0, h_t)$$

where $h_t = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$.

The coefficients α_i are estimated from the data. A useful generalization of this model is the GARCH process introduced by Bollerslev (1986). The most widely used GARCH specification asserts that the best predictor of the variance in the next period is a weighted average of the long-run average variance, the variance predicted for this period, and the new information in this period that is captured by the most recent squared residual (Engle, 2007). The GARCH process with order p and q is denoted as GARCH(p,q), expressed as

$$h_t = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$$

The ARCH(q) model is same as the GARCH(0,q) model. The ARCH/GARCH behavior of the error term depends on the model chosen to represent the data. We might use different models to represent data with different levels of accuracy (Engle, 2007). For this study, the GARCH(0,1) has been used for simplicity, but our estimation procedure in Section 3.9 is applicable for general GARCH(p,q) process. The garch() function in the R library tseries library is used for estimation and prediction.

3.4 Regression Trees

A regression tree is a prediction model represented as a decision tree. A decision tree is a graphical representation where each internal node represents a test on one of the input variables and the terminal nodes (also called leafs) are the decision or prediction. The prediction at the terminal node is the mean of all the response values within that cell. In linear regression, a global model is developed for the whole data set, i.e. it is this global model that will be used for prediction and we know that its performance degrades if there are nonlinear relationships present in the data set or the variables interact in a complicated fashion. A regression tree is a constant piecewise model and therefore can better cope with non-linearities and interactions in the data. It is one of the fundamental techniques employed in data mining and one of its main benefits is that it gives a visual representation of the decision rules at the nodes that are used for making predictions.

A regression tree is grown as a binary tree, i.e. each node in a tree has two child nodes. Basically, all trees start with a root node and then at each node we determine the split using the explanatory variable (from the given set of explanatory variables), which causes the maximum reduction in the deviance. While traversing down the tree at each given node, a condition is being tested on the basis of which we decide whether to move to the left or to the right sub-branch. If the condition is satisfied, we traverse down through the left sub-branch else down the right sub-branch. The decision is made at the leaf node. The prediction at leaf c is calculated using:

$$m_c = \frac{1}{n_c} \sum_{i \in c} y_i$$

where n_c is the number of observations within the leaf node. As we have partitioned the sample space into c regions $R_1, R_2, \ldots, R_c$, the response is modeled as

$$f(x) = \sum_{i=1}^{c} m_c \, I(x \in R_c),$$

and the sum of squared errors for a tree 'T' is calculated as

$$S = \sum_{c \epsilon leaves(T)} \sum_{i \epsilon c} (y_i - m_c)^2$$

The two most widely used R packages for estimation and prediction of regression tree are tree and rpart, where tree package is the simplest package. Here, rpart package is used. It has the following parameters specifying how the tree is to be grown:

1. **Control**: It is a list of parameters controlling the growth of the tree.

- **cp** the threshold complexity parameter, which specifies the reduction in the deviance if a split is attempted.
- **minsplit** specifies the minimum number of observations at a node for which it will try to compute the split.
  The default value for minsplit is 500 and for cp it is 0.01.

3.5 Random Forests

Breiman (2001) proposed the method of random forests, which has been shown to generate accurate predictive models. It automatically identifies the important predictors, which is helpful when the data consists of lots of variables and we are facing difficulties in deciding which of the variables need to be included in the model. The random forest is an ensemble method that combines a large number of separately grown trees.

In the construction of random forests bootstrap samples of the data are used to construct the trees, however, the construction differs from that of a standard tree. At each node, the best variable for the split is decided among a subset of the explanatory variables chosen randomly at that node, while for standard trees, each node is split using the best one among the whole set of variables. This technique is robust against overfitting of the model. Each tree is constructed on about 63% of the original data set supplied. The remaining 37% is available to test any of the trees. Thus a random forest is said to be self-tested. After constructing B trees, the random forest predictor is

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b)$$

where Θ_b characterises the b^{th} random forest tree in terms of split variables, cut-points at each node and leaf nodes.

The R package named randomForest() implements estimation and prediction with random forests. It has the following tuning parameters:

1. **ntree**- number of trees to be constructed to build the random forest. The default number of trees to grow is 500.

2. **mtry**- the number of predictors which should be tried at each node split. When we are performing regression by the random forest method, the default mtry is p/3 where p is the number of predictors.

It was suggested by Breiman (2001) that for selecting mtry, one should try the default value, half of the default and twice of the default and then pick the one that performs best.

3.6 Support Vector Machines (SVM)

SVM is a supervised learning algorithm. It is usually implemented for classification problems but is also used for regression analysis. It simply applies a linear model but in a high dimensional space which is nonlinearly related to its input space. The key point in SVM is to minimize an upper bound on the expected risk, instead of minimizing error on training data while automatically avoiding overfit to the data. SVM can be defined as a system which uses a hypothesis space of linear functions in a high dimensional feature space. It uses an implicit mapping of input data into a high dimensional feature space defined by a kernel function. Using a kernel function is useful when the data is far from being linearly separable. A good way of choosing the kernel function is via a trial and error procedure. Therefore, one has to try out more than one kernel function to acquire the best solution for a particular problem. In regression analysis, SVM employs the Ɛ-insensitive loss function, i.e.

$$\|y - f(x)\|_{\varepsilon} = \max\{0, \|y - f(x)\| - \varepsilon\}$$

By using the above function, errors less than the threshold, Ɛ, will be ignored.

The R package e1071 implements SVM. It has the following parameters:

1. **Kernel**: Specifies the type of kernel to be employed. The e1071 package has the following menu of choices: radial, polynomial, sigmoid and linear kernel.

2. **Epsilon**: As described earlier, this is the epsilon in Ɛ-insensitive loss function. The default value is 0.1.

3.7 Bayesian Additive Regression Trees (BART)

BART (Chipman *et al.*, 2012) is a non-parametric regression approach. Like the random forests, BART is a sum of trees model, which can more easily incorporate additive effects as compared to a single tree. The essential idea is to elaborate the sum-of-trees model by imposing a prior that regularizes the fit by keeping the individual tree effects small. The sum-of tree model is:

$$y = f(x) + \varepsilon,$$

where f is the sum of many tree models and $\varepsilon \sim N(0, \sigma^2)$. More specifically,

$$y = \sum_{j=1}^{m} g\left(x, T_j, M_j\right) + \varepsilon$$

where T_j represents a regression tree with its associated terminal node parameters M_j and $g\left(x, T_j, M_j\right)$ is the function, which assigns $\mu_{ij} \in M_j$ to x. Here, i represent the terminal node of the tree j.

This model can incorporate main as well as interaction effects. The trees in the BART model are constrained by a regularization parameter to be weak learners. Fitting and inference are being accomplished via an iterative Bayesian backfitting MCMC algorithm. Effectively, it uses dimensionally adaptive random basis elements. This approach enables full posterior inference including point and interval estimates of the unknown regression function as well as the marginal effects of potential predictors. By keeping track of predictor inclusion frequencies, BART can also be used for model-free variable selection. BART's flexibility comes at a computational cost, however.

The R package BayesTree contains the function bart() implementing BART. It has a tuning parameter ntree specifying the number of trees that should be constructed when estimating the BART model. The default value is 200.

3.8 Least Absolute Shrinkage and Selection Operator (LASSO)

In high dimensions, traditional statistical estimation such as procedure OLS tends to perform poorly. In particular, although OLS estimators typically have low bias, they tend to have high prediction variance, and may be difficult to interpret (Brown, 1993). The paper by Tibshirani (1996) suggested LASSO which performs coefficient shrinkage and variable selection simultaneously. It minimizes the mean squared error subject to the constraint that the sum of absolute values of coefficients should be less than a constant. This constant is known as a tuning parameter. LASSO has the favorable features of two techniques: shrinkage and covariate selection. It shrinks some coefficients and sets others to 0, thereby providing interpretable results. The LASSO solution computation is a quadratic programming problem with linear inequality constraints, and can be tackled by standard numerical analysis algorithms.

In LASSO the model fit is:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \mathcal{E}$$

The criterion used is as follows:

$$\min \sum (y - \hat{y})^2$$

subject to the constraint

$$\sum |\beta_j| <= s$$

where s is the tuning parameter, controlling the amount of shrinkage to be applied to the estimates. Alternatively, it can be thought as solving the penalized likelihood problem

$$\min \frac{1}{n} (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{d} |\beta_j|$$

Here LASSO shrinks each coefficient by a constant factor lambda λ, which truncates at zero. This is called *soft thresholding* (Hastie, 2001).

If the tuning parameter is large enough, then the constraint will have no effect and the solution obtained will be just as with multiple linear least squares regression. Cross validation is a useful technique for estimating the best value of s. The most common forms of cross validation are k-fold, leave-one-out and the generalized cross validation. The R package lars implements LASSO. It uses k-fold cross validation.

3.9 Estimation Procedure

In time series forecasting, we are always interested in forecasting on the basis of all the information available at the time of the forecast. Therefore, we consider the distribution of the variable 'Y' at time t conditional on the information available at time t, i.e. X_t. To the best of our knowledge, no previous study has implemented flexible machine learning with volatility clustering models. The Cochran-Orcutt iterative procedure (Cochran and Orcutt, 1949) is a procedure in econometrics for estimating a time series linear regression model in the presence of auto-correlated errors. Using the same concept but on the machine learning models, we propose the estimation procedure described below.

The steps for forecasting returns are based on the equations:

$$\widehat{Y_t} = f(\Theta; X_t) \qquad (1)$$

$$h_t = \omega + \alpha\varepsilon_{t-1}^2, \qquad (2)$$

$where\ \varepsilon_t = Y_t - \widehat{Y_t}\ and\ h_t\ is\ the\ variance\ of\ Y_t.$

1. Estimate the model (1) using the machine learning tools. Then, compute its residuals.
2. Fit a GARCH(0,1) to the residuals from Step 1 to estimate the time varying variance using model (2). This yields $\sqrt{h_t}$, the estimated standard deviation where $t=1,2,\ldots..T$.
3. Transform Y_t using $\sqrt{h_t}$ from Step 2 to reduce the volatility clustering effect on Y_t.

$$Y_t^* = Y_t / \sqrt{h_t}$$

4. Using Y_t^*, repeat Steps 1-3 until convergence.

We investigate the forecasting performance with 0, 1 and 2 iterations of the algorithm. The predicted values are transformed back to the original scale so as to see the change in the performance of the model.

The computations needed to carry out all of the repeated estimations and predictions over time are very demanding, especially for the random forest, SVR and BART on daily returns. The need of parallel computing becomes a priority to accomplish this, and we use a computing cluster with a total of 96 CPU cores and 368 GB RAM divided into four computational nodes. With the help of this, the tasks are distributed among the cores giving a manageable computational burden.

4. Results

Following the models described in the previous section, the performance of each of them is evaluated in this section. The column named *parameter* in the tabular results specifies the choice of the tuning parameter for the corresponding model. In order to compare, performance using a constant model is also computed. This is simply the mean of the dependent variable in the training set. The default parameters for constructing a regression tree are minsplit=20 and cp=0.01. Random forests using default ntree=500 with three different recommended choices of the mtry parameter, suggested by Breiman (2001), are tried. The SVR is estimated using eps-regression with epsilon=0.1 (default) with all four choices of kernel: linear, radial, sigmoid and polynomial available in the e1071 package. For BART the default choice of ntree=200 is used.

4.1 Results for Monthly Returns

For the data on monthly returns, the total number of observations in the whole sample is 140. The data is then divided into a 70:30 split for training and testing. This results in 98 observations in the first training set and the remaining 42 in the test set. Table 2 shows the summary of the descriptive statistics for monthly returns.

Table 2. Descriptive statistics for monthly returns

Currency pair	Minimum	Maximum	Mean	Median	Skewness	Kurtosis	Standard Deviation
EUR/SEK	-5.816	6.447	0.04223	-0.01786	0.23342	6.9451	1.4153
USD/SEK	-7.1010	10.840	-0.1945	-0.2413	0.39760	3.8339	2.9200
EUR/USD	-7.7810	6.4220	0.2365	0.3111	-0.1091	3.2222	2.5809

It can be seen that the distribution of the monthly returns for all currency pairs are non-normal with a kurtosis greater than three. The monthly returns of EUR/SEK and

USD/SEK have a positive skewness coefficient while the monthly returns of EUR/USD are slightly negatively skewed.

The performance evaluation for EUR/SEK, EUR/USD and USD/SEK monthly returns are summarized in Table 3, 4 and 5 respectively. Figure 5 shows the comparison of actual and predicted monthly returns for EUR/SEK, EUR/USD and USD/SEK using the best model, which is random forest for EUR/SEK and SVR for the other two pair of exchange rates. The actual and predicted monthly returns are shown using blue and red curves respectively. With this graphical representation, accuracy can be judged upon in terms of magnitude as well as direction. Figure 6 shows the comparison of actual and predicted monthly returns for EUR/SEK, EUR/USD and USD/SEK using an AR(1) model.

