Big Data Impacts on Stochastic Forecast Models: Evidence from FX Time Series

Sebastian Dietz
Department of Business Administration and Economics
University of Passau, Germany
sjd@hotmail.de

Abstract

With the rise of the Big Data paradigm new tasks for prediction models appeared. In addition to the volume problem of such data sets nonlinearity becomes important, as the more detailed data sets contain also more comprehensive information, e.g. about non regular seasonal or cyclical movements as well as jumps in time series. This essay compares two nonlinear methods for predicting a high frequency time series, the USD/Euro exchange rate. The first method investigated is Autoregressive Neural Network Processes (ARNN), a neural network based nonlinear extension of classical autoregressive process models from time series analysis (see Dietz 2011). Its advantage is its simple but scalable time series process model architecture, which is able to include all kinds of nonlinearities based on the universal approximation theorem of Hornik, Stinchcombe and White 1989 and the extensions of Hornik 1993. However, restrictions related to the numeric estimation procedures limit the flexibility of the model. The alternative is a Support Vector Machine Model (SVM, Vapnik 1995). The two methods compared have different approaches of error minimization (Empirical error minimization at the ARNN vs. structural error minimization at the SVM). Our new finding is, that time series data classified as "Big Data" need new methods for prediction. Those new methods should be able to be "customized" to nonlinearity and other non-standard effects, which come along with increasing data volume and can not be standardized to be included in tradional time series models. Estimation and prediction was performed using the statistical programming language R. Besides prediction results we will also discuss the impact of Big Data on data preparation and model validation steps.

Keywords: FX prediction, High Frequency Data, Big Data Analytics, Autoregressive Neural Networks, Support Vector Machines, Computational Intelligence.

1. Introduction

The origins of the expression "Big Data" traces back to the turn of the millennium. It was first mentioned in the context of data mining (see Weiss and Indurkhya 1998) and econometrics (see Diebold 2000). It was basically motivated by the advancing technical development in storing and processing data and tries to summarize its influences on data analysis. The term describes a paradigm shift in data models which can be explained in the following: Increasing volume, variety and velocity (the 3 V's , mentioned in an unpublished research note at META Group from 2001) of data leads on the one hand to increasing missing or corrupted values, on the other hand also to noisier data streams. To counteract those tasks new data preprocessing and data analysis methods are necessary. One of their basic features should be to include nonlinearity which comes along with more frequent/noisier data.

Our aim is to investigate the out-of-sample prediction behavior of two time series models for a Big Data time series. The concept of the first model (Autoregressive Neural Network Processes, see section 4.1) is based on the standard assumptions of econometric time series process models extended by a generalized nonlinear part. The other (Support

Vector Machines, see section 4.2) is a concept from data mining and only recently used for regression and time series tasks. Other nonlinear models have been already applied at data with similar properties to the data used here. It was shown that nonlinear methods outperform traditional time series models (see e.g. Boreo and Marrocu 2002 and Leung et al. 2000 and many others). Such observation is only consistent to early theoretic discussions of nonlinear time series analysis rooted in Granger and Teräsvirta (1993). There the essential assumption is, if nonlinearity can be identified in the data, it has to be included in the models to avoid misspecification and improve predictive power. For this essay we follow the model building process of Box and Jenkins (1976): Data preparation, model estimation and model validation. Section 2 introduces the data, section 3 proceeds with initial data preparation steps (with a short discussion how big data properties can be identified during the data preparation phase), in section 4 the models are introduced, section 5 provides the estimation and forecast results. Section 6 summarizes the results.

2. The Data

The USD to Euro exchange rate is especially appropriate for econometric forecasting, as this is an exchange rate determined by a free market with high trade volume (the opposite is for example the USD to Chinese Yuan exchange rate). The data contain bid and ask prices and range from 2002-12-02 to 2007-11-30, in total 13,077,451 observations. The bid rate is the highest price a buyer of Euro is willing to pay in USD. It's opposite, the ask rate is the price in USD a seller of Euro is willing to accept. Fig. 1 shows the data in levels and first differences. High volatility including several outliers, concentrated in the years 2004 and 2005 and at the end of 2006 are observable. Graphics let us assume that the distance between measurement points is not completely equal (number of observations per day varies from 4,500 to 5,500). Therefore further analysis and manipulation is necessary.

