ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΕΙΔΙΚΕΥΣΗΣ ΣΤΗ ΣΤΑΤΙΣΤΙΚΗ

FINANCIAL FORECASTING USING MACHINE LEARNING

Anna-Maria G. Klada

ΕΡΓΑΣΙΑ

Που υποβλήθηκε στο Τμήμα Στατιστικής

του Οικονομικού Πανεπιστημίου Αθηνών

ως μέρος των απαιτήσεων για την απόκτηση

Μεταπτυχιακού Διπλώματος Ειδίκευσης στη Στατιστική(Full-time).

Αθήνα

Σεπτέμβριος 2018


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

To my wonderful parents,

Iakovos and Evangelia


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my supervisor, Professor Ioannis D.Vrontos, who gave me the opportunity to deal with this interesting topic. Also, I feel very pleased from the academic staff and secretary from Msc in Statistics, for their professionalism and care for my studies. I owe an additional thank to doctoral candidate Karavida Elina, who, with her patience motivated me to improve the structure of my thesis.

I was also lucky to have by my side my friends from Rhodes, the place of my origin, and my colleagues, who were encouraging me every time I needed their help.

I want to include last the most important people I have in my life. These two people are my mother and my father, who always put me and my needs first, even if it was difficult sometimes for them. During my school and university years they helped me both, materially and morally, and without them I could not reach this level in my studies.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

VITA

My name is Anna-Maria Klada and I come from Rhodes. I finished high school the year 2011 and I succeeded to enter the Mathematics Department of National and Kapodistrian University of Athens the same year. During the third semester of my studies, I realized how much interested I was for the courses of Statistics and Operational Research. So, after completing my Bachelor's Degree in Mathematics, i applied for the Msc in Statistics of Athens University of Economics and Business. I have completed successfully the courses and learned a lot of new things about areas that I was not familiar before. The fields that i am more interested about are, the Financial Econometrics, the Statistical Quality Control and the Statistical Learning.

I have been working for one year as a professor of Mathematics for high school students, and this experience has empowered me with patience. I would like to encounter with a job including Statistics, to evolve my knowledge in this topic.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ABSTRACT

Anna-Maria Klada

ENGLISH TITLE

September 2018

Financial forecasting is an extraordinary issue. Hedge funds are companies that bind investors' money for a while, and are trying to raise their capital. In order for investors to benefit from this technique, it is good to know the performance of hedge funds.

The issue that we are assigned to, is the monthly prediction for 10 hedge fund returns, which are time series, and our forecasts last for 24 months. In our data there are 15 risk factors, from which we are called upon to decide which ones are important for our forecasting. The models we develop are from Machine Learning and some of them have been involved in finance. What we are concerned with is to compare these models, with traditional economic series prediction models, such as ARMA models and multiple regression models.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΕΡΙΛΗΨΗ

Άννα-Μαρία Κλαδά

ΕΛΛΗΝΙΚΟΣ ΤΙΤΛΟΣ

Σεπτέμβριος 2018

Η πρόγνωση στα χρηματοοικονομικά στοιχεία είναι ένα εξαιρετικό ζήτημα. Τα *hedge funds* ή αλλιώς *αντισταθμιστικά κεφάλαια* είναι εταιρείες οι οποίες δεσμεύουν για ένα διάστημα λεφτά επενδυτών, και με αυτό τον τρόπο, οι εταιρείες προσπαθούν να αυξήσουν το κεφάλαιό τους. Προκειμένου οι επενδυτές να επωφεληθούν με αυτή την τεχνική, είναι καλό να γνωρίζουν την απόδοση των hedge funds.

Το θέμα το οποίο μας έχει ανατεθεί είναι η μηνιαία πρόβλεψη για 10 hedge fund αποδόσεις, οι οποίες αποτελούν μια χρονολογική σειρά, και οι προβλέψεις μας εκτείνονται για 24 μήνες. Στα δεδομένα μας υπάρχουν 15 παράγοντες κινδύνου, από τους οποίους καλούμαστε να αποφασίσουμε ποιοι είναι σημαντικοί για την πρόβλεψή μας. Τα μοντέλα που θα αναπτύξουμε είναι από την περιοχή του Machine Learning και κάποια από αυτά έχουν απασχολήσει επιστήμονες στο χώρο των οικονομικών. Αυτό που μας ενδιαφέρει, είναι να συγκρίνουμε αυτά τα μοντέλα, με τα παραδοσιακά μοντέλα πρόβλεψης οικονομικών σειρών, όπως τα ARMA μοντέλα και τα μοντέλα πολλαπλής παλινδρόμησης.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

CONTENTS **Page**

LIST OF TABLES

LIST OF FIGURES


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

INTRODUCTION

The purpose of a financial forecast is to evaluate the current and future economic conditions to derive policy and programmatic decisions. A financial forecast is a management tool that helps us, identify the expected returns and trends of economic assets. An effective forecast allows us to make well directed decisions . For example, an investor should predict equity market sentiments before investing in stocks. The prediction of equity market and stock movements is not an easy part. Though, the financial analysts are developing, since the last decades, methods that could predict market movements.