Table 3. Predictive performance for monthly returns of EUR/SEK

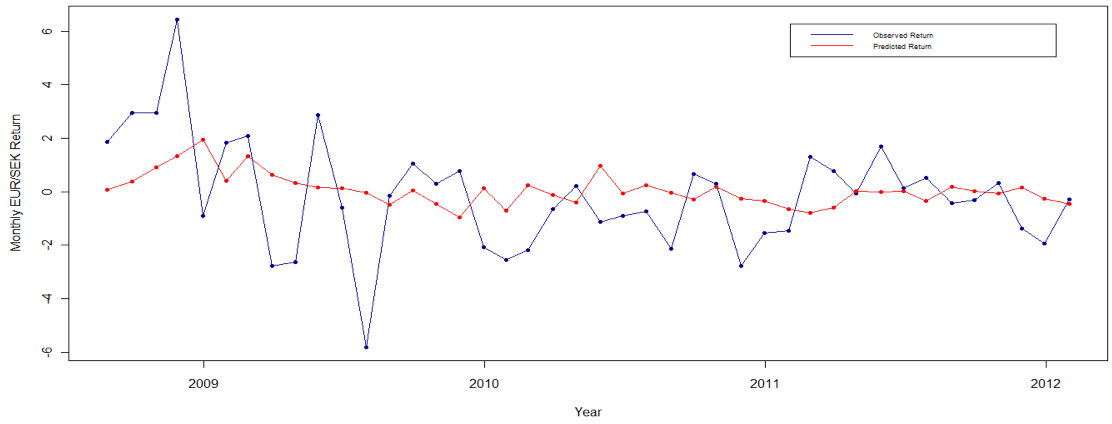
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	2.070	2.058	2.058
AR(1)	-	2.056	2.028	2.037
Regression Tree	Default	2.212	2.071	2.164
Random Forest	mtry=half(default)	2.048	2.002	2.002
	mtry=default	2.049	2.008	2.014
	mtry=double(default)	2.051	1.978	2.005
Support Vector Regression	Kernel=linear	2.285	2.397	2.040
	Kernel=radial	2.049	2.062	2.049
	Kernel=polynomial	3.054	10.607	2.993
	Kernel=sigmoid	2.241	2.345	2.104
LASSO	Default	2.344	2.493	2.122
BART	default	2.184	2.040	1.982

Table 4. Predictive performance for monthly returns of EUR/USD

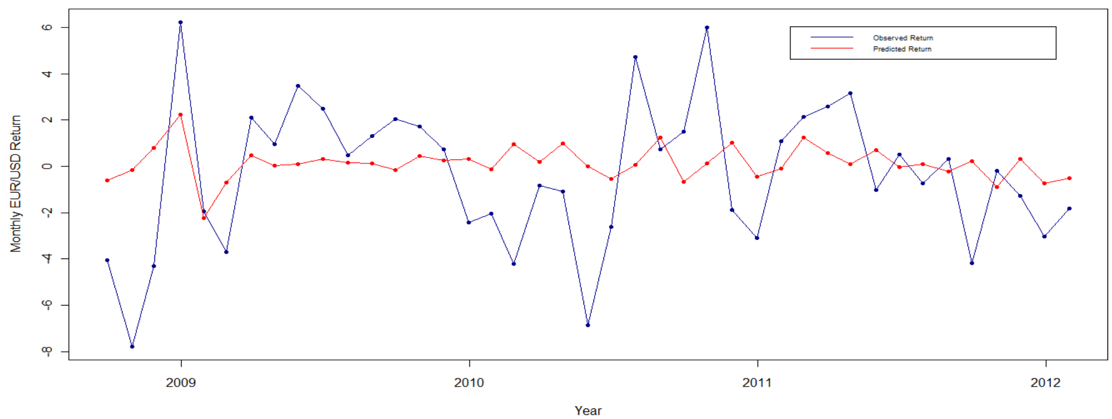
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	3.171	3.182	3.132
AR(1)	-	3.037	3.038	3.034
Regression Tree	Default	3.563	4.132	3.221
Random Forest	mtry=half(default)	3.185	3.207	3.079
	mtry=default	3.184	3.208	3.070
	mtry=double(default)	3.166	3.194	3.059
Support Vector Regression	Kernel=linear	3.399	3.487	3.091
	Kernel=radial	3.096	3.159	3.065
	Kernel=polynomial	6.812	31.112	5.314
	Kernel=sigmoid	3.115	3.330	2.975
LASSO	Default	3.407	3.429	3.084
BART	default	3.239	3.245	3.051

Table 5. Predictive performance for monthly returns of USD/SEK

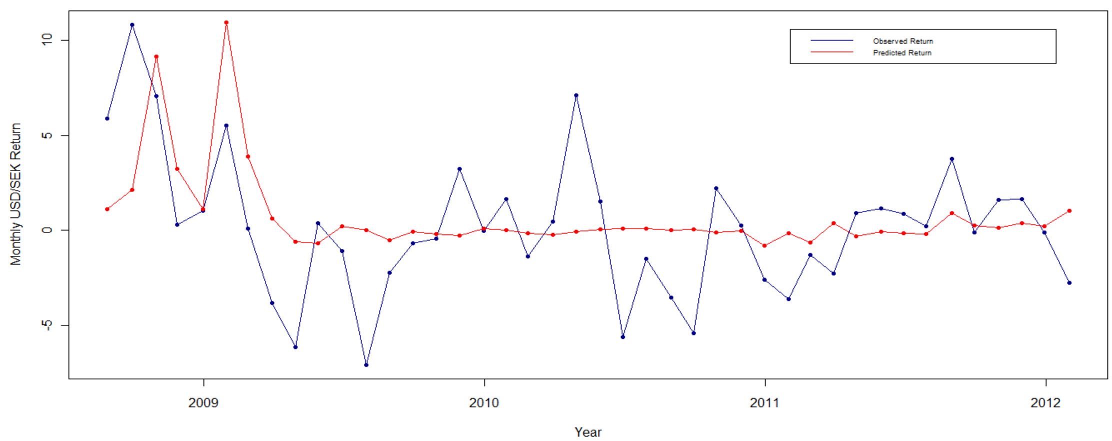
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	3.655	3.655	3.668
AR(1)	-	3.279	3.285	3.438
Regression Tree	Default	3.949	3.959	3.541
Random Forest	mtry=half(default)	3.551	3.506	3.478
	mtry=default	3.550	3.573	3.478
	mtry=double(default)	3.533	3.541	3.454
Support Vector Regression	Kernel=linear	3.504	3.550	3.393
	Kernel=radial	3.534	3.569	3.543
	Kernel=polynomial	7.791	30.528	3.344
	Kernel=sigmoid	3.878	4.143	3.523
LASSO	Default	3.627	3.707	3.365
BART	default	3.607	3.594	3.435



(a)

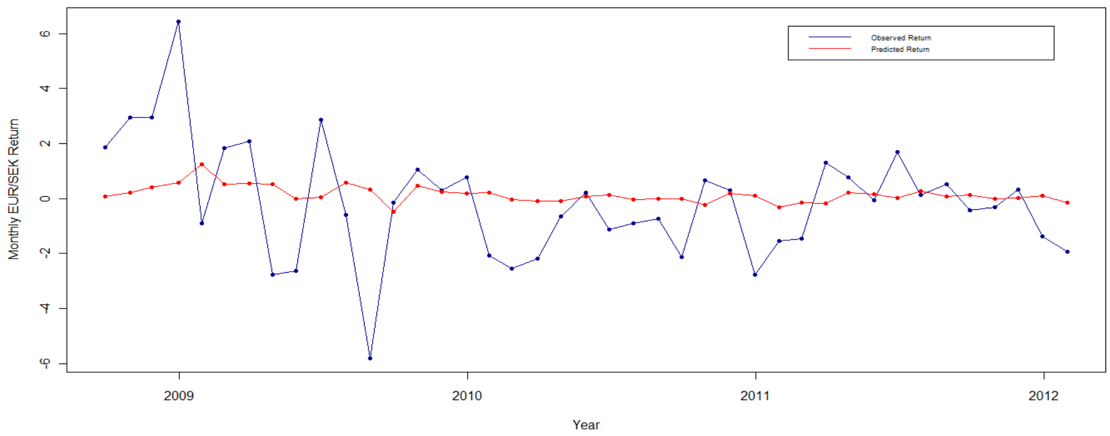


(b)

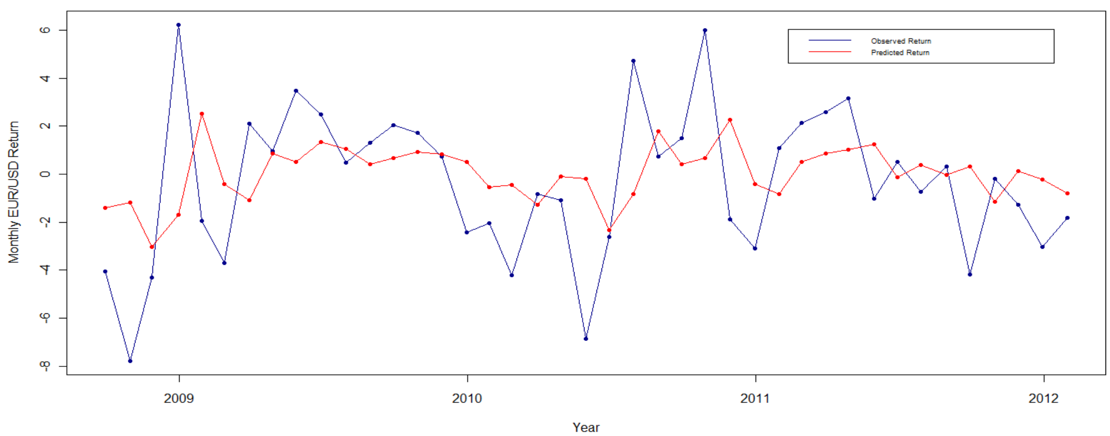


(c)

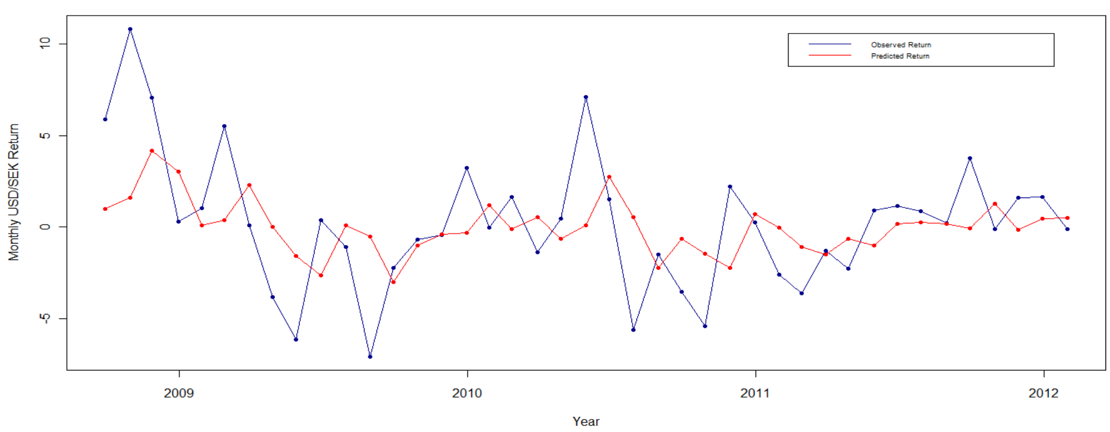
Figure 5. Predicted and observed monthly returns shown in red and blue curves using the best model: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)



(b)



(c)

Figure 6. Predicted and observed monthly returns shown in red and blue curves using an AR(1) model: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK

4.2 Results for Daily Returns

For the data on daily returns, the total number of observations is 3064. The data is divided into 70:30 for training and testing. This results in 2141 observations in the preliminary training sample and the remaining 918 in the test sample. Table 6 shows the summary of descriptive statistics for daily returns.

Table 6. Descriptive statistics for daily returns

Currency pair	Minimum	Maximum	Mean	Median	Skewness	Kurtosis	Standard Deviation
EUR/SEK	-2.963	3.086	0.0024	-0.0021	0.1823	8.06	0.4553
USD/SEK	-5.246	4.160	-0.0071	-0.0247	0.0359	5.481	0.8361
EUR/USD	-2.797	3.719	0.0097	0.0175	0.0948	4.535	0.6603

Table 6 shows that the kurtosis for all three exchange returns are greater than 3, implying that the distribution is non-normal. It is also seen that the mean of all returns are almost zero. All returns have a positive skewness coefficient. The performance evaluation for EUR/SEK, EUR/USD and USD/SEK daily returns are summarized Table 7, 8 and 9 respectively. Figure 7 and 8 shows the comparison of actual and predicted daily returns for EUR/SEK, EUR/USD and USD/SEK using the best model and an AR(1) model. The actual and predicted monthly returns are shown in blue and red curves respectively.

Table 7. Predictive performance for daily returns of EUR/SEK

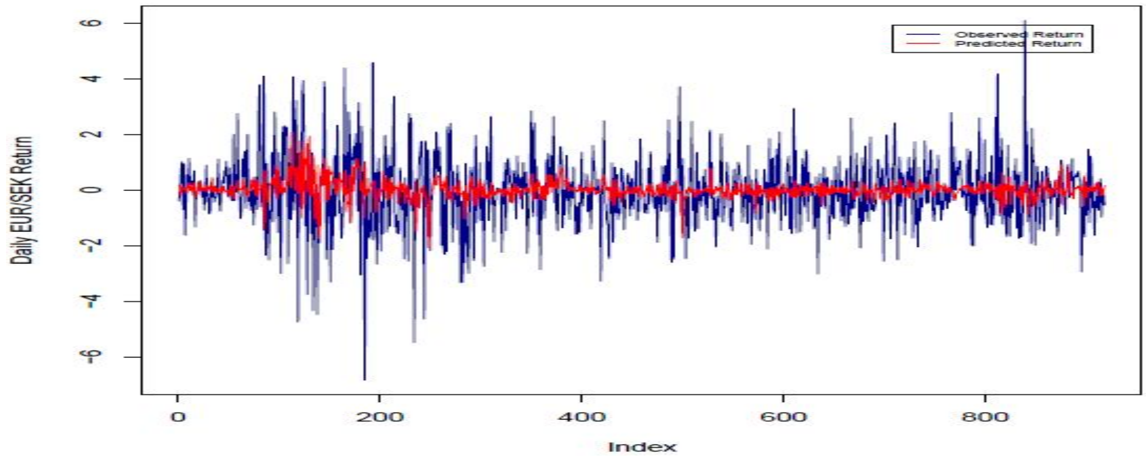
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	0.641	0.641	0.641
AR(1)	-	0.641	0.642	0.642
Regression Tree	Default	0.658	0.672	0.699
Random Forest	mtry=half(default)	0.657	0.673	0.693
	mtry=default	0.659	0.675	0.702
	mtry=double(default)	0.661	0.689	0.721
Support Vector Regression	Kernel=linear	0.644	0.650	0.658
	Kernel=radial	0.654	0.661	0.702
	Kernel=polynomial	1.000	2.998	3.132
	Kernel=sigmoid	12.162	169.61	58.557
LASSO	Default	0.646	0.647	0.652
BART	default	0.707	0.690	0.764

Table 8. Predictive performance for daily returns of EUR/USD

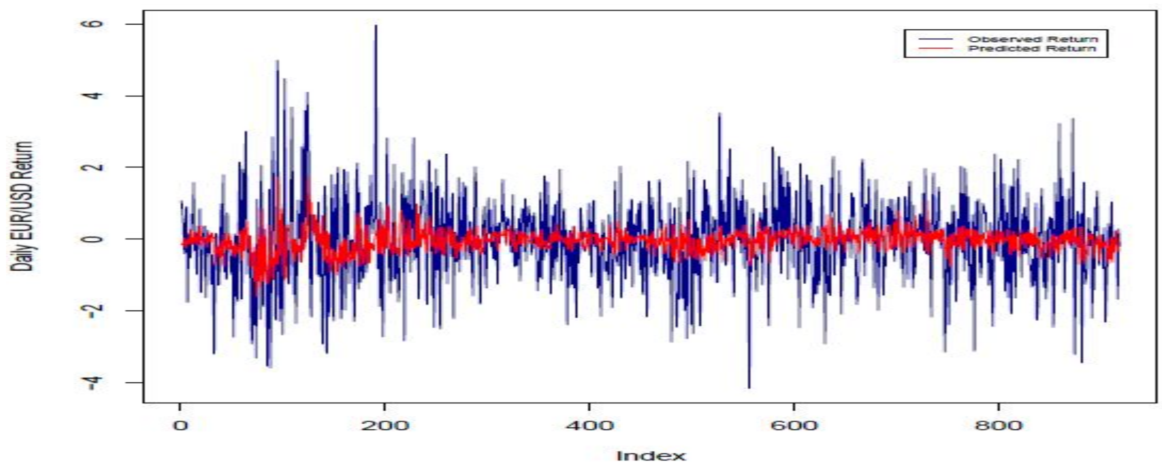
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	0.811	0.811	0.811
AR(1)	-	0.811	0.811	0.811
Regression Tree	Default	0.829	0.832	0.855
Random Forest	mtry=half(default)	0.831	0.827	0.850
	mtry=default	0.833	0.834	0.860
	mtry=double(default)	0.837	0.839	0.869
Support Vector Regression	Kernel=linear	0.823	0.824	0.839
	Kernel=radial	0.829	0.832	0.865
	Kernel=polynomial	1.264	3.181	2.650
	Kernel=sigmoid	18.603	214.821	62.661
LASSO	Default	0.819	0.820	0.832
BART	default	0.850	0.851	0.894