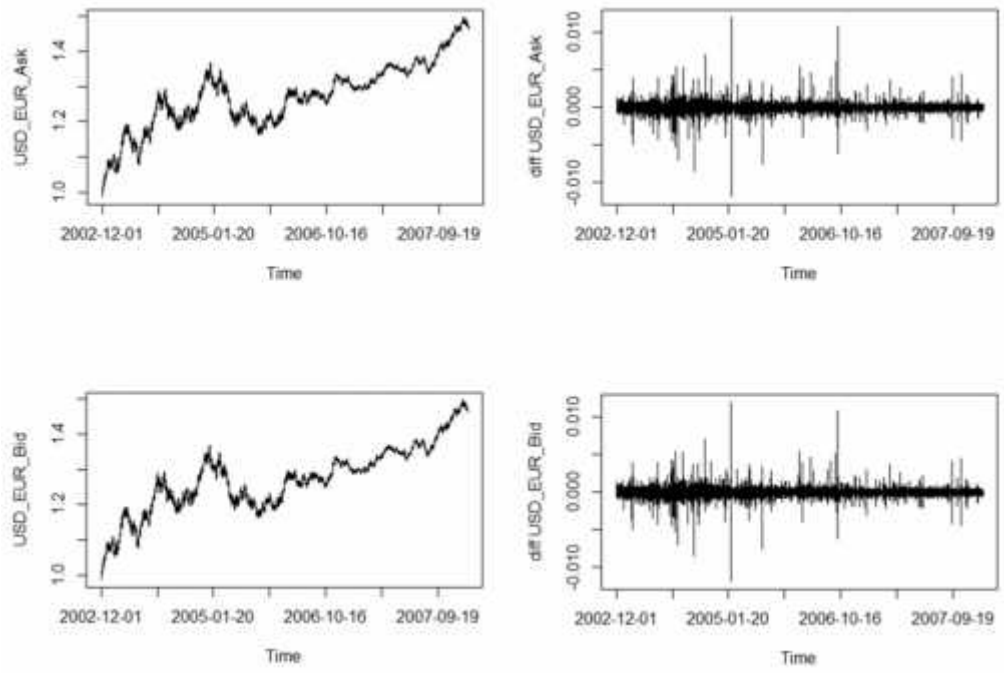

Fig. 1: Original data in levels and first differences

3. Data Preparation

Before we can start our model building procedure, we need to execute some stability and nonlinearity tests as well as some transformations of the initial data set. The non-equal distance between the daily observations makes it necessary to aggregate the data in a way to get a fix distance between the observations. Here the mean was calculated for each hour based on all observations available for this hour. This leads to a reduction to 31,987 observations. The aggregation was performed using a NoSQL database. Fig. 2 shows the merged data in levels and first differences. Alternatively aggregation on lower time distances (every quarter of an hour, every minute, etc.) is possible using the data for investigation. For the following reasons aggregation on hourly basis is sufficient: Firstly the results concerning the prediction behavior of the models are similar. The other reason is simply a computational one: Using numeric optimization algorithms means to compute large dimensional derivative matrices. Using standard hardware equipment this is not feasible with the whole data set if aggregated on shorter time distances. But reducing the series from more than 13 million to 32 thousands leads to one critical question: Are those time series still "Big Data"? A reply can be given on the one hand by Diebold (2000): "In the not-too distant future, we will be working with thousands of indicators, with many measured at daily or higher frequency". On the other hand especially the residual investigation in section 5 shows the properties of a Big Data or High Frequency time series: A α-stable distribution according to the Lévy generalized central limit theorem can be observed. Such is the case according to the general concept of central limit theorems if the number of observations is large enough. This observation is different from models and data typically used for econometric analysis where the distribution of the residuals is not be identified that obvious.

The STL (based on Loess smoother) decomposition of Cleveland et al. (1990) was used to investigate seasonality in the data. Seasonal decomposition rather should be performed at the data in levels to illustrate the seasonal influence which disappears after differentiation. The general principle of time series decomposition is to build a model of the data generating process as a function of previous observations or lags, if necessary a seasonal part and a stochastic part. With y_t denoting the observation at time t, a time series model can be described by the following equation:

$$y_t = f(y_{t-1},\ldots, y_{t-n}) + s_t + \varepsilon_t$$

$f(\cdot)$ denotes the trend function and s the seasonal part. Both are called predictable part. ε is the stochastic part. Fig. 3 and 4 show the results. The small grey column (in the very right side in the figures, marked by an arrow) represents the range of values of the seasonals.