The structure of this dissertation includes the *Introduction,* where we present the main subject, the goals that we expect to achieve and a little explanation about the methodology we have followed. The chapters that are analysing the main subject are five. In *Chapter 1,* there is a presentation about the efforts of researchers around the world, to develop efficient techniques for forecasting. In *Chapter 2*, we use exploratory analysis for our data. We have hedge fund data, and they have some special characteristics and properties we should take into account. In *Chapter 3*, there is a theoretical elaborative analysis about the methods used. We introduce also the Machine Learning. The formal definition, the types of algorithms and the importance of the Machine Learning field. In *Chapter 4*, we move to an application of the methods we referred at the previous chapter, using the data, to predict the hedge fund returns. Finally, we reach at *Chapter 5*, where we summarize the results occurring from this application.

The target of this thesis is, to investigate the predictability of the hedge fund indices by using different methods. We want also to find out also which factors influence our variables of interest. We have collected sources concerning this subject from the literature, and we have tried to implement some of these methods at the practical part of the thesis.

At the end, there are all the *References* we have used to develop this work, collected from the bibliography about this topic, such as scientific papers, books and websites. Also, at the final pages there is the *Appendix,* for more details about the data and the methods.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 1

Review in Financial Forecasting

1.1 Efforts for financial forecasting

Financial forecasting is a task that has been very active over the last decades and is defined as the prediction about future values of the data. It is important to predict economic trends to make decisions concerning a business or an investment. Prediction of stocks is a very difficult task according to academics. Hellstrom et. al (1998), mention that stocks behave like random-walk process, analysed in the book of Lawler and Limic (2010) . The serial correlation in stock prices is statistically insignificant, and the noise level and volatility in prices change as the procedure moves on. So, there are periods of great turbulence and periods with low observed volatility. Traditional methods for forecasting started to exist in the end of 20's, when Yule (1926) invented the *Autoregressive process* to predict sunspots. Another model which was used was the *Moving Average Model.* Autoregressive and Moving Average Models led to their popular combination, the ARMA model, which was dominant for prediction for more than 50 years.

At the book of Guerard (2007) , we see that past values of a variable are used for forecasts for the future. They refer to some models for forecasting like time series modelling, regression and exponential smoothing. Specifically, a regression analysis is a statistical technique that helps us make forecasts for quantitative variables. An example referred comes from Guerard and Schwarz (2007), with the variable of interest being the personal consumption expenditures in the United States, named Y (dependent variable). The variable that we investigate if it has impact on Y, is the personal disposable income, named X (independent variable). The regression analysis tries to find the best fit for the data points available. Regression line is the best way to approximate this relationship. In this study, it is assumed that the regression line assumes that this relationship from the past, will continue to exist for the future values. Though, regression analysis can be expanded to more than one independent variable, and the model used in this case is the Multiple Regression model.

Vrontos et. al (2008), used hedge fund pricing models. They developed a Bayesian model averaging model to study the uncertainty in hedge fund pricing. The model selected with the Bayesian methodology was compared to standard model selection approaches in the literature, i.e. stepwise regression procedure, model selection approaches based on Akaike's criterion (1973) and Schwarz criterion (1978). In this study, the heteroscedasticity is modelled at the same time with the Bayesian Model Averaging method, and this resulted to a better predictive performance of the pricing model.

Vrontos et. al (2011), proposed a flexible threshold regression model that allows nonlinear risk exposures of hedge funds to various risk factors. The results from this study, revealed asymmetric risk exposures to different risk factors. Each hedge fund index included in the analysis, was affected by different risk factors. This result was reasonable since the fund managers follow different investment strategies. The methodology using thresholds in multiple regression model, showed improved performance than standard linear regression model or multiple linear regression models without thresholds.

Continuing at the paper of Bali et. al (2011), it is investigating hedge funds' exposures to various financial risk factors. The exposures are analysed through univariate, bivariate and multivariate estimates of factor betas. This effort was the first for the sensitivity analysis of expected hedge fund returns to factor loadings (betas). The significance of factor loadings was checked with two tests. The first was the parametric test of Fama-MacBeth cross-sectional, and the second was the panel regressions of one-month ahead hedge fund returns on previous months factor betas. All the test agree that, there is a positive and significant relation between hedge fund expected returns and default premium betas. Also, there is a negative and significant relation between inflation betas and expected hedge fund returns.