Table 9. Predictive performance for daily returns of USD/SEK

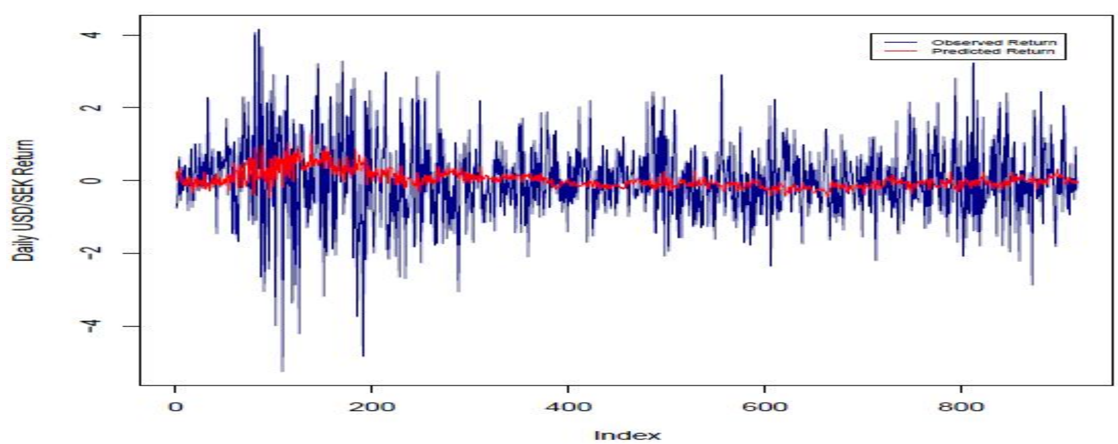
Model Used	Parameter	RMSE		
		Iteration 0	Iteration 1	Iteration 2
Constant	-	1.132	1.132	1.132
AR(1)	-	1.132	1.133	1.133
Regression Tree	Default	1.181	1.194	1.180
Random Forest	mtry=half(default)	1.158	1.170	1.168
	mtry=default	1.167	1.180	1.181
	mtry=double(default)	1.170	1.193	1.179
Support Vector Regression	Kernel=linear	1.154	1.165	1.159
	Kernel=radial	1.154	1.170	1.166
	Kernel=polynomial	1.891	4.027	2.436
	Kernel=sigmoid	22.479	267.829	53.257
LASSO	Default	1.142	1.147	1.143
BART	default	1.216	1.238	1.231



(a)

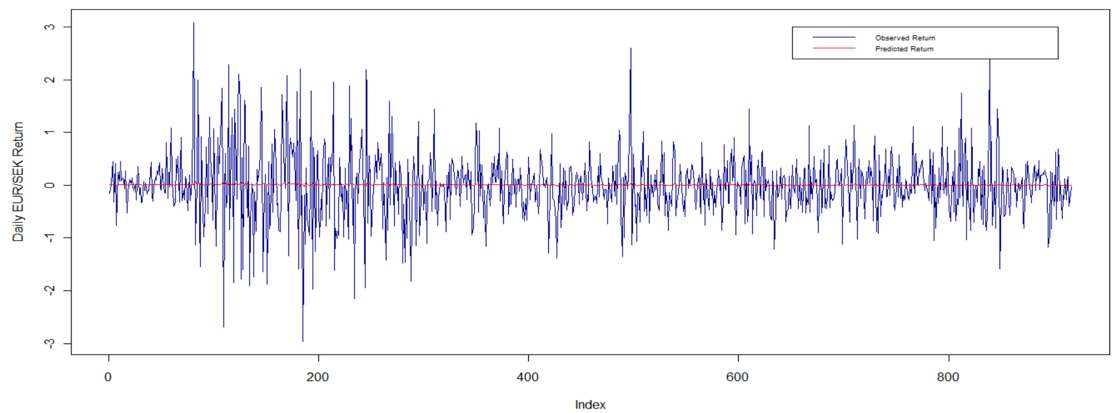


(b)

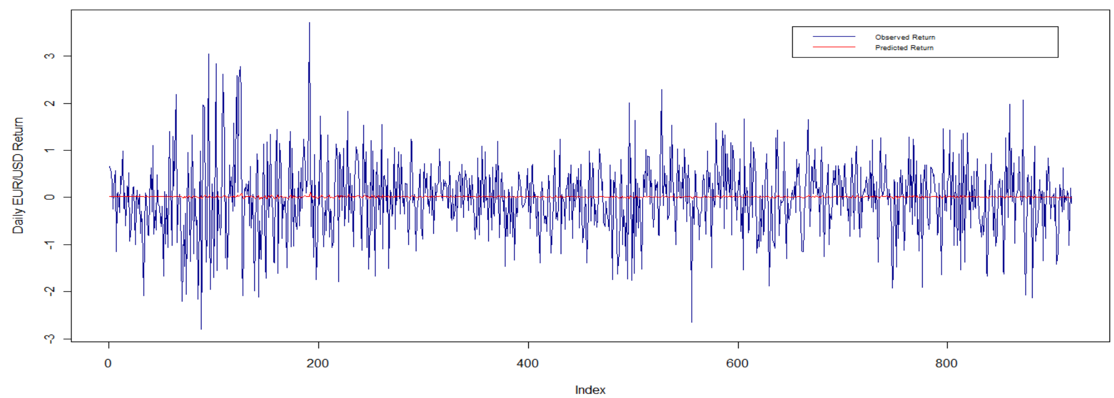


(c)

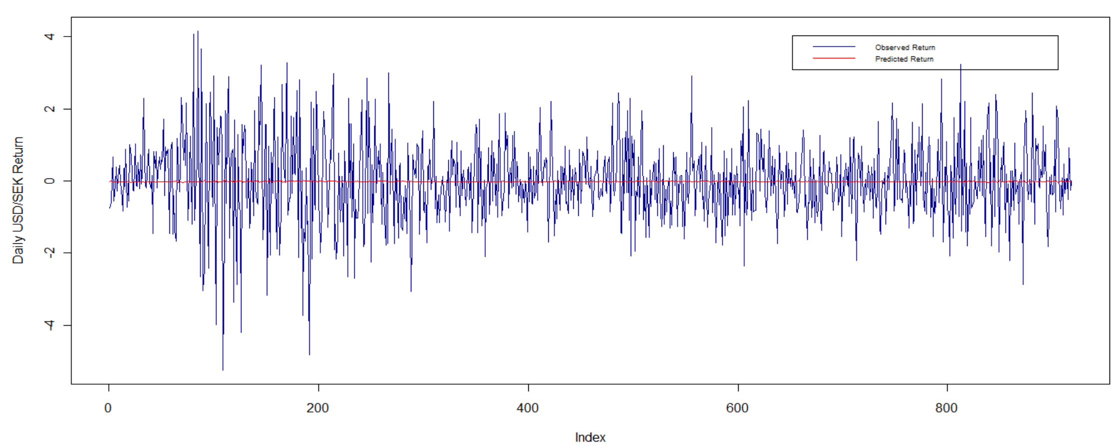
Figure 7. Predicted and observed daily returns shown in red and blue curves using the best model: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)



(b)



(c)

Figure 8. Predicted and observed daily returns shown in red and blue curves using an AR(1) model: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK

Variable importance in regression is an important issue which needs to be considered. To gain an insight of variable importance, the analysis is carried out on the whole sample. The structure of the regression tree includes the most important variables explaining the dependent variables and removing the insignificant ones. For random forests two different measures of variable importance are computed: scaled average of the prediction accuracy and total decrease in node impurity. The first measure *Mean Decrease Accuracy* is computed from permuting the out-of-bag data. For each tree, the prediction error on the out-of-bag portion is recorded. Then the same is done after permuting each variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. The second measure *Mean Decrease RSS* is the total decrease in node impurities from splitting on the variable, averaged over all trees. For ordinary regression analysis, it is measured by residual sum of squares. To analyze variable importance in BART, we use ntree=20. This small value of ntree is chosen as each variable must then compete to get into a smaller number of trees. Then the mean across draws of the percentage of times each variable is used is computed. It is shown in figures in Appendix A under the heading vppostmean, which stands for *variable percentage posterior mean*. In the plots, the blue line is a reference line and is chosen by the analyst. The higher the line is, the smaller is the number of variables that will be selected. The reference line is chosen at 0.05 and 0.04 for monthly and daily returns respectively. As we know, LASSO is a shrinkage method that also does variable selection. Due to the nature of constraint, LASSO sets some of the coefficients to zero, which signals that they are not important.

Table 10 and 11 summarizes variable importance for monthly and daily returns respectively for all currency pairs with important variables marked with black bubble. The detailed tables and figures are found in Appendix A.

Table 10. Important variables for monthly returns marked with black bubble

Variable	EUR/SEK				EUR/USD				USD/SEK			
	R.Tree	RF	BART	LASSO	R. Tree	RF	BART	LASSO	R.Tree	RF	BART	LASSO
Short Term IRD	-	-	-	-	●	●	●	-	-	●	-	-
Long Term IRD (2 years)	-	●	●	-	●	●	●	-	-	●	●	-
Long Term IRD (5 years)	-	●	●	●	●	●	●	●	●	●	●	●
Risk Appetite Measure, VIX	●	-	●	-	-	-	●	-	-	-	-	-
Equity, STOXX Index	●	-	-	-	-	-	●	-	●	-	●	-
Equity, OMSX Index	-	-	-	●	-	-	-	-	-	-	●	●
GDP, swgdpaqq Index	-	-	-	-	-	-	-	●	-	-	-	-
GDP, EUGNEMUQ Index	-	-	-	-	-	-	-	-	-	-	-	-
Inflation, SWCPMOM index	-	-	-	-	-	-	-	-	●	-	-	-
Inflation, ECCPEMUM Index	-	-	●	-	●	-	-	-	-	-	-	-
Confidence variable, SWETSURV Index	-	-	-	-	-	-	-	-	-	-	●	-
Confidence variable, GRZEEUEX Index	●	●	-	●	●	-	-	-	-	-	-	-
Sweden Money Supply	●	-	-	-	-	-	-	●	-	-	-	-
Money Supply, ECMSM2 Index	-	-	-	-	-	-	-	-	●	-	-	-
Lag1	●	-	●	●	●	●	●	-	-	●	●	●
Lag2	●	-	●	-	-	-	●	●	●	-	●	●
Lag3	●	●	●	●		●	-	-	-	-	-	-
Lag4	-	-	●	-	●	-	-	-	-	-	-	●
EUR/USD daily return	-	●	●	-	-	-	-	-	●	●	●	-
USD/SEK daily return	●	●	●	-	-	●	-	●	-	-	-	-
EUR/SEK daily return	-	-	-	-	●	-	-	-	●	-	-	-

Table 11. Important variables for daily returns marked with black bubble

Variable	EUR/SEK				EUR/USD				USD/SEK			
	R.Tree	RF	BART	LASSO	R. Tree	RF	BART	LASSO	R.Tree	RF	BART	LASSO
Short Term IRD	-	●	-	-	-	-	-	-	-	●		
Long Term IRD (2 years)	-	●	-	-	-	●		-	-	●	-	-
Long Term IRD (5 years)	-	●	-	●	-	●	-	●	●	●	-	●
Risk Appetite Measure, VIX	●	●	●	-	-	-	●	●	-	●	●	-
Equity, STOXX Index	●	-	●	-	-	●	-	●	●	-	●	-
Equity, OMSX Index	-	-	●	-	-	●	●	-	-	-	-	
GDP, swgdpaqq Index	-	-	●	-	-	-	-	-	-	-	-	-
GDP, EUGNEMUQ Index	-	-	●	-	-	-	●	-	-	-	-	-
Inflation, SWCPMOM index	-	-	-	-	-	-	●	-	-	-	-	-
Inflation, ECCPEMUM Index	-	-	-	●	-	-	-	-	-	-	-	-
Confidence variable, SWETSURV Index	-	●	-	-	-	●	-	-	-	●	●	●
Confidence variable, GRZEEUEX Index	-	-	-	●	-	-	-	●	●	-	●	●
Sweden Money Supply	-	-	-	●	-	-	●	●	-	-	●	●
Money Supply, ECMSM2 Index	-	-	-	●	-	-	-	●	-	-	-	●
Lag1	-	●	●	-	●	-	●	●	●	-	●	●
Lag2	●		● -	●	-	-	-	●	●		●	●
Lag3	-	-	●	●	-	-	-	●	●	-	●	●
Lag4	●	-	●	●	-	-	●	●	-	-	●	●
EUR/USD daily return	-	-	-	-	-	-	-	-	-	-	-	-
USD/SEK daily return	-	●		●	●	●	●	●	-	-	-	-
EUR/SEK daily return	-	-	-	-	●	-	●	●	-	-	●	-

5. Discussion

From the descriptive statistics on both daily and monthly frequency, it can be seen that the distribution of all currency pairs are non-normal: the kurtosis is greater than three. This is particularly true for daily returns. This is also shown using normal probability plots of the returns series in Figure 9 and 11 in Appendix A. The monthly returns of EUR/SEK and USD/SEK have a positive skewness coefficient, viz. 0.23342 and 0.39760 and EUR/USD has a negative skewness coefficient of -0.1091. All the daily returns series are positively skewed. The means of the returns for monthly and daily frequency are both close to zero, another commonly found characteristic of financial returns. The acf() function in R outputs autocorrelation plots. It describes the strength of relationships between different points in the series. In Appendix, Figure 10 and 12 are shown such plots for all currency pairs on monthly and daily returns respectively. For all daily returns, the autocorrelation coefficients lie within the 95% approximate confidence bounds which implies very weak serial dependence. This simply means that given yesterday's return, today's return is equally likely to be positive or negative.