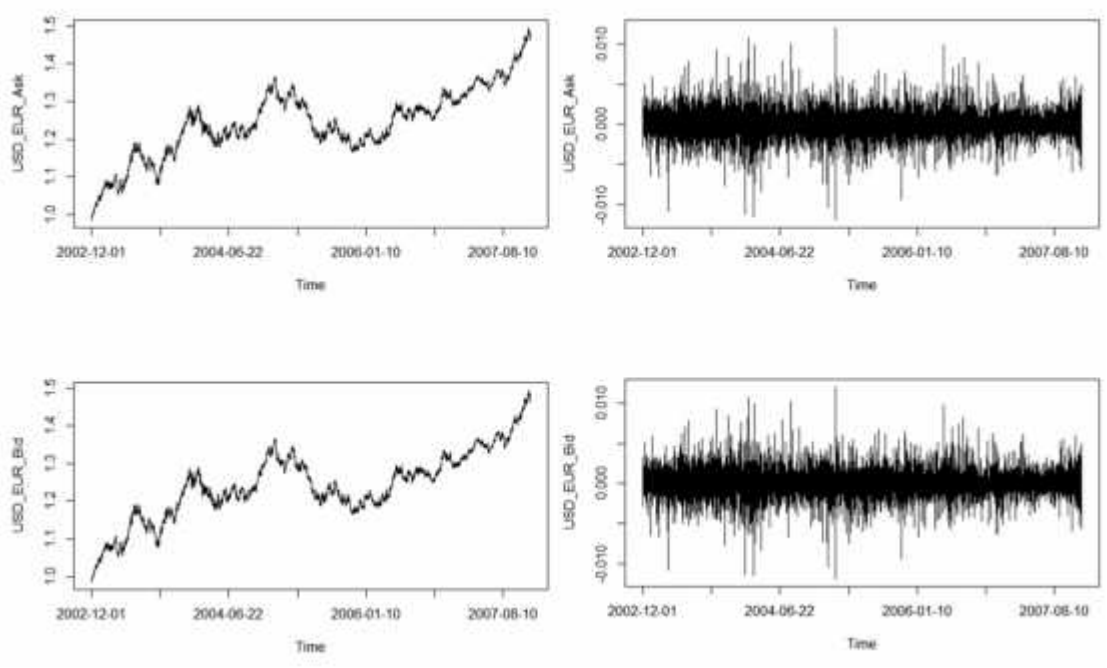

Fig. 2: Aggregated data in levels and first differences

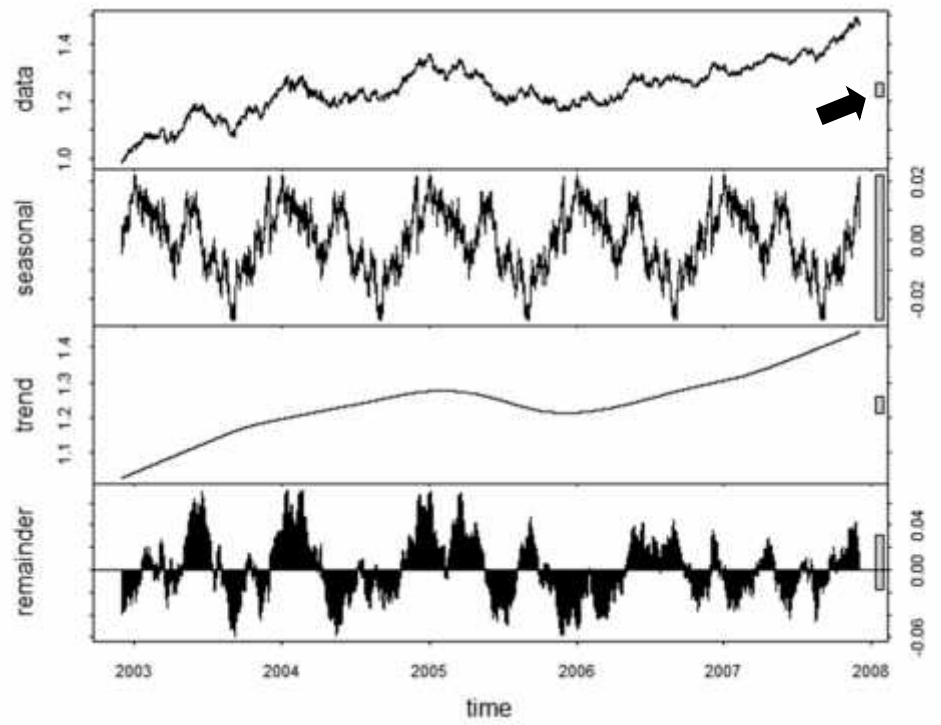

Fig. 3: STL seasonal decomposition USD/EUR exchange rate ask series

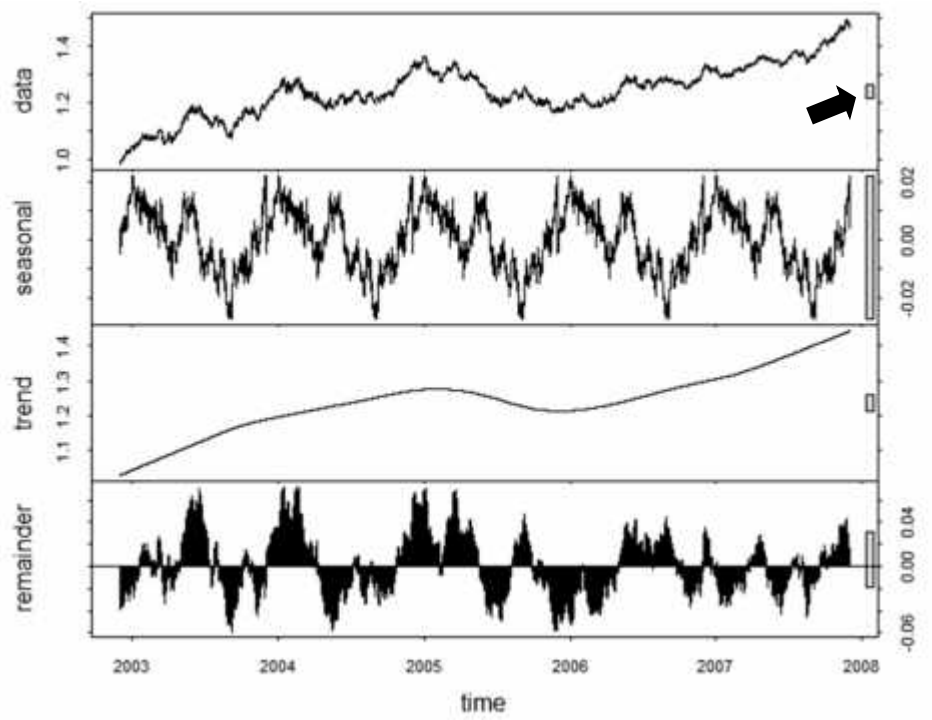

Fig. 4: STL seasonal decomposition USD/EUR exchange rate bid series

Note that the trend here is already represented by a nonlinear function (rather than a linear if we would use e.g. the Holt-Winters decomposition). Excluding the seasonal influence before modeling would mean to use only the trend plus remainder for the following modeling procedures and add the seasonal part again after model estimation.

To evaluate if decomposition is useful we investigate the size relation of the seasonal part and the remainder (or stochastic part) of the model. The range of values of the seasonals is only about 1/3 of the range of values of the residuals in this case or roughly about 1/10 of the range of total values. This means that the information contained in this seasonal part does not explain a significant part of the total observations. Subtracting the seasonal part as any other data manipulation also means loss of information (see e.g. Granger and Newbold, 1974). For this reason we refrain from it. The nonlinear models as we use them in the following sections should be also able to include permanent cylces (such as seasonalities) in the models. Considering the residuals for hourly observations and the residuals (from the same type of model) for higher aggregated series, we observe the more information available (the shorter the time distance between the observations), the more the time series models tend to explain the observations by the stochastic part (see also fig. 5 for a sketch of the idea). For example the series with hourly observations we use has an in-sample residual sum of squares (RSS) for the STL decomposition model of 22.32, whereas a series of aggregated daily observations has an in-sample RSS of only 1.09. This observation outlines a general problem of time series models: The more observations available, the more attention is paid to the residual terms represented by the stochastic part of the models. For example advanced time series process models which consider conditional heteroscedastic variance (the GARCH model of Bollerslev, 1986) or long memory (see Granger and Teräsvirta, 1993) just became popular as financial data recording and storage improved. In the Big Data age time series analysis - we will see

that below in documentation of the results in section 5 – modeling and prediction can no more be restricted on Gaussian errors but needs to keep in concept other error distributions to explain the stochastic part and keep the prediction results consistent with a theoretical model of the data generating process.