1.2 Review on machine learning techniques for forecasting

Arindam Chaudhuri (2012) used Multiple Regression, Multi Layer Perception, Radial Basis Function and Adaptive Neuro Fuzzy Inference System Models, for predicting Financial Stress percent. The result from this study was, that the performance of the traditional Multiple Regression model was lower than the performance of other models in this study. So, the Multi-Layer Perception and the Radial Basis Function exhibit the higher

performance in terms of robustness and fault tolerance than the other methods. But, the results in this research come from a limited population and it is clear that the results should be acceptable for a preliminary stage of design.

Bontempi et.al (2013) , states that the increasing amount of historical data and the need of accurate forecasting models, led to the rise of Machine Learning models. The machine learning techniques for time series forecasting focused initially in one-step forecasts. The local learning techniques was discussed if they could deal with temporal data, and finally what is happening if we move from one-step forecasting to multi-step forecasting.

Based on the previous paper, Tyralis and Papacharalampous, (2017) attempted to improve the one-step forecasts' performance in time series forecasting. They proposed the Random Forest algorithm and proved that this is a competent algorithm. They emphasized that the use of few predictor variables achieves higher predictive accuracy. However, this methodology was based on short time series and the researchers set the foundations for more experiments.

Another study of Chan-Lau (2017) , stress that machine learning techniques can deal with the high-dimensionality problem. It is stated that LASSO regression is a technique that can perform well in forecasting when we have large number of covariates and small number of observations. The result was, that the LASSO regression builds more stable forecasting models, and can handle the high-dimensional problems, including these in financial area.

One recent study from Fischer et.al (2018) , eight different machine learning algorithms are presented for forecasting linear and non-linear time series. The results in the absence of noise give the Multi-Layer Perceptron as the best forecasting model and as the worst, the Single Decision Trees and the Naïve Bayes models. It is also proved that the inclusion of lagged variables improves the predictive performance. In the end of the analysis, it is showed that if we add migitation measures (Moving Averages), the most robust results come from the Logistic Regression algorithm, because it is not so sensitive to noise presence as the other Machine Learning models.

1.3 Objective

The aim of this thesis is, to explore the predictive performance of hedge fund returns, using selected risk factors. For this purpose, we use algorithms from the machine learning bibliography. These algorithms are, Least Absolute Shrinkage and Selection Operator (LASSO), Regression Trees and Random Forests. The performance of the above is compared with standard approaches for forecasting such as, Multiple Regression and time series model ARMA(1,1).

Chapter 2

Data

2.1 Hedge fund returns

Our data are financial data, and especially hedge fund indices from Hedge Fund Research [1] , computed on a monthly basis. Hedge funds are investment companies that bind investors money for a certain period of time (usually one year) to raise their capital. After the end of one year, the investor chooses whether to withdraw his capital from the hedge fund or it has to grow. There are few employees usually, and among them are economists and distinguished university professors who attract investor confidence. Employees are paid out of the profits made by the hedge fund by 10-15% of profits. For example, if the fund has a profit of 100 million euros, then the employees will share 10 million euros between them.

The performance of hedge funds over the last 20 years has been investigated in a study of Atilgan et al. (2013) , which presents as two dimensions of the performance of hedge funds, the returns and the risk. The results from this study showed that, the knowledge of only one of these dimensions does not make sense if we do not have knowledge about the other. Also, the distribution of hedge funds did not follow the normal distribution, showing left skewness and leptokurtic distributions. They tended also to be high-volatile and in periods of crisis, the majority of them produced highly negative returns. This last finding led to doubts for benefits in investing to hedge funds.

2.2 Time series and financial data characteristics.

The variables we are going to use as dependent variables, are 10 hedge fund indices which belong to the period from April 1990 until December 2005. Along with these variables we have some risk factors , which will be used as explanatory variables. We have *time series* data. Time series data is a sequence of data which take some values in specific time periods. In our case, the period is a month. For example if we have a variable X, the sequence is Xt, where t=1,2,…,T describes the time parameter.

[1] https://www.hedgefundresearch.com/

The returns of financial assets are usually calculated as:

$$R_t = ln(P_t) - ln(P_t - 1) = ln\left(\frac{P_t}{P_{t-1}}\right)$$

$$, \text{ or } Rt = \frac{P_t - P_{t-1}}{P_{t-1}} \quad . \textbf{(2.2.1)}$$

P_t is the price of the financial asset at time *t*. In the dataset we analyze the returns Rt, computed based on the logarithmic formula of (2.2.1)

There are some properties, especially in the financial assets, which make them interesting in their analysis, for example leptokyrtosis, leverage effect, fat tails. We should pay attention on how we could capture these properties with a model. We present the response variables and the risk factors at the next tables and figures.