The results on EUR/SEK monthly returns show that random forest and BART perform comparatively better than AR(1). Random forest and BART performances are comparable as both these are ensemble techniques of regression trees. The RMSE using random forest, BART and AR(1) are 2.002, 1.982 and 2.037 respectively. The random forest model is chosen over BART due to its comparatively less execution time. SVR also shows the potential for having good predictive performance in this case. For EUR/USD and USD/SEK monthly returns SVR performed slightly better comparatively. The variability in the performance of SVR is seen from the different choice of kernel parameters. It shows that SVR is sensitive to the choice of parameters. The RMSE for monthly returns of EUR/USD using SVR with sigmoid kernel and AR(1) is 2.975 and 3.034 respectively. The RMSE for monthly returns of USD/SEK using SVR with polynomial kernel and AR(1) are 3.344 and 3.438 respectively. It is to be noted that when we do not account for heteroscedasticity, then the conventional AR method performs best for EUR/USD and USD/SEK. This is the reason why most of the econometrists still use these methods on these types of financial series. Also, the forecasting performance using the autoregressive model is better than the random walk model comparatively.

Figure 5 and 6 show the comparison of actual and predicted monthly returns for EUR/SEK, EUR/USD and USD/SEK using the best chosen model and an AR(1) model. The EUR/SEK and EUR/USD forecast in Figure 5 seems to be much more accurate in terms of direction as well as in magnitude compared to the autoregressive forecast in Figure 6. The directional accuracy is approximately uniform over time. The accuracy from the perspective of magnitude also seems to be good enough and it can be further improved by modeling it recursively. This will lead to a decrease in mean squared error and the magnitude of the forecasted value will get closer to the observed values. However, USD/SEK forecast from the autoregressive model seems to be comparatively better than the chosen SVR model, at least from a visual inspection point of view.

The results on daily returns are summarized in Table 7, 8 and 9 for EUR/SEK, EUR/USD and USD/SEK respectively. The RMSE for daily EUR/SEK using AR(1), LASSO and SVR with a linear kernel is 0.642, 0.652 and 0.658. The RMSE for daily EUR/USD using AR(1), SVR with linear kernel and LASSO is 0.811, 0.839 and 0.832 respectively. The RMSE for daily USD/SEK using AR(1), SVR with linear kernel and LASSO is 1.133, 1.159 and 1.143 respectively. From these, we observe that SVR and LASSO have the potential of forecasting daily returns effectively. We can observe that there is an indeterministic increase or decrease in error measures in terms of RMSE in the initial iteration steps. It will take a few iteration steps until we see a significant improvement in forecasting. Later, when we visualized the LASSO predictions, they were almost like those of an AR(1). On the other hand, SVR predictions seem to cover some variation why SVR will be preferred. It is also interesting to note here that there is no significant difference between the performances of the autoregressive and the random walk model.

Figure 7 and 8 shows the comparison of actual and predicted daily returns for EUR/SEK, EUR/USD and USD/SEK using the best model and an AR(1) model respectively. It is observed that the forecasted values using SVR are much better than those of the autoregressive model. Although the volatility in daily returns is very high, SVR manages to take into account most of the variation and it can be improved further by modeling it recursively. Despite a continuously updated training data, the AR model is unable to capture the high variation in the series and the predictions are around zero, which is the mean of the returns.

SVR has several properties, which make it a good model for the forecasting of returns. There are several reasons to why SVR can outperform other model; however, the most important property is structural risk minimization (SRM), which has been shown to be superior to traditional empirical risk minimization (ERM). SRM minimizes an upper bound on the expected risk while ERM minimizes the error on training data. In addition to this, the function to be minimized is quadratic and linearly restricted. This ensures that the solution does not get trapped into local minima. In addition to this, SVR does not easily overfit. However the major disadvantage of SVR models is that they are difficult to interpret and that they provide little insight into which variables contribute much greater while making predictions. So far, there is not much literature focusing on assessing variable importance in SVRs. This may be because SVR is a nonlinear and kernel based methodology, which makes it challenging to assess variable importance. This shortcoming also applies to the random forest model because of its generation of many random trees to form the predictions.

Table 10 and 11 summarize variable importance for monthly and daily returns respectively for all currency pairs. From Table 10, it is observed that long term IRD (5 years) and lag 1 of the exchange rate are identified as important variables for all currency pairs in almost each of the methods used. They are also identified as important variables for daily returns in most of the methods for all currency pairs.

GDP and Inflation are not appearing persistently in the group of important variables nor for monthly neither for daily returns. For all monthly returns, this is also the case for money supply and confidence indicator (SWETSURV). Other currency pair returns are in the group of important variables in EUR/USD daily returns and EUR/SEK monthly returns. In addition to this, lagged variables are also important variables for EUR/SEK monthly returns and USD/SEK daily returns.

6. Conclusions and comments for the future

In this thesis, we analyze the performance of widely used models from the machine learning field for predicting exchange rate returns. A novelty of our study is that we extend the machine learning models with a GARCH model for capturing the well documented volatility clustering in financial returns series. We analyze three different exchange rates, both on a daily and monthly frequency. These GARCH-extended machine learning models are then applied to make one-step-ahead predictions by recursive estimation so that the parameters estimated by this model are also updated with the new information. In order to obtain robust results, this study is repeated on three different exchange rate returns: EUR/SEK, EUR/USD and USD/SEK on monthly and daily frequency.

Although the results were mixed, it is concluded that GARCH-extended SVR shows the ability of improving the forecast of exchange rate returns on both monthly and daily frequency. The important variables when accessed on the whole sample across almost all currency pairs are long term IRD (5 years) and one-period lagged exchange rate while GDP and Inflation were unimportant variables for both monthly and daily returns.

For future work, GARCH can be applied instead of GARCH(0,1) to make further comparisons. In the literature, many different extensions of ARCH/GARCH-models have been proposed. It could be interesting to extend GARCH-extended machine learning models by GARCH-type models with Student-*t* distributed errors. By assuming the conditional ***t***-distribution in connection with a GARCH model, it should be possible to capture excess kurtosis. Also, we can always test the effectiveness of these GARCH-extended machine learning models on other financial returns, like stock market returns.

7. Literature

Anaraki K (2007), Meese and Rogoff's Puzzle revisited, *International Review of Business Paper*, pp. 278-304.

Anil B (1993): ARCH Models: Properties, Estimation and Testing, *Journal of Economic Surveys*, Vol.7, No-4, pp. 305-362.

Chipman H, George E and Mcculloch R (2010): BART: Bayesian Additive Regression Trees, *Annals of Applied Statistics*, 4, pp. 266-298.

Cont R: Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models, working paper series. Retrieved on: 14.03.2012 from: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1411462

Cont R, Empirical properties of asset returns: Stylized facts and statistical issues, *Quantitative Finance*, 1 (2001), pp. 1–14.

Dees R and Cronin P (2011). WholeSoldier Variable Selection Using BART, working paper. Retrieved on: 20.05.2012 from: http://www.robdees.com/uploads/1/0/6/5/10651736/deescroninbayespaper.pdf

Engle R, Focardi S and Fabozzi F (2008), ARCH/GARCH Models in Applied Financial Econometrics, In: Handbook Series in Finance (F.J. Fabozzi,ed.), John Wiley & Sons.

Hauner D, Lee J and Takizawa H (2011), In Which Exchange Rate Models Do Forecaster Trust?, *International Monetary Fund working papers*, Working Paper No. 11/116.

Hossain A and Nasser M, Comparison of the finite mixture of ARMA-GARCH, back propagation neural networks and support-vector machines in forecasting financial returns (2011), *Journal of Applied Statistics*, Vol. 38, No. 3, pp. 533-551.

Karatzoglou A, Meyer D and Hornik K (2006), Support Vector Machines in R, *Journal of Statistical Software*, Vol. 15, Issue 9, pp. 1-28.

Liaw A and Wiener M (2002), Classification and Regression by randomForest, *R News*, Vol. 2, No. 3. pp. 18-22.

Kumar M and Thenmozhi M. (2007): Predictability and trading efficiency of S&P CNX nifty index returns using support vector machines and random forest regression, *Journal of Academy of Business and Economics*, Vol. 7 (1), pp. 150-164.

Mehrara M and Oryoie A (2011), Forecasting Exchange Rate Return Based on Economic variables, *International Journal of Economics and Research*, pp. 119-125.

Smola A and Schőlkopf B (2004), A Tutorial on Support Vector Regression, *Statistics and Computing*, 14, pp. 199-222.

Tibshirani R: Regression Shrinkage and Selection via Lasso (1996), *Journal of the Royal Statistical Society,* Series B (Methodological), vol. 58, No. 1. pp. 267-288.

Chatfield C. (1989), *The Analysis of Time Series: An Introduction*, Fourth edition, Chapman & Hall.

Hastie T, Tibshirani R and Friedman J (2001). *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer.

Gujarati N (2009), *Basic Econometrics*, Fifth edition, McGrawHill.

8. Appendix A: Tables and Figures

Table 12 and 13 show the variable importance measure computed for random forest models on the monthly and the daily returns respectively.

Table 12. Variable Importance using random forest for monthly returns

Variable	EUR/SEK		EUR/USD		USD/SEK	
	Mean Decrease Accuracy	Mean Decrease RSS	Mean Decrease Accuracy	Mean Decrease RSS	Mean Decrease Accuracy	Mean Decrease RSS
Short Term IRD	1.867	7.323	**3.127**	27.875	**3.601**	37.961
Long Term IRD (2 years)	**3.814**	8.970	**5.072**	40.950	**3.694**	61.010
Long Term IRD (5 years)	**3.052**	7.705	**6.211**	51.701	**9.537**	100.350
Risk Appetite Measure, VIX	2.675	18.412	1.679	34.848	0.787	39.766
Equity, STOXX Index	1.677	7.845	0.188	42.221	1.640	54.612
Equity, OMSX Index	1.437	8.098	2.815	36.376	-1.088	48.689
GDP, swgdpaqq Index	2.285	9.265	-0.617	33.076	-1.009	49.112
GDP, EUGNEMUQ Index	-0.463	6.543	1.425	29.338	0.457	43.048
Inflation, SWCPMOM index	-2.008	5.203	2.136	37.713	-1.338	46.391
Inflation, ECCPEMUM Index	1.577	10.826	-0.0796	16.873	-1.833	23.482
Confidence variable, SWETSURV Index	-2.198	24.013	2.465	38.487	1.062	52.332
Confidence variable, GRZEEUEX Index	**3.024**	10.536	1.576	53.198	1.065	58.168
Sweden Money Supply	-0.959	47.007	1.370	28.794	-2.915	44.366
Money Supply, ECMSM2 Index	0.490	10.913	0.777	29.765	0.563	37.417
Lag1	1.581	13.122	**8.302**	89.753	**4.630**	95.239
Lag2	-0.308	17.461	-0.308	62.853	-0.472	53.251
Lag3	**3.174**	21.287	**3.548**	56.084	-2.097	46.948
Lag4	-1.462	19.130	-1.986	36.925	-1.949	45.900
EUR/USD daily return	**4.985**	20.891	-	-	**8.527**	117.866
USD/SEK daily return	**8.888**	23.935	**5.114**	92.635	-	-
EUR/SEK daily return	-	-	1.001	46.330	2.017	67.812

Table 13. Variable Importance using random forest for daily returns

Variable	EUR/SEK		EUR/USD		USD/SEK	
	Mean Decrease Accuracy	Mean Decrease RSS	Mean Decrease Accuracy	Mean Decrease RSS	Mean Decrease Accuracy	Mean Decrease RSS
Short Term IRD	**14.126**	29.183	13.709	42.460	**18.521**	102.518
Long Term IRD (2 years)	**15.575**	28.219	**17.651**	40.205	**16.117**	109.505
Long Term IRD (5 years)	**13.222**	29.414	**15.645**	43.221	**20.519**	112.697
Risk Appetite Measure, VIX	**19.751**	51.791	14.994	47.882	**21.042**	155.589
Equity, STOXX Index	11.936	37.620	**22.722**	61.404	13.588	128.140
Equity, OMSX Index	11.029	40.061	**20.519**	64.527	12.035	132.526
GDP, swgdpaqq Index	6.641	13.285	10.240	20.875	8.176	51.703
GDP, EUGNEMUQ Index	6.327	10.369	12.156	21.401	10.361	43.937
Inflation, SWCPMOM index	4.650	12.986	5.546	19.094	5.373	49.974
Inflation, ECCPEMUM Index	6.431	14.784	1.313	15.609	5.232	47.918
Confidence variable, SWETSURV Index	**16.484**	29.055	16.147	36.898	**15.350**	78.163
Confidence variable, GRZEEUEX Index	6.978	18.455	11.930	39.194	10.027	77.686
Sweden Money Supply	0.533	4.417	1.271	3.297	6.595	22.456
Money Supply, ECMSM2 Index	-0.826	2.926	-2.296	3.436	-0.241	17.992
Lag1	**12.335**	49.929	3.160	43.756	11.436	129.401
Lag2	8.605	47.900	0.901	45.018	5.984	147.458
Lag3	5.951	51.148	0.982	41.467	2.855	164.150
Lag4	3.015	53.048	-0.054	43.663	3.632	167.425
EUR/USD daily return	4.243	38.958	-	-	9.939	121.415
USD/SEK daily return	**13.021**	37.055	**176.398**	611.521	-	-
EUR/SEK daily return	-	-	2.732	46.814	7.087	146.871

Table 14 and 15 show the LASSO coefficient estimates for the monthly and daily returns with nonzero coefficients highlighted in bold.