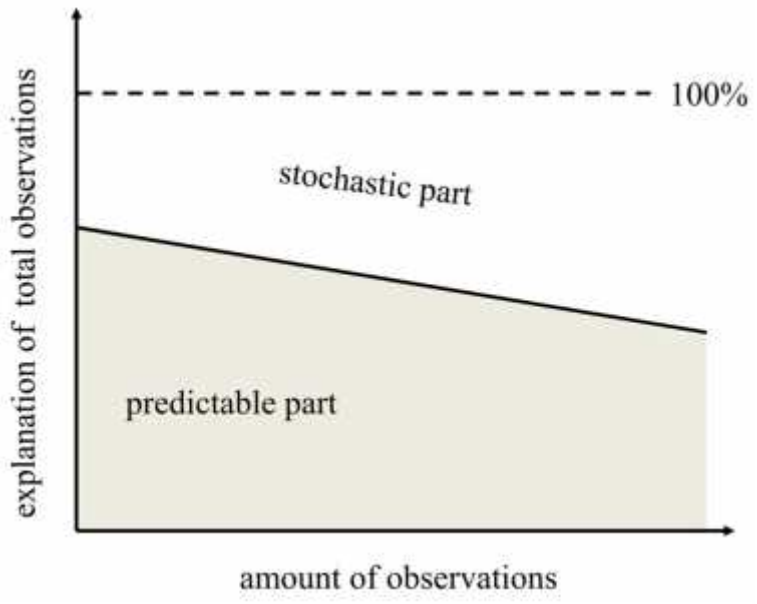

Fig. 5: "Big Data" problem of time series models

Now the degree of integration of the data has to be determined for estimation of the models. To investigate this, one of the well known unit root tests can be used, for example the test of Kwiatkowski et al. (1992), referred as KPSS test. Such tests indicate that the data for the two series are nonstationary and their first differences are stationary, table 1 shows the results. This means that they are integrated of order 1. For further procedure the first differences of the series are used.

Table 1: KPSS rest results p-Value

	USD_EUR_Ask	**USD_EUR_Bid**
Levels	<0.01	<0.01
1st Differences	>0.1	>0.1

Further it has to be ex-ante examined if a nonlinear part would significantly improve the econometric model or if a linear model is sufficient. The test of Teräsvirta, Lin and Granger (1993) can be used for this purpose: It uses a Taylor polynomial to approximate an unknown nonlinear part and examines if this reduces the residual sum of squares significantly. Table 2 shows the test results, which indicate strong nonlinearities for several lag orders. As calculating the polynomial takes much computational effort, the series of first differences were split into 5 parts of equal size. For each part the test was executed and then an average test statistic was calculated and compared to the 5 % critical value. The test indicates high nonlinearity for each lag here as the test statistic for each lag is significantly larger than the critical value.

Table 2: Results for nonlinearity test of Teräsvirta, Lin and Granger (1993)

Series	Lag	Chi-square test statistic	Critical value (95%)
USD_EUR_Ask	1	10.3056	5.9915
	2	58.8408	14.0671
	3	96.7758	26.2962
	4	119.9268	43.7730
	5	158.2158	67.5048
USD_EUR_Bid	1	12.0264	5.9915
	2	60.7532	14.0671
	3	96.4320	26.2962
	4	117.9712	43.7730
	5	158.4340	67.5048

4. The Prediction Models

4.1 Autoregressive Neural Network Process

In this section Autoregressive Neural Network Processes are introduced (see Dietz, 2011). At first we start with a linear univarite autoregressive process (AR). Let y_t be a variable representing a stochastic process depending of time t. A linear AR(n) model of this process is given by the following equation:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + ... + \alpha_n y_{t-n} + \varepsilon_t$$

ε_t represents i.i.d Gaussian distributed errors with expectation 0. The first n terms are called predictable part, as they can be determined by appropriate methods, whereas the last term is the stochastic or uncertain part, which accounts for unpredictable events. If nonlinearity is present in the data representing the process, the assumptions on the stochastic part are not appropriate, as it includes also the nonlinearity not captured by the predictable part. Therefore a nonlinear part is introduced step by step.

We will use a special kind of artificial neural networks, which have a universal approximation property. That means that they can approximate any function arbitrary precise. Those neural networks have a linear as well as a nonlinear part, have only one hidden layer and a bounded nonpolynomial activation function. In neural network terminology the expression layer means a set of variables: In the 3-layer network - used in the following - an input layer (the lags of y_t), a output layer (y_t) and a hidden layer (the nonlinear transformation of the lags of y_t via the nonlinear function g: $1 \rightarrow 1$) exist.