EH	M	RVA	ED	CA	DS	EMN	MA	EM	FIA
-0.01541	-0.01411	0.008292	0.004892	0.008092	-0.00021	0.000592	0.003092	-0.02541	0.015592
0.052525	0.033125	0.009025	0.012625	0.010825	0.001525	-0.00168	0.016125	0.059025	-0.00348
0.018533	0.009833	0.002933	0.007033	0.010533	0.029833	0.007033	0.000633	0.012133	-0.00517
0.013558	0.031858	0.007058	0.004958	0.005058	0.013758	0.001258	-0.00624	0.052358	0.000358
-0.02515	-0.04415	-0.01095	-0.05165	-0.00815	-0.02535	0.01165	-0.01455	-0.12705	-0.00605
0.010358	-0.01594	0.004258	-0.03394	-0.01084	-0.04194	0.011958	-0.05194	-0.07324	-0.00124
0.001567	0.019867	-0.00053	-0.02213	-0.02173	-0.03473	0.007567	0.001167	0.042067	0.006067
-0.02894	0.005358	0.007158	0.001958	-0.00654	0.003958	0.002258	0.015858	-0.02794	-0.00054
0.004675	0.011575	-0.00153	0.010775	-0.01043	0.005675	0.014575	0.006575	-0.00523	0.000175
0.043683	-0.00782	0.020583	0.004683	-0.00092	0.014683	0.019783	-0.00522	0.018983	0.034683
0.046833	0.057933	0.025833	0.039033	0.010933	0.036233	-0.00477	0.010733	0.076733	0.019033
0.067275	0.063575	0.021675	0.028775	0.008975	0.063475	0.022075	0.018075	0.036675	0.010275
-0.00005	-0.02155	0.01325	0.02585	0.01015	0.04855	-0.00485	0.02355	0.01465	0.01405
0.027267	-0.00463	0.002267	0.023767	0.004667	0.008867	-0.00493	0.010767	0.020367	0.018667
0.001158	0.021458	-0.00924	-0.00234	0.005058	0.012458	0.000858	0.006458	-0.01644	0.009158
0.009358	0.019558	0.009658	0.018558	0.010958	0.027058	0.020258	0.009658	0.049558	0.014858
0.017133	0.062433	0.000533	0.008033	0.016333	0.009133	-0.00177	0.001833	0.003333	-0.01277
0.038617	0.055217	0.015917	0.010417	0.008717	0.014717	0.014817	0.006617	-0.00468	-0.03018
0.007475	0.019175	-0.00923	0.019275	0.008075	0.016075	0.005575	0.009975	-0.00043	-0.00443
-0.01453	0.003775	0.002875	0.001475	0.012875	0.000175	0.007975	0.010075	0.015175	-0.01543
0.046908	0.071108	-0.01519	0.013608	0.013008	0.005008	0.017408	0.008708	0.119408	0.011308
0.021617	0.024717	0.053917	0.038817	0.017917	0.067317	0.000317	0.016317	0.077317	0.043717

Table 1 : A subset of the response variables used for the predictive model.