Table 14. LASSO coefficient estimates for the monthly returns

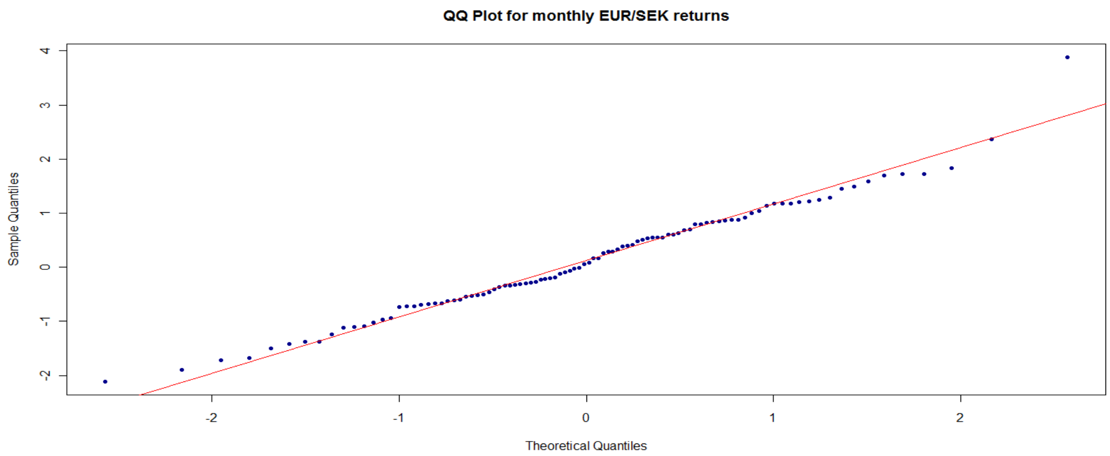
Variable	**EUR/SEK**	**EUR/USD**	**USD/SEK**
Short Term IRD	0.000	0.000	0.000
Long Term IRD (2 years)	0.000	0.000	0.000
Long Term IRD (5 years)	**0.066**	**-0.666**	**1.018**
Risk Appetite Measure, VIX	0.000	0.000	0.000
Equity, STOXX Index	0.000	0.000	0.000
Equity, OMSX Index	**-0.785**	0.000	**-3.045**
GDP, swgdpaqq Index	0.000	**-0.028**	0.000
GDP, EUGNEMUQ Index	0.000	0.000	0.000
Inflation, SWCPMOM index	0.000	0.000	0.000
Inflation, ECCPEMUM Index	0.000	0.000	0.000
Confidence variable, SWETSURV Index	0.000	0.000	0.000
Confidence variable, GRZEEUEX Index	**-0.0005**	0.000	0.000
Sweden Money Supply	0.000	**5.817**	0.000
Money Supply, ECMSM2 Index	0.000	0.000	0.000
Lag1	**0.084**	0.000	**0.227**
Lag2	0.000	**-0.080**	**-0.022**
Lag3	**0.139**	0.000	0.000
Lag4	0.000	0.000	**0.001**
EUR/USD daily return	0.000	-	0.000
USD/SEK daily return	0.000	**-0.251**	-
EUR/SEK daily return	-	0.000	0.000

Table 15. LASSO coefficient estimates for the daily returns

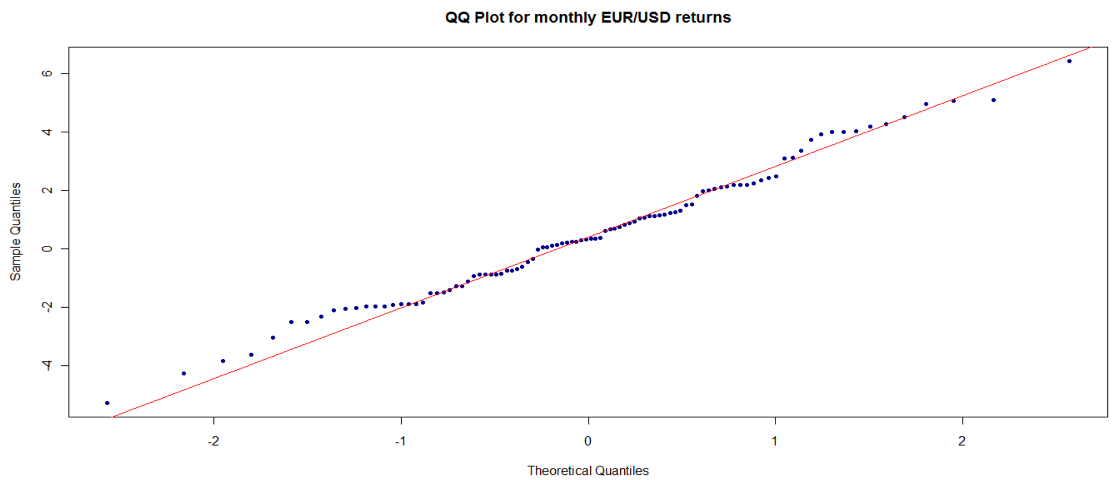
Variable	EUR/SEK	EUR/USD	USD/SEK
Short Term IRD	0.000	0.000	0.000
Long Term IRD (2 years)	0.000	0.000	0.000
Long Term IRD (5 years)	**0.020**	**-0.037**	**0.048**
Risk Appetite Measure, VIX	0.000	**0.0002**	0.000
Equity, STOXX Index	0.000	**-0.576**	0.000
Equity, OMSX Index	0.000	0.000	0.000
GDP, swgdpaqq Index	0.000	0.000	0.000
GDP, EUGNEMUQ Index	0.000	0.000	0.000
Inflation, SWCPMOM index	0.000	0.000	0.000
Inflation, ECCPEMUM Index	**-0.035**	0.000	0.000
Confidence variable, SWETSURV Index	0.000	0.000	0.000
Confidence variable, GRZEEUEX Index	**-0.0001**	**0.0002**	**-0.0003**
Sweden Money Supply	**-0.903**	**-10.002**	**2.327**
Money Supply, ECMSM2 Index	**-3.271**	**20.559**	**-12.415**
Lag1	0.000	**-0.325**	**-0.008**
Lag2	**-0.041**	**-0.014**	**-0.034**
Lag3	**-0.033**	**-0.014**	**-0.008**
Lag4	**0.012**	**0.028**	**0.004**
EUR/USD daily return	0.000	-	0.000
USD/SEK daily return	**-0.009**	**-0.420**	-
EUR/SEK daily return	-	**0.430**	0.000

Table 16. Index of the covariates

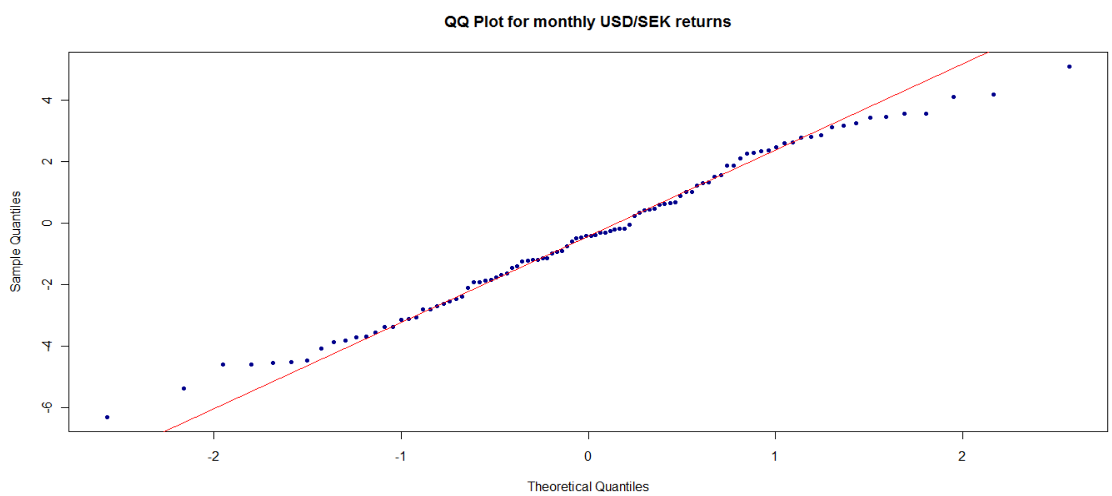
Index	Monthly Returns			Daily Returns		
	EUR/SEK	EUR/USD	USD/SEK	EUR/SEK	EUR/USD	USD/SEK
1	Short Term IRD	Short Term IRD	Short Term IRD	Short Term IRD	Short Term IRD	Short Term IRD
2	Long Term IRD (2 years)	Long Term IRD (2 years)	Long Term IRD (2 years)	Long Term IRD (2 years)	Long Term IRD (2 years)	Long Term IRD (2 years)
3	Long Term IRD (5 years)	Long Term IRD (5 years)	Long Term IRD (5 years)	Long Term IRD (5 years)	Long Term IRD (5 years)	Long Term IRD (5 years)
4	Confidence variable, SWETSURV	Confidence variable, SWETSURV	Confidence variable, SWETSURV	Confidence variable, SWETSURV	Confidence variable, SWETSURV	Confidence variable, SWETSURV
5	Confidence variable, GRZEEUEX	Confidence variable, GRZEEUEX	Confidence variable, GRZEEUEX	Confidence variable, GRZEEUEX	Confidence variable, GRZEEUEX	Confidence variable, GRZEEUEX
6	Risk Appetite Measure, VIX	Risk Appetite Measure, VIX	Risk Appetite Measure, VIX	Risk Appetite Measure, VIX	Risk Appetite Measure, VIX	Risk Appetite Measure, VIX
7	Equity, STOXX	Equity, STOXX	Equity, STOXX	Equity, STOXX	Equity, STOXX	Equity, STOXX
8	Equity, OMSX	Equity, OMSX	Equity, OMSX	Equity, OMSX	Equity, OMSX	Equity, OMSX
9	Inflation, SWCPMOM	Inflation, SWCPMOM	Inflation, SWCPMOM	GDP, swgdpaqq	GDP, swgdpaqq	GDP, swgdpaqq
10	Inflation, ECCPEMUM	Inflation, ECCPEMUM	Inflation, ECCPEMUM	GDP, EUGNEMUQ	GDP, EUGNEMUQ	GDP, EUGNEMUQ
11	Sweden Money Supply	Sweden Money Supply	Sweden Money Supply	Inflation, SWCPMOM	Inflation, SWCPMOM	Inflation, SWCPMOM
12	Money Supply, ECMSM2 Index	Money Supply, ECMSM2 Index	Money Supply, ECMSM2 Index	Inflation, ECCPEMUM	Inflation, ECCPEMUM	Inflation, ECCPEMUM
13	USD/SEK	EUR/SEK	USD/SEK	Sweden Money Supply	Sweden Money Supply	Sweden Money Supply
14	EUR/USD	USD/SEK	EUR/USD	Money Supply, ECMSM2 Index	Money Supply, ECMSM2 Index	Money Supply, ECMSM2 Index
15	GDP, swgdpaqq	GDP, swgdpaqq	GDP, swgdpaqq	USD/SEK	EUR/SEK	EUR/SEK
16	GDP, EUGNEMUQ	GDP, EUGNEMUQ	GDP, EUGNEMUQ	EUR/USD	USD/SEK	EUR/USD
17	Lag1	Lag1	Lag1	Lag1	Lag1	Lag1
18	Lag2	Lag2	Lag2	Lag2	Lag2	Lag2
19	Lag3	Lag3	Lag3	Lag3	Lag3	Lag3
20	Lag4	Lag4	Lag4	Lag4	Lag4	Lag4



(a)

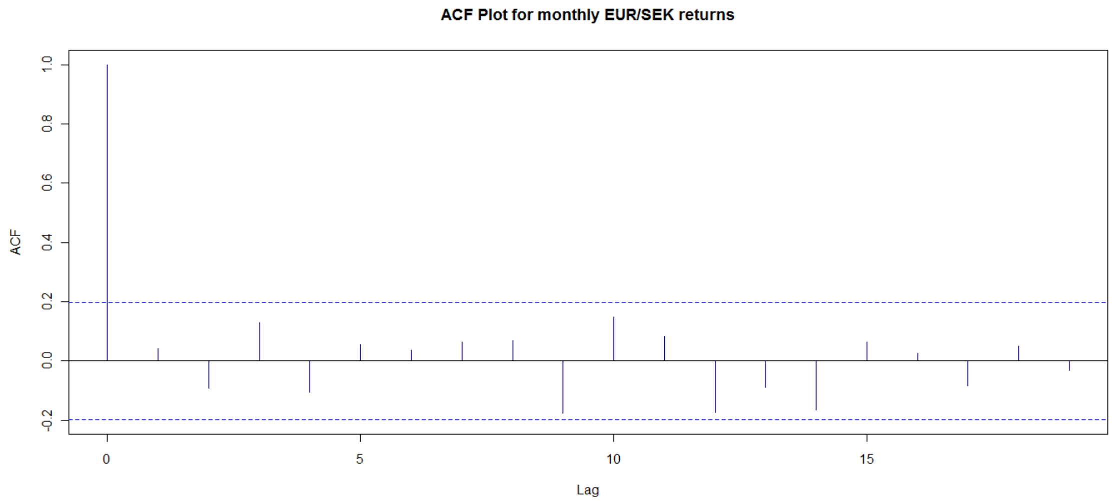


(b)

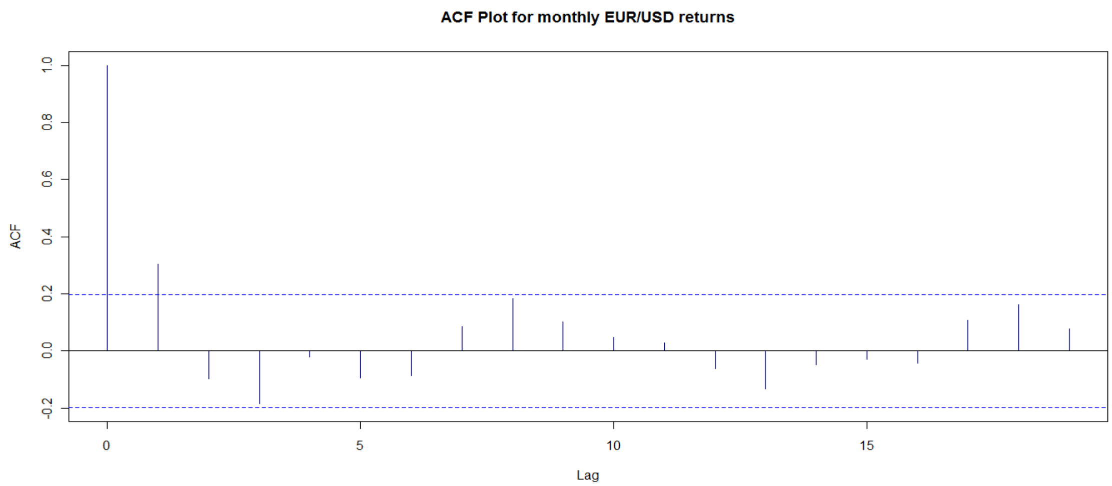


(c)