The hidden layer has to be investigated further: It consists of a linear part as well as of h so called neurons. Thus such a neural network can be interpreted as an augmented linear model: Because the linear part is not sufficient to explain the behaviour of the data, it is extended by h neurons. In each of them the input is at first transformed using a bias

weight γ_0 (a constant) and weights γ_1 to γ_n (for each lag). Now the nonlinear transformation is performed using the activation function $\varphi: 1 \rightarrow 1$, which is bounded and nonpolynomial. In the artificial neural network literature sigmoid functions like the tangens hyperbolicus or the logistic function are used, but of course others like the cosine or a threshold function are possible. Finally the results of the neurons are weighted by a scalar β and summarized with the linear part. The activation function is the same for all h neurons, whereas the weights have individual values which can be estimated using a nonlinear optimization algorithm. The ARNN is given by the following function:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \ldots + \alpha_n y_{t-n} + \sum_{i=1}^{h} \varphi(\gamma_{0i} + \gamma_{1i} y_{t-1} + \ldots + \gamma_{ni} y_{t-n}) \beta_h + \varepsilon_t$$

4.2 Support Vector Machines

Supervised learning SVM as we use them here were originally applied at classification issues. The concept can easily be transferred to regression as the task is the same, just in the one case the range of the output values is discrete and in the other continuous. The basic idea is mapping the nonlinear input space (space spanned by the vector of input values, which is here *n*-dimensional) into a high dimensional (*m*-dimensional) feature space. In the feature space a linear regression can be performed by adjusting a regression hyperplane between the values. Subsequently a linear regression is performed to bring the results from the feature space back to the output space (space spanned by the output values, here 1-dimensional). Doing this regression can be performed by linear separation of data which are highly nested in the input and output data space. In mathematical terms we can note: The input space is featured by the function φ into the output space, $\varphi: n \rightarrow m$, $m > n$. The weight vector $\boldsymbol{W}=[w_1,\ldots,w_m]^T$ is used to transform the feature space back to the output space, b is a scalar bias. The following equation shows the SVM time series model:

$$y_t = \boldsymbol{W}^T \varphi(\boldsymbol{Y}_t) + b + \varepsilon_t$$

$\boldsymbol{Y_t}=[y_{t-1},\ldots,y_{t-n}]^T$, ε_t represents the errors. As we do not use a least-squares minimization algorithm to estimate the equation, we have no assumptions concerning the error term. The ε-regression of Vapnik (1995) is used for estimation (in contrast to the concept of empirical risk minimization used for standard econometric time series process models and also for the ARNN here a concept of structural risk minimization is used: Errors larger than a predefined ε are not punished). For solving the optimization problem - which we will not discuss in detail here (for references see Vapnik 1995) – the cross products $\varphi(\boldsymbol{Y_t}) \times \varphi(\boldsymbol{Y_s})$ have to be calculated, which can be feasible in a high dimensional feature space. Therefore a kernel function $K(\cdot)$ is used to approximate the unknown nonlinear function $\varphi(\cdot)$. The important property of the kernel function is, that a feature space exists where $K(\cdot)$ is scalar. As kernels various alternatives can be used, the most common are linear, sigmoid, polynomial and Gaussian radial base functions. In section 5.1 a linear and a Gaussian radial base kernel are compared. The Gaussian radial base function is denoted as follows, with condition $\sigma > 0$:

$$K(\boldsymbol{Y}_t, \boldsymbol{Y}_s) = \exp\left(-\frac{\| \boldsymbol{Y}_t - \boldsymbol{Y}_s \|^2}{2\sigma^2} \right)$$

5. Results

5.1 The estimation and validation process

For estimation the data are split into two subsets: The estimation and the out-of-sample validation subset. As we deal with high frequency data here and we are interested in long term predictions, we choose 25.000 of the total data to be the estimation subset and the 6.986 observations to be the validation subset. Prediction is performed as rolling one-step-ahead out-of-sample prediction. The procedure is as follows: The model is estimated once, a prediction is made for one hour ahead, the values (lags) used for prediction are updated with the new observations and again prediction is made for one hour ahead. This is repeated until the end of the validation subset is reached.

For performance evaluation the estimated values are compared to the observations from the validation subset. As validation criterion the total hit rate (HIT) and Theil's inequality coefficient (THEIL) is used. HIT only compares the presign of the predicted returns with the observed ones. An improvement to a naïve forecast (just guessing the direction of the outcome) is in place if the total hit rate is higher than 50%. In practical application HIT can be relevant, if a trading strategy based on falling or increasing rates is executed (e.g. using Bonus and Discount certificates, not high risk derivatives).

THEIL compares the predicted results to a naive forecast, which is here the arithmetic mean of the time series (only of the estimation subset). A THEIL >1 means that the naive forecast is a better predictor than the estimated result. For a series y_t of length T the THEIL is calculated for a k-step prediction by:

$$THEIL = \sqrt{\frac{\sum_{t=T+1}^{k}(y_t - \hat{y}_t)^2}{\sum_{t=T+1}^{k}(y_t - \frac{1}{T}\sum_{i=1}^{T} y_i)^2}}$$

5.2 Results of the ARNN

An ARNN with 4 lags is calculated. Models are estimated for nonlinear parts with varying number of hidden neurons (h=1, h=5, h=10, h=20). The estimation of parameter values is done by a Levenberg-Marquardt algorithm (a quasi-Newton method). For avoiding overfitting an in-estimation-sample separation (99% estimation subset, 1% validation subset) is performed and the model with the best prediction power is selected. Results are shown in tables 3-6 and figure 6. At first additional hidden neurons seem to have no effect, as the in- and out-of-sample RSS does not change (tables 1 and 2). HIT and THEIL indicate, that the models are not appropriate for predictions. The performance is even lower than a random process model (HIT) or calculating the average (THEIL). The problem might be, that the nonlinear part is not able to handle the highly nonlinear structure of the data (indicated in table 2). Reason for this can be the dominance of the linear part in the model caused by inappropriateness of the parameter estimation algorithm (table 3 indicates this, as the in-sample RSS does not improve by adding

nonlinear terms). Our assumption is, that the problem of the algorithm in general is the least-squares (RSS minimizing) estimation. No better alternative than the linear model could be identified and the complex nonlinearity just could not be modeled (this could be due to a relatively symmetric distribution of the outliers). The out-of-sample residuals in fig. 6 show a Lévy- or α-stable behavior for some selected models (the class of distributions was introduced in Lévy 1925). In this class of distributions the shape of the distribution depends from a parameter $\alpha \in (0;2]$, which is 2 in the case of a Gaussian distribution (expectation=0 and finite variance) and 1 in the case of a Cauchy distribution (no expectation and infinite variance). The variance becomes already infinite, if $\alpha < 2$. If $1 < \alpha \leq 2$, the expectation is defined, otherwise the expectation is not defined. The results in fig. 6 seem to resemble more a Cauchy- than a Gaussian distribution (especially at the tails of the distribution).