RUS-Rf	RUS(-1)-R	MXUS-Rf	MEM-Rf	SMB	HML	MOM	SBGC-Rf	SBWG-Rf	LHY-Rf	DEFSPR	FRBI-Rf	GSCI--Rf	VIX	Rf
-0.03557	0.016289	-0.01946	0.056009	-0.0038	-0.0264	0.0042	-0.01541	-0.0098	-0.26108	-0.0011	-0.00069	-0.01723	-0.0021	0.006708
0.077211	-0.03557	0.098165	0.070701	-0.0271	-0.0377	0.0418	0.020685	0.025998	0.003079	0.0014	-0.01255	-0.01408	-0.0215	0.006675
-0.01329	0.077211	-0.01691	0.025924	0.013	-0.022	0.0177	0.009515	0.011514	0.008039	0.0009	-0.00311	-0.02229	-0.0187	0.006667
-0.01917	-0.01329	0.0066	0.075455	-0.0319	0.0006	0.0308	0.006612	0.024398	0.014973	-0.0001	-0.02199	0.072856	0.0561	0.006442
-0.10938	-0.01917	-0.10736	-0.14603	-0.0358	0.0146	0.0169	-0.02136	-0.01419	-0.07189	-0.0007	-0.02144	0.147151	0.0879	0.00635
-0.06268	-0.10938	-0.15297	-0.1454	-0.0368	0.0062	0.0366	0.002791	0.004975	-0.10331	0.0009	-0.01321	0.200381	-0.0079	0.006142
-0.01702	-0.06268	0.128817	-0.02622	-0.0562	0.0032	0.0615	0.007164	0.03759	-0.03138	0.0027	-0.02287	-0.06856	0.0093	0.006133
0.05657	-0.01702	-0.06487	-0.06195	0.005	-0.0331	-0.0061	0.014813	0.010399	0.022967	0.0021	-0.00417	-0.07391	-0.0788	0.006042
0.017452	0.05657	0.009883	0.028555	0.0093	-0.0181	0.0629	0.009997	0.0043	0.010945	0.0012	0.011884	-0.0254	0.0422	0.005525
0.045677	0.017452	0.023125	0.072729	0.0389	-0.0181	-0.0946	0.005749	0.019371	0.018683	1.00E-04	4.45E-05	-0.10389	-0.0547	0.005317
0.06346	0.045677	0.093438	0.132013	0.0397	-0.0056	-0.1147	0.001443	-0.00486	0.106387	-0.0014	-0.01039	0.008659	0.0032	0.005167
0.022248	0.06346	-0.06591	0.031872	0.0395	-0.0124	-0.0407	0.002527	-0.04193	0.026215	-0.0024	0.032281	0.032009	-0.0435	0.004925
-0.00581	0.022248	0.003709	-0.00174	0.0037	0.017	-0.0224	0.007905	0.010534	0.016191	-0.0008	0.011069	0.017639	0.0136	0.00475
0.032582	-0.00581	0.005771	0.067951	-0.0037	-0.0052	0.0282	-0.00023	-0.00601	-0.0078	-0.0011	0.002351	-0.00524	-0.0231	0.004733
-0.0536	0.032582	-0.07982	-0.04322	0.0006	0.0102	0.001	-0.00499	-0.01526	0.018259	-0.0011	0.010853	-0.02869	0.0362	0.004742
0.039245	-0.0536	0.040106	0.043581	-0.0073	-0.0153	0.0269	0.008063	0.016405	0.02505	-0.0006	-0.00278	0.045057	-0.0437	0.004742
0.018313	0.039245	-0.02563	0.013598	0.0174	-0.0099	0.0218	0.017284	0.0146	0.035511	0.0013	-0.0114	0.010717	-0.0072	0.004567
-0.01852	0.018313	0.045525	-0.04464	0.0158	-0.0089	0.0192	0.016974	0.033996	-0.00116	0.0011	-0.01458	0.003492	0.0139	0.004383
0.010963	-0.01852	0.009901	0.034961	0.0103	-0.0061	0.0282	0.003965	0.006296	0.027241	0.001	-0.00439	0.024167	-0.0037	0.004125
-0.04556	0.010963	-0.05177	-0.0197	-0.0086	-0.0169	-0.0032	0.006701	0.011785	-0.01072	0.0007	-0.01215	-0.04499	0.0478	0.003725
0.100141	-0.04556	0.043001	0.102621	-0.0199	-0.0434	0.0928	0.029646	0.0474	0.01095	0.0014	-0.00787	-0.07667	-0.0095	0.003292
-0.01164	0.100141	-0.0247	0.105951	0.0818	0.0497	-0.111	-0.01722	-0.0212	0.024104	-0.0007	0.000487	0.023834	-0.0191	0.003283

Table 2: A subset of the explanatory variables used for the predictive model.

Before building any model we should start by showing some descriptive statistics and recognizing some properties in the data.

				Rt	**Rt^2**
Assets	Mean	St.dev	Kurtosis	LB-Q(20)	LB-Q(20)
EH	0.0102	0.024	1.45	21.26	50.74
M	0.0092	0.023	0.55	35.66	37.27
RVA	0.0059	0.010	10.86	29.47	10.70
ED	0.0082	0.018	4.84	35.32	10.08
CA	0.0047	0.010	2.28	92.89	14.45
DS	0.0084	0.017	5.39	73.04	26.00
EMN	0.0039	0.010	0.51	42.80	40.54
MA	0.0049	0.010	10.18	17.63	2.43
EM	0.0100	0.042	3.85	56.99	30.86
FIA	0.0031	0.012	10.30	71.47	44.43

Table 3 : Summary statistics of hedge fund returns.

Below, we present some graphical descriptive measures for EH,M. The graphs for the other response variables are presented at the Appendix.

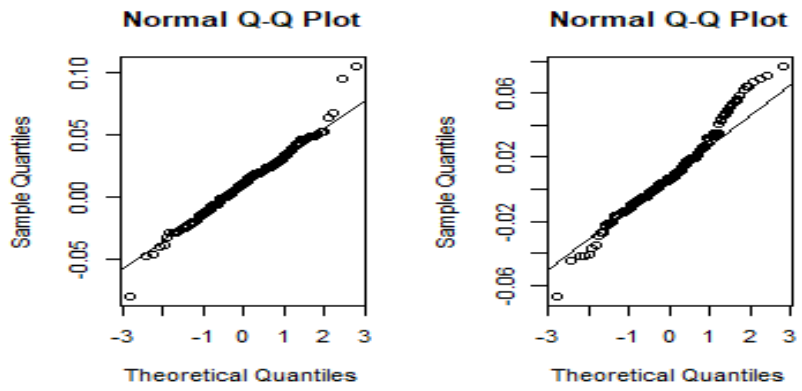


Figure 1: Normal quantile plots of the returns of hedge funds EH,M.