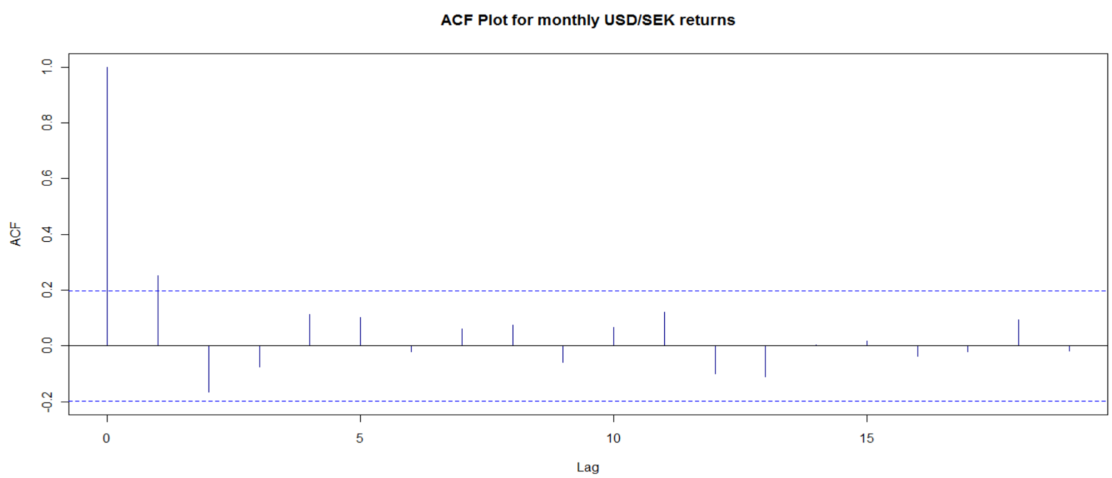
Figure 9. Normal probability plots of the monthly returns: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)

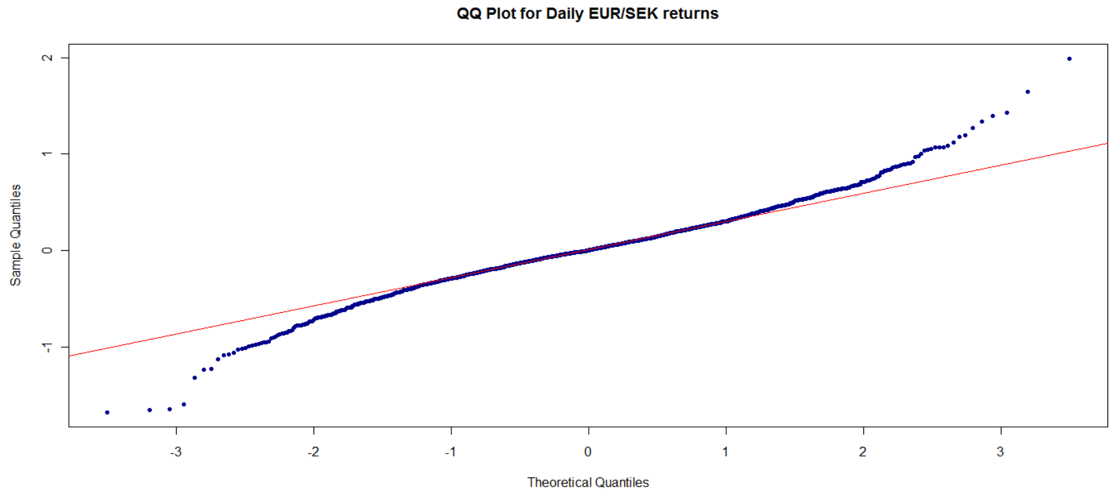


(b)

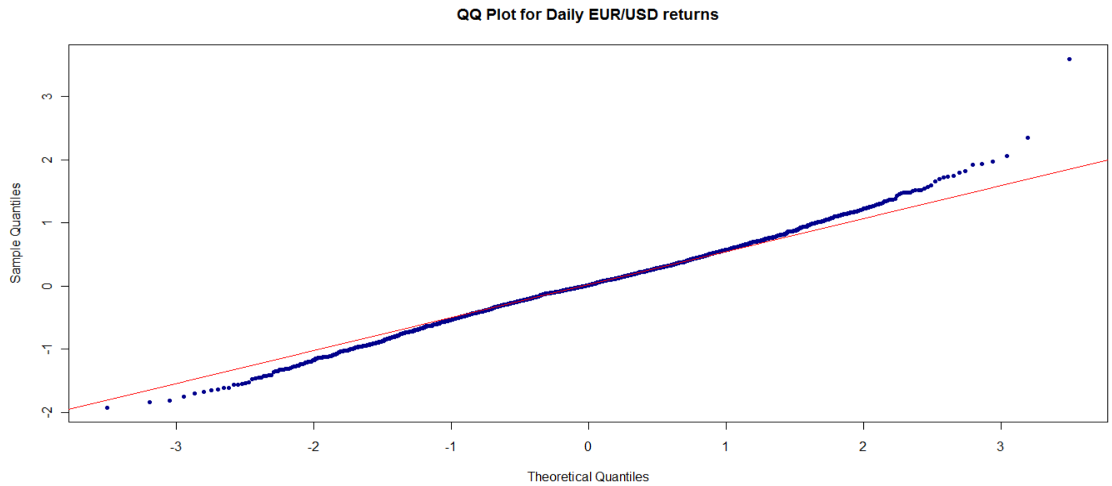


(c)

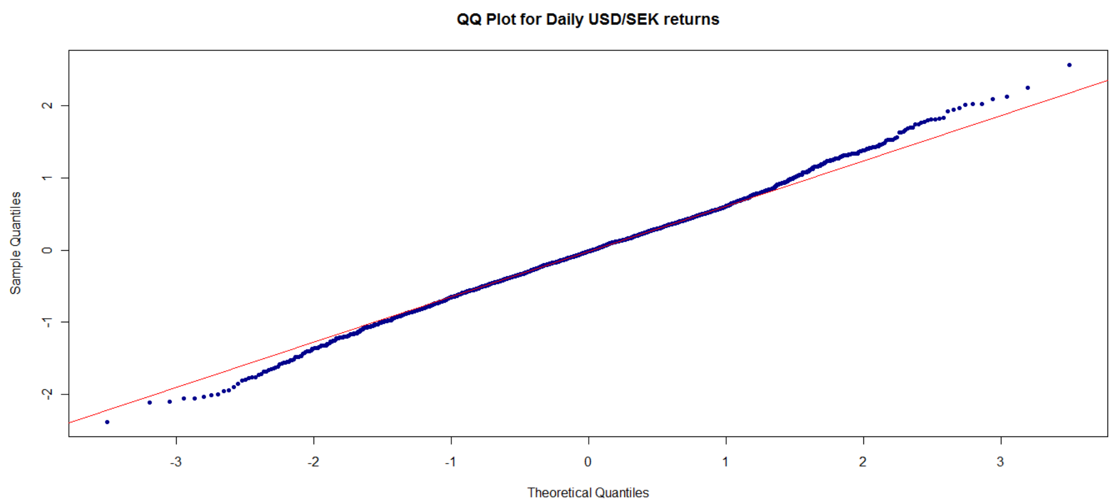
Figure 10. ACF plots for the monthly returns: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)

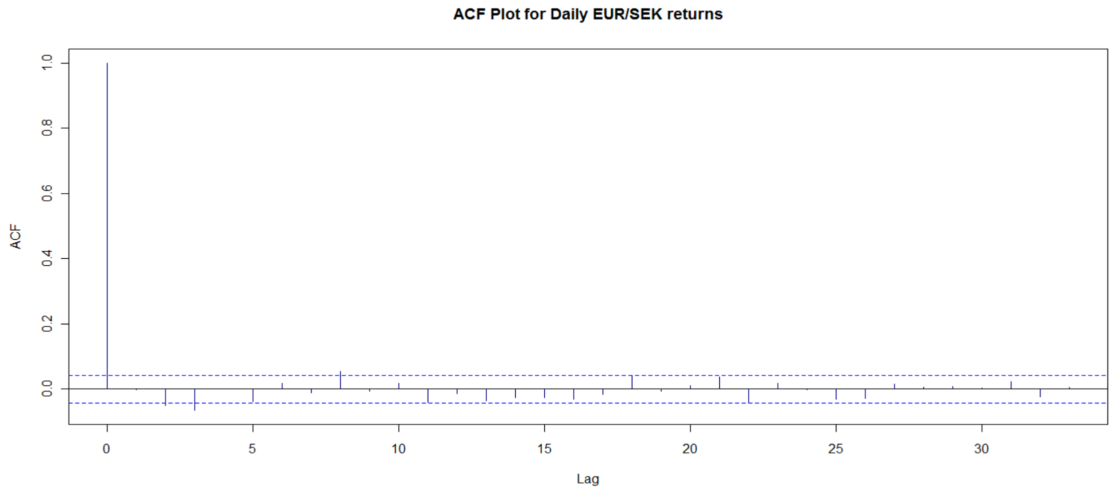


(b)

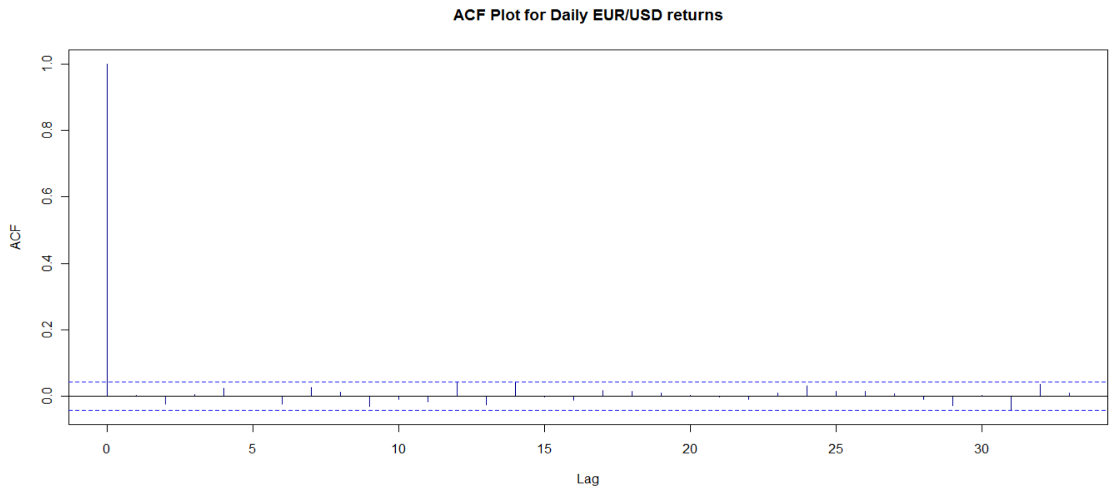


(c)

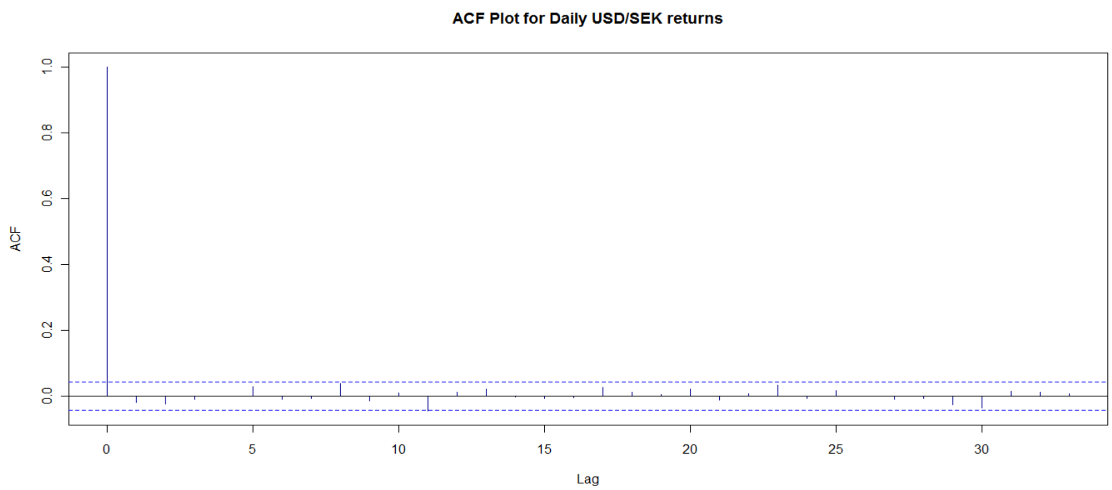
Figure 11. Normal probability plots of the daily returns: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)



(b)

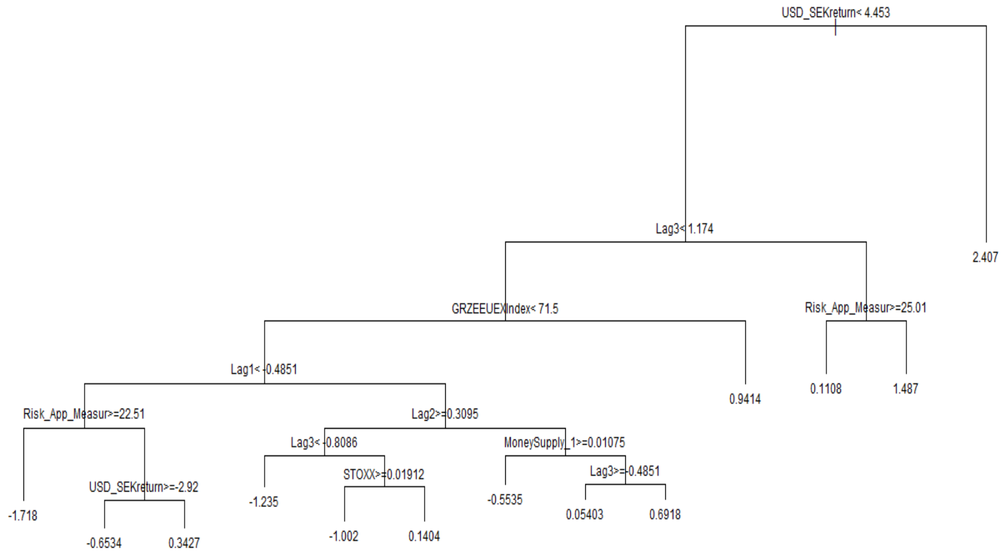


(c)

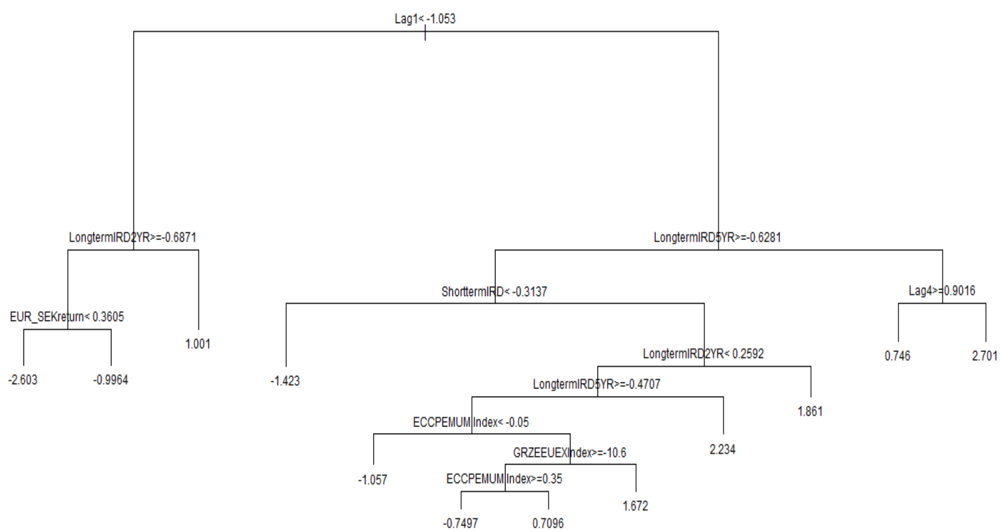
Figure 12. ACF plots for the daily returns: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK

Figure 13 shows the regression tree structure for the monthly returns.