Table 3: In-sample RSS for ARNN

Series	**h=1**	**h=5**	**h=10**	**h=20**
USD_EUR_Ask	0.05918	0.05918	0.01805	0.05918
USD_EUR_Bid	0.05905	0.05905	0.01801	0.03902

Table 4: Out-of-sample RSS for ARNN

Series	**h=1**	**h=5**	**h=10**	**h=20**
USD_EUR_Ask	0.01115	0.01115	0.00597	0.01115
USD_EUR_Bid	0.01115	0.01115	0.00595	0.00889

Table 5: Out-of-sample HIT (total hit rate) for ARNN

Series	**h=1**	**h=5**	**h=10**	**h=20**
USD_EUR_Ask	0.51217	0.51217	0.50558	0.51217
USD_EUR_Bid	0.51074	0.51074	0.50387	0.50802

Table 6: Out-of-sample Theil inequality coefficient for ARNN

Series	**h=1**	**h=5**	**h=10**	**h=20**
USD_EUR_Ask	1.44117	1.44117	1.05458	1.44117
USD_EUR_Bid	1.44391	1.44391	1.05488	1.28879

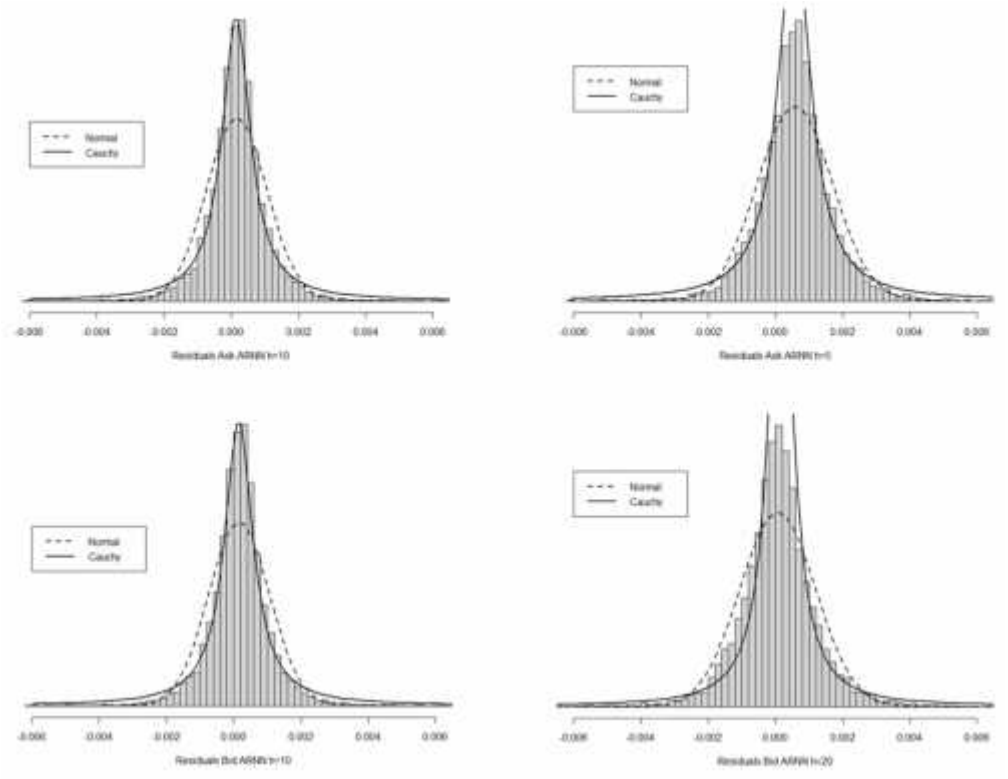

Fig. 6: Out-of-sample residuals ARNN

5.3 Results of the SVM

Two SVM models with two kernel functions are estimated: A radial basis kernel and a linear kernel for comparison. The fitted and predicted values have to be scaled to the range of values of the observations. Results are shown in tables 7 – 10 and fig. 7 – 9. In- and out-of-sample RSS seems far better than at the ARNN, also the THEIL, which is <1 for all models. However, concerning the out-of-sample RSS and the THEIL, a linear kernel seems to be sufficient. Things changes if HIT is considered: Here the radial basis kernel seems to produce stable results >50%. This could be due to the ability of handling nonlinear movements. The nonlinear structure of the models is shown in fig. 7: The predictions are plotted based on the previous hour's prediction. Plotting this is basically wrong as in reality the predictions are based on the previous period actuals. However, the series here are plotted against the actuals (grey line). Only by such a plot a difference between predicted and actuals (black line) is observable. The figure shows a much more nonlinear dynamic in the radial base kernel predictions than in the linear kernel predictions. To show the correct predictions, the first 20 predicted values (continuous line) are plotted against the actuals (points) in fig. 8. Fig. 9 shows the out-of-sample residuals. Results are very similar to those from the previous section, but here we have not induced Gaussian distribution. On a first view the range of values of the residuals is slightly below the residuals from the ARNN. Also the kurtosis seems to resemble a little bit more to a Gaussian distribution. However additional investigation on how to overcome the problem of infinite variance has to be done if such SVM are used for point prediction (and not only HIT, which seem to be the only manageable performance indicator because of the error distribution).