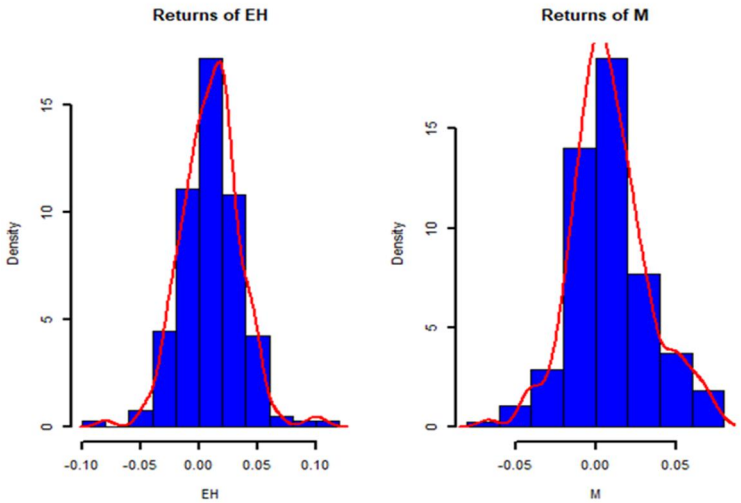


Figure 2: Histograms of hedge fund returns EH,M with a normal distribution curve overlaid.

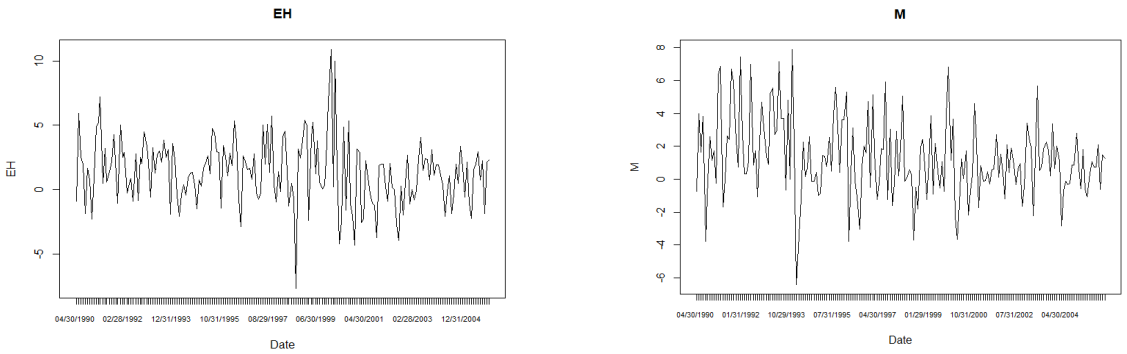


Figure 3: Time series plots of the returns of hedge funds EH,M.

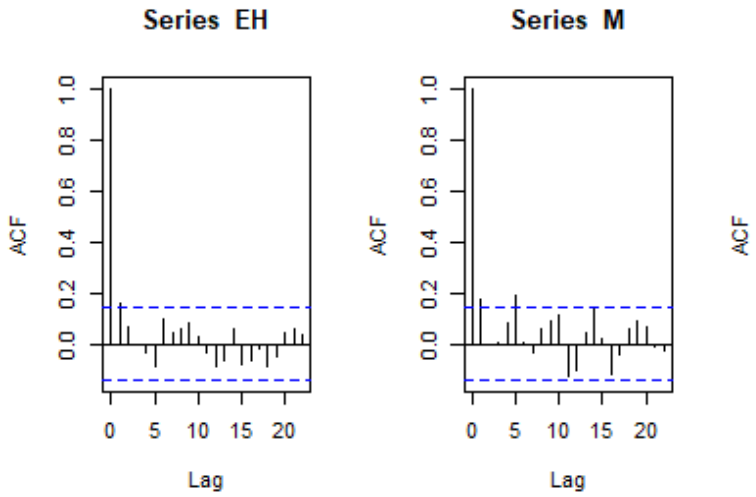


Figure 4: Autocorrelation plots of hedge fund returns EH,M.

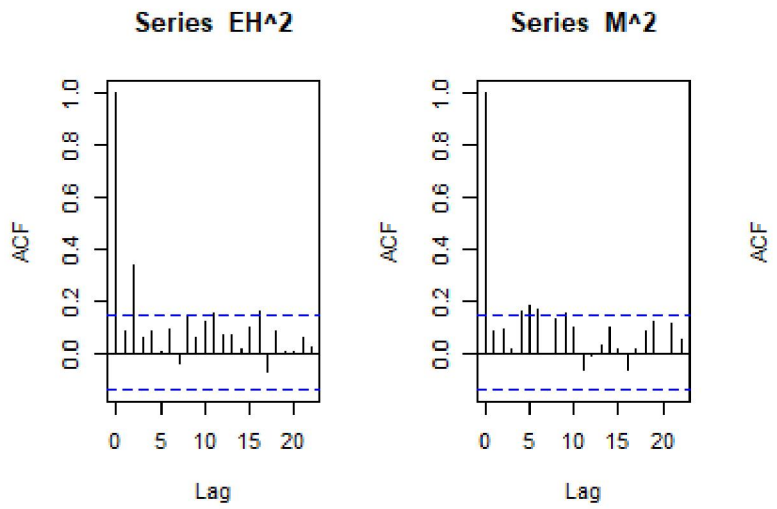


Figure 5: Autocorrelation plots of squared hedge fund returns of EH,M.