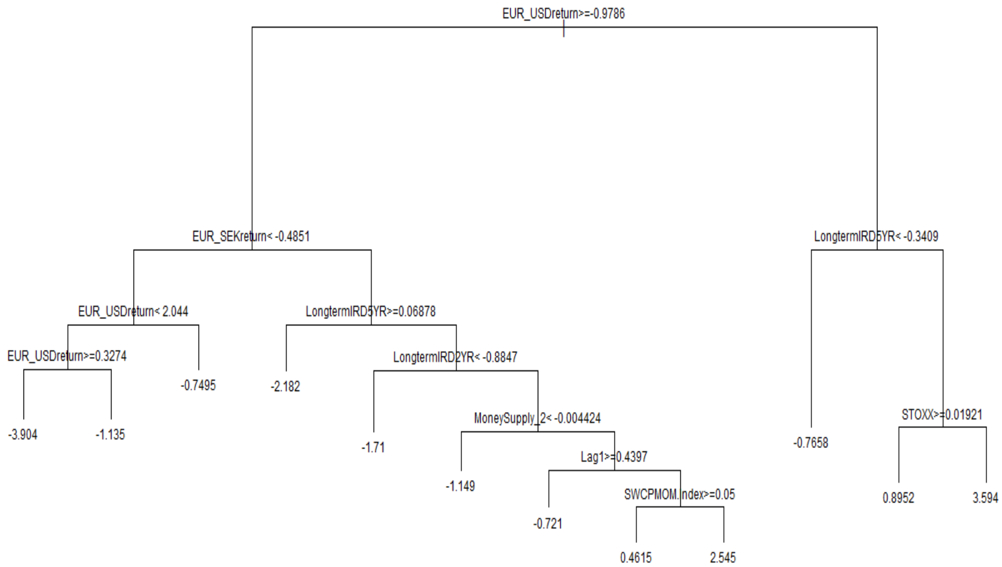
Figure 13. Regression tree: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)



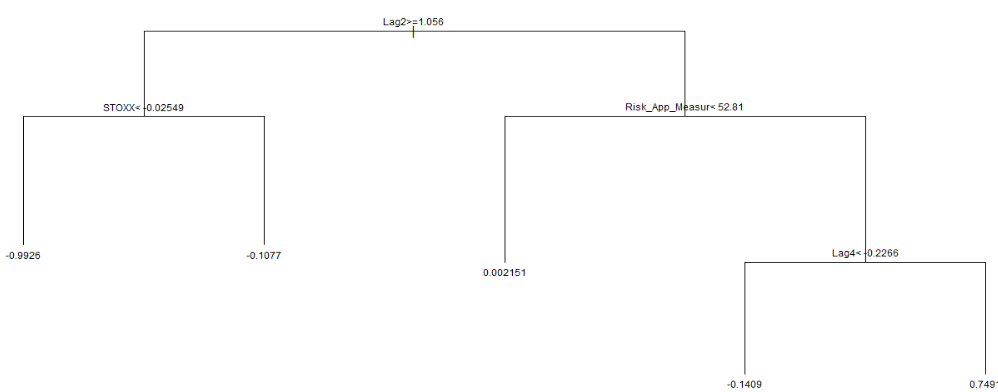
(b)



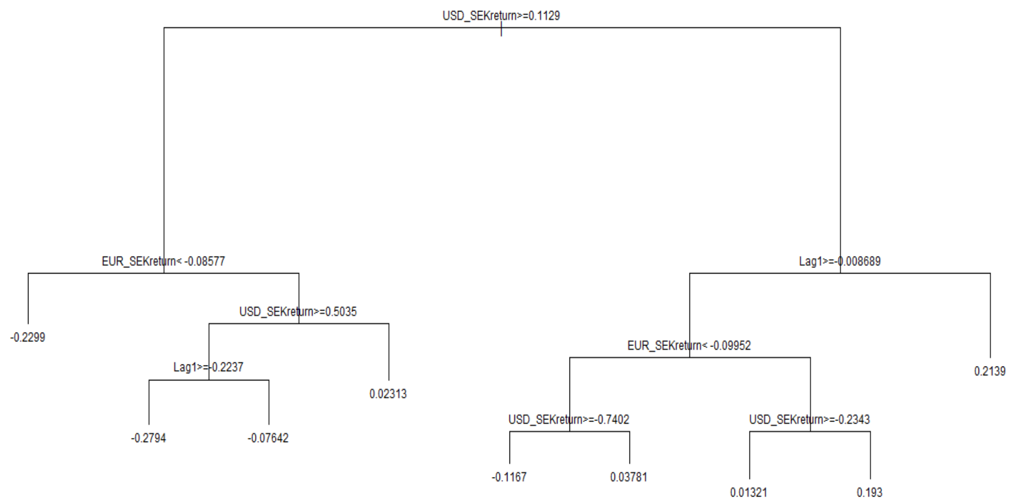
(c)

Figure 14 shows the regression tree structure for daily returns. The tree structure of EUR/SEK daily returns was constructed using default settings, while for other two pairs of currencies it was just a root node. In order to analyze variable importance, the tree was pruned using minsplit=500 and cp=0.001.

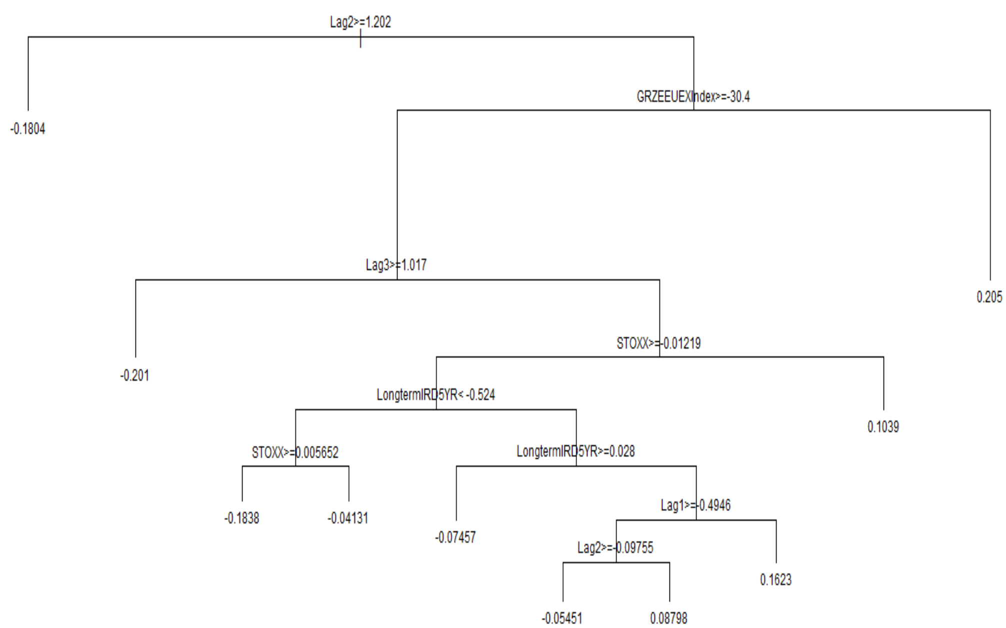
Figure 14. Regression tree: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)

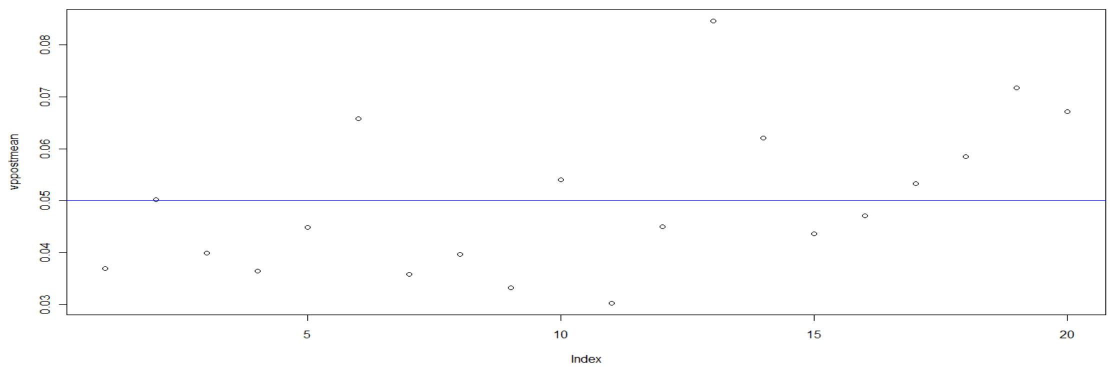

USD_SEKreturn>=0.1129
EUR_SEKreturn< -0.08577
Lag1>=-0.008689
-0.2299
USD_SEKreturn>=0.5035
EUR_SEKreturn< -0.09952
0.2139
Lag1>=-0.2237
0.02313
USD_SEKreturn>=-0.7402
USD_SEKreturn>=-0.2343
-0.2794
-0.07642
-0.1167
0.03781
0.01321
0.193

(b)

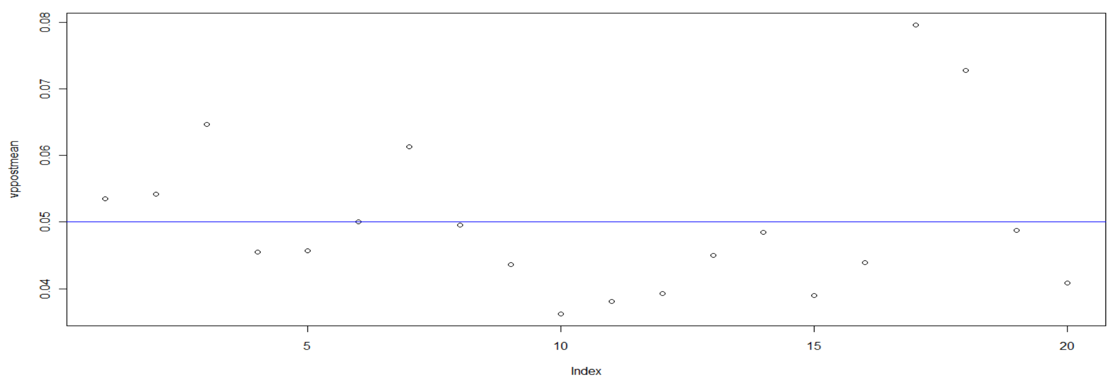

Lag2>=1.202
-0.1804
GRZEEUEXIndex>=-30.4
Lag3>=1.017
0.205
-0.201
STOXX>=-0.01219
LongtermIRD5YR< -0.524
0.1039
STOXX>=0.005652
LongtermIRD5YR>=0.028
-0.1838
-0.04131
-0.07457
Lag1>=-0.4946
Lag2>=-0.09755
0.1623
-0.05451
0.08798

(c)

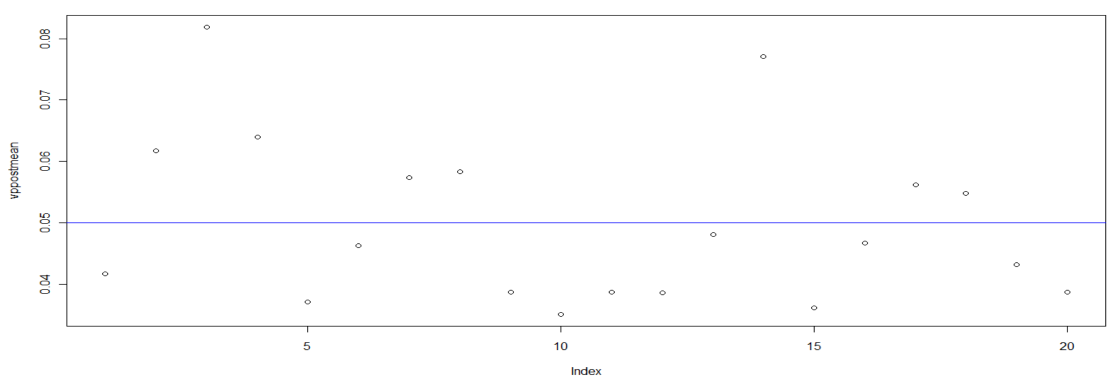
Figure 15 and 16 show vppostmean (variable percentage posterior mean) computed using BART for the monthly and the daily returns. The horizontal axis lists the index of the covariates used in the model which are listed in Table 16.