Table 7: In-sample RSS for SVM

Series	Linear kernel	Radial kernel
USD_EUR_Ask	0.03395	0.03790
USD_EUR_Bid	0.03810	0.04312

Table 8: Out-of-sample RSS for SVM

Series	Linear kernel	Radial kernel
USD_EUR_Ask	0.00501	0.00505
USD_EUR_Bid	0.00501	0.00505

Table 9: Out-of-sample HIT (total hit rate) for SVM

Series	Linear kernel	Radial kernel
USD_EUR_Ask	0.56799	0.57028
USD_EUR_Bid	0.51059	0.57315

Table 10: Out-of-sample Theil inequality coefficient for SVM

Series	Linear kernel	Radial kernel
USD_EUR_Ask	0.93347	0.96963
USD_EUR_Bid	0.93542	0.94141

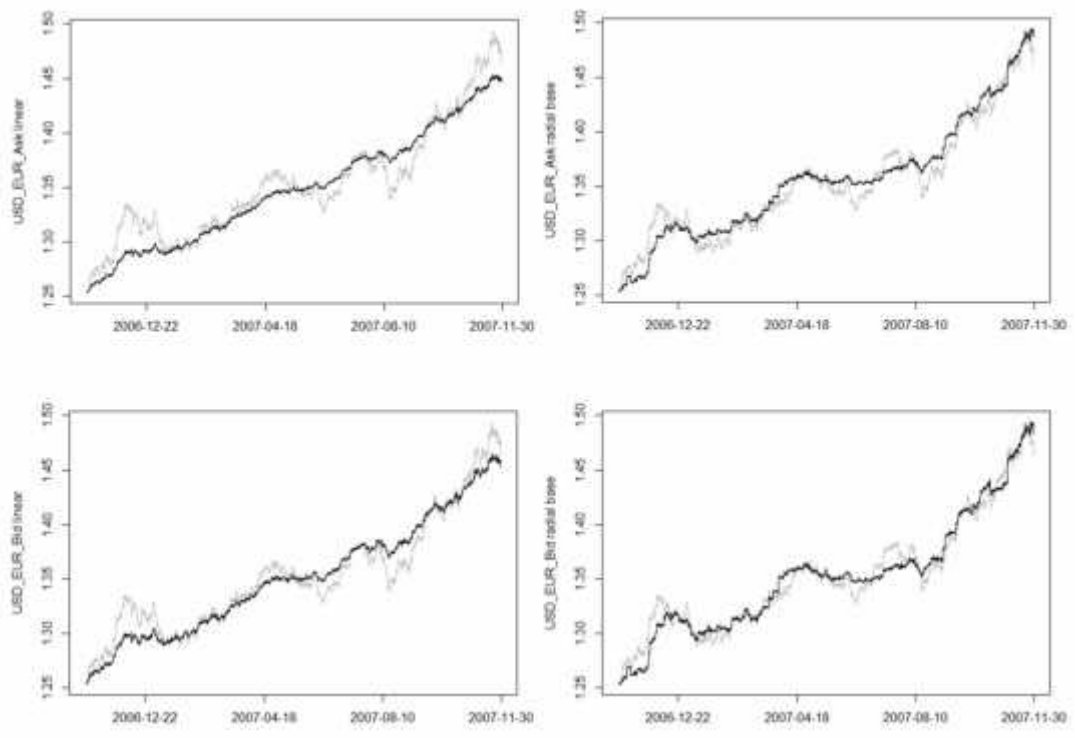

Fig. 7: Out-of-sample prediction SVM vs. actual: Predicted values are based on previous period predictions (not on previous periods actuals) to outline the difference to the original values

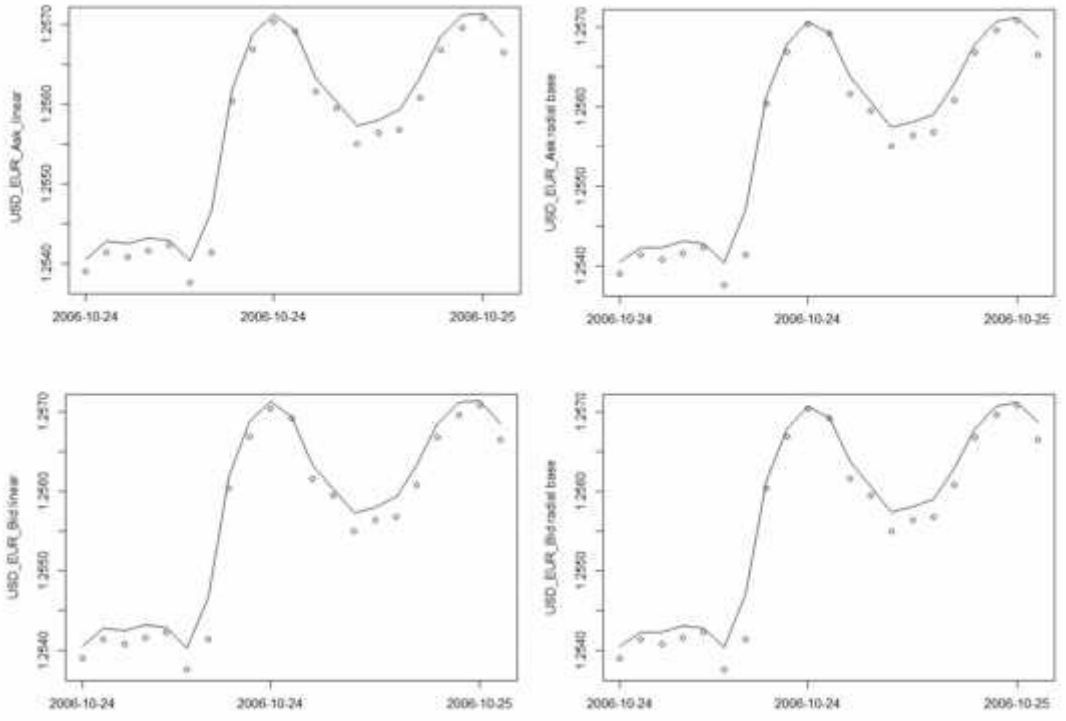

Figure 8: Out-of-sample prediction SVM vs. actual for the first 20 observations: Predicted values are based on previous actuals (correct visualization)