2.3 Discussion about the data

In Table 3, we present the summary statistics for the returns of hedge funds. We also have calculated the Ljung-Box statistic (LB) based on 20 lags, for the return series and the squared return series. The Ljung-Box statistic of autocorrelation is defined as:

H_0: The data are uncorrelated

H_1: The data are correlated

The test statistic has a general form:

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k} \qquad \textbf{(2.3.1)}$$

where n is the sample size, $\hat{\rho}_k$ is the sample autocorrelation at lag k, and h is the number of lags being tested. If **H_0** is valid, the equation **(2.3.1)** follows a $x^2_{(h)}$ distribution. In our analysis, Q-statistic must follow a $x^2_{(20)}$ distribution.

distribution. For significance level 5%, the critical region for rejection of the null hypothesis is:

$$Q > \chi^2_{1-0.05,20}$$

The corresponding critical value from the table of X-square distribution, is 31.41.

The results from the Table 3 show, that we have excess *kurtosis*, varying from 3.85 for EM to 10.86 for RVA, at six out of ten return series. This fact indicates *fat tails* in most of the hedge fund return distributions. We observe that the distributions of RVA, ED, CA, DS, MA, EM, FIA present fat tails and deviate from normality. The LB statistic tests the null hypothesis of no autocorrelation of the return series, versus the alternative hypothesis, that there is autocorrelation. The null hypothesis of no autocorrelation is tested at significance level 5%. We see from the Table 3, that the null hypothesis is rejected for the return series M, ED, CA, DS, EMN, EM, FIA. The LB statistic for the squared returns in the same way gives us that, the null hypothesis of no autocorrelation in the squared return series at significance level 5%, is rejected for the series EH, M, EMN, FIA. So, from this last result we conclude that there is evidence for *heteroscedastic effects* in four out of ten return series. The indication of heteroscedasticity can be shown from the time series plots in figures 3 and 28 (Appendix), where we observe that in some periods volatility is large and in other periods is lower. This characteristic is named *volatility clustering,* and in some way it causes the fat tails and the kurtosis in the distribution of the data.

To capture these characteristics, we could use models borrowed from time series theory. We can construct models that explain the return series (yt), by adding past values ($y_{t-1}, y_{t-2,\ldots}$) and past stochastic terms ($\varepsilon_{t-1}, \varepsilon_{t-2,\ldots}$). This could explain the autocorrelation between the latest and newest observations. This model is called *Autoregressive Moving Average*. There will be an analytical explanation of this model in Chapter 3.

The conditional heteroscedasticity requires a model for the variance. In our analysis we will use GARCH-type models for the variance. There will be also an analytical reference for these models in Chapter 3.

Chapter 3

Methods of forecasting

3.1 Machine Learning algorithms

Machine learning is a section of computer science that gives to the computers the ability to execute jobs efficiently by the use of algorithms. The more formal definition proposed by Tom Mitchell (1997) for the algorithms analyzed in Machine Learning is the following: : "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E".

Machine learning is constructed on the field of Mathematics and Computer Science. So, our algorithms will be based on certain mathematical language from the field of:

1)Probability : study of likelihood of events

2)Statistics : study of the methods to collect, analyze, describe and present data.

3)Artificial intelligence: Study and construction of systems that imitate things that a human can do. It was first used by McCarthy (1956). Some examples of Artificial Intelligence include speech recognition, language translation, self-driving cars.

Machine learning algorithms can learn from and make, data-driven predictions or decisions through building a model from sample inputs. Some of the applications , (Thagard ,1990) include email filtering, detection of network intruders, pattern recognition, speech recognition, medical diagnosis, natural language processing, physics, problem solving, game playing, robots. From the side of data analytics machine learning is a method used to derive models and algorithms useful for prediction, known as predictive analytics. These models allow data scientists to make decisions and uncover relations and trends in the data through the time.

3.1.1 Historical information for Machine Learning

The initial step for the foundation of Machine Learning has begun in 1955, from the British mathematician Alan Turing raising the question whether a machine can think. This idea led to the birth of the first computer. Some years later, Arthur Samuel introduced

the term "Machine Learning" while working at IBM . Over these 65 years many people around the world have contributed to this growth, especially people studying at universities and software developers.