(a)

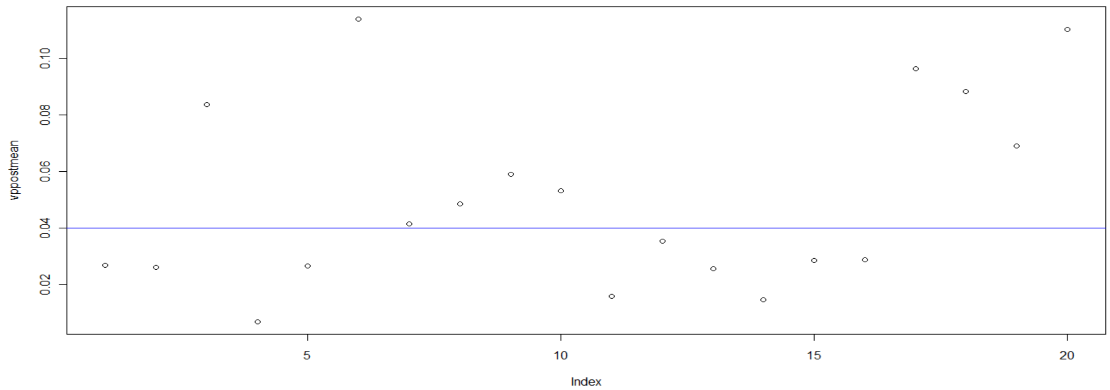


(b)

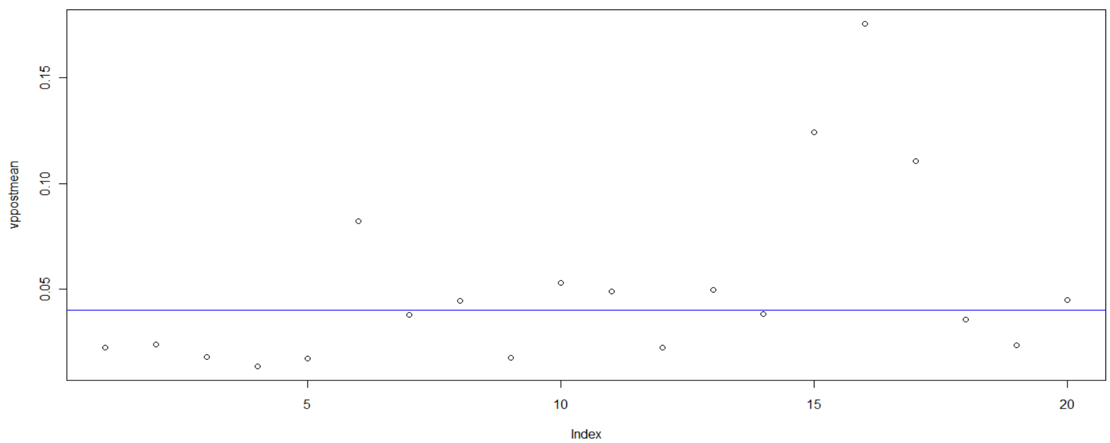


(c)

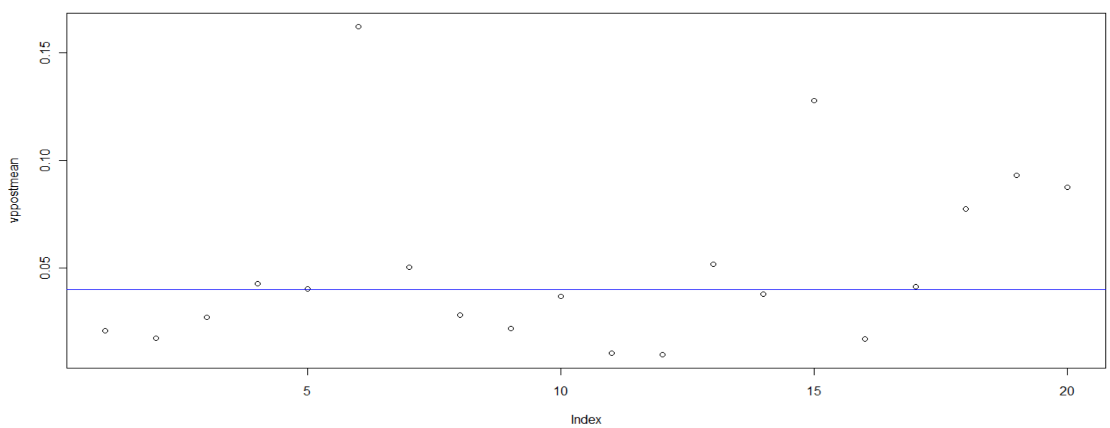
Figure 15. Variable percentage posterior mean using BART: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK



(a)



(b)



(c)

Figure 16. Variable percentage posterior mean using BART: (a) EUR/SEK (b) EUR/USD and (c) USD/SEK

8. Appendix B: R Code

The R code illustrating the implementation of the estimation procedure on EUR/SEK monthly return is given below. The code was modified accordingly for other returns and for daily frequency. It includes the function definitions for each of the machine learning models used.

```
library(tseries)
# For Constant Model
constant<-function(training,testing){
mean<-c()
error<-c()
pred<-c()
train<-training$EUR_SEKreturn
pred[1]<-mean(train)
errortrain<-training$EUR_SEKreturn-mean(train)
error[1]<-testing$EUR_SEKreturn[1]-pred[1]
j<-2
for(i in 1:(nrow(testing)-1)){
train<-append(train,testing$EUR_SEKreturn[i])
pred[j]<-mean(train)
error[j]<-pred[j]-testing$EUR_SEKreturn[j]
j<-j+1
}
res<-c(errortrain,error)  #residuals vector
# modelling residuals using ARCH model
r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)        # returns +/- the conditional standard deviation predictions

#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]
```

```
#modelling new model
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return
#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]
return(list(newtraining,newtesting,pred,error,estimatedsd))
}
```

#Autoregressive model

```
library(forecast)
autoreg<-function(training,testing){
train<-training$EUR_SEKreturn
test<-testing$EUR_SEKreturn
arimamodel1<-arima(train,order=c(1,0,0))
trainerror<-arimamodel1$residuals
p<-c()
p[1]<-(forecast.Arima(arimamodel1,h=1)$mean)
e<-c()
e[1]<-test[1]-p[1]
j<-2
for( i in 1:(length(test)-1)){
train<-c(train,test[i])
arimamodel1<-arima(train,order=c(1,0,0))
p[j]<-forecast.Arima(arimamodel1,h=1)$mean
e[j]<-test[j]-p[j]
j<-j+1
```

```
}

res<-c(trainerror,e)  #residuals vector

r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)       # returns +/- the conditional standard deviation predictions

#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return

#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
dim(newtesting)
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]
return(list(newtraining,newtesting,e,p,estimatedsd))
}
```

#Regression Tree

```
library(rpart)
tree<-function(training,testing){
training1<-training
```

```
treemodel<-
rpart(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD5YR+Ris
k_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+MoneySupply_1+MoneySupply_2+GRZEEUEXIndex+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,method="anova")
trainpred<-predict(treemodel)
errortrain<-training$EUR_SEKreturn-trainpred
predict<-c()
predict[1]<-predict(treemodel,new=testing[1,])
error<-c()
error[1]<-testing$EUR_SEKreturn[1]-predict[1]
j<-2
for( i in 1:(nrow(testing)-1)){
training1<-rbind(training1,testing[i,])
treemodel<-
rpart(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD5YR+Ris
k_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+GRZEEUEXIndex+MoneySupply_1+MoneySupply_2+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,method="anova")
predict[j]<-predict(treemodel,new=testing[j,])
error[j]<-testing$EUR_SEKreturn[j]-predict[j]
j<-j+1
}

res<-c(errortrain,error)  #residuals vector

r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)       # returns +/- the conditional standard deviation predictions
```

```
#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return

#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]

return(list(newtraining,newtesting,predict,error,estimatedsd))
}
```

#Random Forest

```
library(randomForest)
forest<-function(training,testing,x){
training1<-training
rf<-
randomForest(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD
5YR+Risk_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+MoneySupply_1+MoneySupply_2+GRZEEUEXIndex+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,mtry=x)
rfpred<-predict(rf)
errortrain<-training$EUR_SEKreturn-rfpred
predict<-c()
```

```
predict[1]<-predict(rf,new=testing[1,])
error<-c()
error[1]<-testing$EUR_SEKreturn[1]-predict[1]
j<-2
for( i in 1:(nrow(testing)-1)){
training1<-rbind(training1,testing[i,])
rf<-
randomForest(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD
5YR+Risk_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+MoneySupply_1+MoneySupply_2+GRZEEUEXIndex+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,mtry=x)
predict[j]<-predict(rf,new=testing[j,])
error[j]<-testing$EUR_SEKreturn[j]-predict[j]
j<-j+1
}

res<-c(errortrain,error)  #residuals vector

r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)        # returns +/- the conditional standard deviation predictions

#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return

#split into 70:30
```

```
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
dim(testing)
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]
return(list(newtraining,newtesting,predict,error,estimatedsd))
}
```

#Support Vector Regression

```
library(e1071)
support<-function(training,testing,name){
training1<-training
SVM<-
svm(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD5YR+Ris
k_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+MoneySupply_1+MoneySupply_2+GRZEEUEXIndex+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,type="eps-regression",
kernel=name)
SVMpred<-predict(SVM)
errortrain<-training$EUR_SEKreturn-SVMpred
predict<-c()
predict[1]<-predict(SVM,new=testing[1,])
error<-c()
error[1]<-testing$EUR_SEKreturn[1]-predict[1]
j<-2
for( i in 1:(nrow(testing)-1)){
training1<-rbind(training1,testing[i,])
```

```
SVM<-
svm(EUR_SEKreturn~ShorttermIRD+LongtermIRD2YR+LongtermIRD5YR+Ris
k_App_Measur+STOXX+OMSX_Index+swgdpaqq_index+
EUGNEMUQ.Index+SWCPMOM.Index+ECCPEMUM.Index+SWETSURVIndex
+MoneySupply_1+MoneySupply_2+GRZEEUEXIndex+Lag1+Lag2+Lag3+Lag4
+USD_SEKreturn+EUR_USDreturn,data=training1,type="eps-regression",
kernel=name)
predict[j]<-predict(SVM,new=testing[j,])
error[j]<-testing$EUR_SEKreturn[j]-predict[j]
j<-j+1
}
res<-c(errortrain,error)  #residuals vector

r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)        # returns +/- the conditional standard deviation predictions

#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return

#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
dim(testing)
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]
```

```
return(list(newtraining,newtesting,predict,error,estimatedsd))
}
```

#LASSO

```
library(lars)
lassomodel<-function(training,testing){
training1<-training
w<-which(names(training) %in% 'EUR_SEKreturn')
X<-training[,-w]
Y<-training[,w]
plasso<-c()
predict<-c()
lassomodel<-lars(as.matrix(X),as.matrix(Y),type="lasso")
lassopred<-predict(lassomodel,newx=X,type="fit")
o<-order(lassomodel$Cp)
trainerror<-Y-lassopred$fit[,o[1]]
plasso<-predict(lassomodel,newx=testing[1,-w],type="fit")
predict[1]<-plasso$fit[o[1]]
error<-c()
error[1]<-testing[1,w]-predict[1]
j<-2
for( i in 1:(nrow(testing)-1)){
training<-rbind(training,testing[i,])
X<-training[,-w]
Y<-training[,w]
lassomodel<-lars(as.matrix(X),as.matrix(Y),type="lasso")
o<-order(lassomodel$Cp)
plasso<-predict(lassomodel,newx=testing[j,-w],type="fit")
predict[j]<-plasso$fit[o[1]]
error[j]<-testing[j,w]-predict[j]
```

```
j<-j+1
}

res<-c(trainerror,error)  #residuals vector
# modelling residuals using ARCH model
r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)        # returns +/- the conditional standard deviation predictions

#scale response (return) variable
newdata<-rbind(training1,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
newscaleddata<-rbind(training1[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return



#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
dim(newtesting)

estimatedsd<-pred_sd[(integer1+1):n_newdata,1]
return(list(newtraining,newtesting,predict,error,estimatedsd))
}
```

#BART

```
library(BayesTree)
```

```
bartmodel<-function(training,testing){
w<-which(names(training) %in% 'EUR_SEKreturn')
xtrain<-training[,-w]
ytrain<-training[,w]
xtest<-rbind(xtrain[nrow(xtrain),],testing[1,-w])
predict<-c()
error<-c()
bartmodel<-bart(x.train=xtrain,y.train=ytrain,x.test=xtest)
predict[1]<-bartmodel$yhat.test.mean[length(bartmodel$yhat.test.mean)]
errortrain<-training$EUR_SEKreturn-bartmodel$yhat.train.mean
error[1]<-testing$EUR_SEKreturn[1]-predict[1]
j<-2
for( i in 1:(nrow(testing)-1)){
xtrain<-rbind(xtrain,testing[i,-w])
ytrain<-append(ytrain,testing[i,w])
xtest<-rbind(xtrain[nrow(xtrain),],testing[i,-w])
bartmodel<-bart(x.train=xtrain,y.train=ytrain,x.test=xtest)
predict[j]<-bartmodel$yhat.test.mean[length(bartmodel$yhat.test.mean)]
error[j]<-testing$EUR_SEKreturn[j]-predict[j]
j<-j+1
}
res<-c(errortrain,error)  #residuals vector

# modelling residuals using ARCH model
r<-garch(res,order=c(0,1)) # ARCH model
pred_sd<-predict(r)        # returns +/- the conditional standard deviation predictions
#scale response (return) variable
newdata<-rbind(training,testing)
new_return<-newdata$EUR_SEKreturn[-1]/pred_sd[-c(1),1]

#modelling new model
```

```
newscaleddata<-rbind(training[-1,],testing)
newscaleddata$EUR_SEKreturn<-new_return



#split into 70:30
integer1<-as.integer(nrow(newscaleddata)*0.7)
n_newdata<-nrow(newscaleddata)
newtraining<-newscaleddata[1:integer1,]
newtesting<-newscaleddata[(integer1+1):n_newdata,]
dim(newtesting)
estimatedsd<-pred_sd[(integer1+1):n_newdata,1]

return(list(newtraining,newtesting,predict,error,estimatedsd))
}
```

Avdelning, Institution
Division, Department

Division of Statistics, Department of Computer and Information Science

Datum
Date

29.06.2012

Språk
Language

☐Svenska/Swedish
☐Engelska/English

☐ Engelska/English

Rapporttyp
Report category

☐Licentiatavhandling
☐Examensarbete
☐C-uppsats
☐D-uppsats
☐Övrig rapport

☐ Examensarbete

ISBN_______________

ISRN_LIU-IDA/STAT-A--12/004—SE

Serietitel och serienummer ISSN
Title of series, numbering ______________

URL för elektronisk version

http://urn.kb.se/resolve?urn= urn:nbn:se:liu:diva-79053

Titel
Title
Forecasting exchange rates using machine learning models with time-varying volatility

Författare
Author
Ankita Garg

Sammanfattning
Abstract
This thesis is focused on investigating the predictability of exchange rate returns on monthly and daily frequency using models that have been mostly developed in the machine learning field. The forecasting performance of these models will be compared to the Random Walk, which is the benchmark model for financial returns, and the popular autoregressive process. The machine learning models that will be used are Regression trees, Random Forests, Support Vector Regression (SVR), Least Absolute Shrinkage and Selection Operator (LASSO) and Bayesian Additive Regression trees (BART). A characterizing feature of financial returns data is the presence of volatility clustering, i.e. the tendency of persistent periods of low or high variance in the time series. This is in disagreement with the machine learning models which implicitly assume a constant variance. We therefore extend these models with the most widely used model for volatility clustering, the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) process. This allows us to jointly estimate the time varying variance and the parameters of the machine learning using an iterative procedure. These GARCH-extended machine learning models are then applied to make one-step-ahead prediction by recursive estimation that the parameters estimated by this model are also updated with the new information. In order to predict returns, information related to the economic variables and the lagged variable will be used. This study is repeated on three different exchange rate returns: EUR/SEK, EUR/USD and USD/SEK in order to obtain robust results. Our result shows that machine learning models are capable of forecasting exchange returns both on daily and monthly frequency. The results were mixed, however. Overall, it was GARCH-extended SVR that shows great potential for improving the predictive performance of the forecasting of exchange rate returns.

Nyckelord
Keyword
Forecasting, exchange rates, volatility, machine learning models

LIU-IDA/STAT-A--12/004--SE