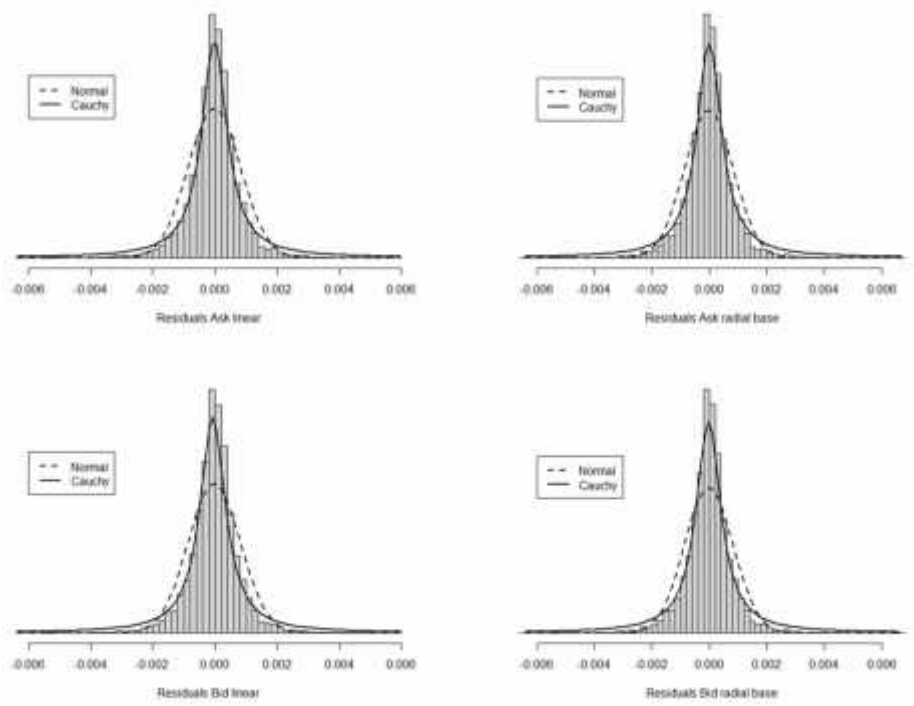

Figure 9: Out-of-sample residual distribution SVM

6. Conclusion

We have investigated the impact of the phenomenon Big Data on time series prediction models using hourly observations of the USD/EUR exchange rate. Two models were compared concerning out-of-sample rolling 1-step prediction. Two similar series (Bid and Ask) as well as about 7,000 observations have been used for comparing out-of-sample predictions to actual observations. This amount of data used supports the representativeness of our investigation. The two methods used are set up with different optimization algorithms (empirical error minimization in the case of ARNN, structural error minimization in the case of SVM). Whereas ARNN follows the concept of classical time series analysis, SVM is an originally data mining method and basically assumption-free. We observe for the Big Data time series the ARNN is not appropriate as a prediction model as it is not able to detect the nonlinearity of the data. The SVM provides acceptable prediction results, whereas the errors have infinite variance. Therefore the SVM is only appropriate for prediction the total hit rate (the correct presign of the values to predict). Investigation of the errors and how to exclude/reduce nonlinearites from them and how to handle the distribution of the errors (which are α-stable but not normal) should be a topic for further investigation.

References

1. Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics, 31: 307-327.
2. Boreo, G. and Marrocu, E. (2002). The Performance of Non-Linear Exchange Rate Models: A Forecasting Comparision. Journal of Forecasting 21: 513-542.
3. Box, G.E.P. and Jenkins, G.M. (1976). Time Series Analysis - Forecasting and Control. Holden Day, San Francisco.
4. Cleveland, R.B., Cleveland, W.S., McRae, J.E. and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics 6: 3–73.
5. Diebold, F.X. (2000). Big Data Dynamic Factor Models for Macroeconomic Measurement and Forecasting. Discussion Read to the Eight World Congress of the Econometric Society, Seattle.http://www.ssc.upenn.edu/~fdiebold/papers/paper40/temp-wc.PDF
6. Dietz, S. (2011). Autoregressive Neural Network Processes - Univariate, Multivariate and Cointegrated Models with Application to the German Automobile Industry. Dissertation University of Passau.
7. Granger, C.W. and Newbold, P. (1974). Spurious Regression in Econometrics. Journal of Econometrics, 2: 110-120.
8. Granger, C.W.J. and Teräsvirta, T. (1993). Modelling Non-Linear Econometric Relationships. Oxford University Press, Oxford.
9. Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2: 359-366.
10. Hornik, K. (1993). Some New Results on Neural Network Approximation. Neural Networks 6:1069-1072.

11. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P. and Shin, Y. (1992). Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root. Journal of Econometrics, 54: 159–178.
12. Leung, M.T., An-Singh, C. and Hazem Daouk (2000). Forecasting Exchange Rates using General Regression Neural Networks. Computers & Operations Research 27: 1093-1110.
13. Lévy, P. (1925). Calcul des Probabilités. Gaulthier-Villars, Paris.
14. Teräsvirta, T., Lin. C.F. and Granger, C.W. (1993). Power of the Neural Network Linearity Test. Journal of Time Series Analysis 14: 209-220.
15. Vapnik, V.N. (1995). The Nature of Statistical Learning Theory, Springer, New York.
16. Weiss, S.M. and Indurkhya (1998). Predictive Data Mining: APractical Guide, Morgan Kaufmann Publishers, San Francisco.