Some interesting recent achievements include the following:

- In 2011 Google's deep neural network learned to discover and categorize objects.

- In 2014 Facebook developed the Deep Face algorithm which could recognize and verify people in photos the way humans do.

- In August 2017 a new Artificial Intelligence system based on Neural Networks was trained to write the first five chapters of the next book of the popular fantasy series Game of Thrones.

3.2 How do machine learning algorithms work

Our goal in machine learning is to learn a target function, usually described as (f) that maps input variables (X) to an output (Y)

$$Y= f(X) \quad \textbf{(3.2.1)}$$

We try to approach the best form of f, which means the best mapping of X's to Y. The main use of the learning task, is to make predictions for the future(Y) given new examples of inputs (X).If we had the exact form of f, we could use it directly to make our predictions. But, we do not. So, we estimate the form of the function from our data and we have an error (e).

$$Y=f(X) + e$$

This error is independent of the inputs (X).

So, the procedure of learning the mapping Y=f(X) to make predictions of Y when we have new X is called predictive modeling . Our aim is to improve the estimate of the function f, so as to have better predictions made by our model.

We will discriminate algorithms by type of learning and similarity of their function .We will also analyze the parametric and non-parametric machine learning algorithms.

3.2.1 Supervised, Unsupervised and Semi-supervised algorithms

1)Supervised Algorithms

Supervised learning is the procedure, when you have input variables (X) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

The aim is to approximate the mapping function (3.2.1) so well that when you have new input data (X) that you can predict the output variables (Y) for that data. We split the data into training and testing set. It is called supervised learning because the process of an algorithm learning from the training set can be thought of as a teacher supervising the learning process. The training set has output variable (Y) which needs to be predicted or classified. Since we found the form of the function, the next step is to apply it to the testing set for prediction or classification. Three important supervised algorithms are:

- Regression algorithms
- Decision/Regression trees
- Supporting Vector Machines

2)Unsupervised Algorithms

In the case of unsupervised learning, we have only input variables (X) and not corresponding output variable (Y). In contrast with supervised algorithms, there is not a teacher supervising the learning process. Algorithms must discover on their own the relations in the data. There are two main categories of unsupervised algorithms:

- Clustering algorithms
- Dimension reduction techniques (e.g. PCA,QDA)

3)Semi-Supervised Algorithms

Semi-supervised algorithms are used when we have a lot of inputs(X) and only a few of these have corresponding output (Y). We can use unsupervised techniques to guess

predictions for the data with no corresponding output (Y). After completing this step, we could continue by setting these data as training data to a supervised algorithm. In the end, when we have found the form of the mapping function we can use it to predict on new unseen data. We have two basic categories:

- Generative Algorithms
- Self-training Algorithms

3.2.2 Types of functions

Our next step is to categorize the algorithms by similarity of their function. We will refer to popular machine learning algorithms, grouped by similarity. So, we have:

- Regression Algorithms

i) Ordinary Least Squares Regression
ii) Linear Regression
iii) Logistic Regression

- Instance based algorithms

i) k-nearest neighbors (Knn)
ii) Learning Vector Quantization (LVQ)

- Regularization algorithms

i) Ridge Regression
ii) Least Absolute Shrinkage and Selection Operator
iii) Elastic Net

- Decision tree algorithms

i) Classification tree
ii) Regression tree

- Bayesian algorithms

i) Naïve Bayes

ii) Gaussian Naïve Bayes

- Clustering Algorithms

i) K-Means

ii) Expectation Maximization algorithm

- Dimensionality reduction algorithms

i) Principal Component Analysis

ii) Quadratic Component Analysis

- Artificial neural network analysis

i) Back-propagation

ii) Perceptron

Complex machine learning problems can be reduced to these 4 basic types:

- Classification
- Regression
- Clustering
- Rule extraction

For every problem we can find and test an algorithm addressed to that problem, which belongs to one of the above basic forms.

3.2.3 Parametric and Nonparametric algorithms

Another discrimination between algorithms include the parametric and non-parametric algorithms.

Parametric machine learning algorithms have a known form of function and a set of parameters of fixed size. A parametric algorithm needs the following steps to be built:

1)A specific form for the function

2)Use the data to learn the coefficients for this function

Typical examples of parametric algorithms are the following:

- Logistic Regression
- Perceptron
- Linear Discriminant Analysis

Nonparametric algorithms do not make assumptions about the form of the mapping function. They have the advantage to learn from a variety of forms occurring from the data. These methods try to fit the existing data by constructing the mapping function, but they try also to generalize to new data. The most familiar nonparametric algorithms are:

1)K nearest neighbor algorithms(k-nn algorithm)

2)Regression and Classification trees

3)Neural Networks

3.3 Methods used in this analysis

Continuing, we will refer to methods that we will be using for our purpose, which is the prediction of hedge fund indices for the next 24 months. Machine Learning include a number of techniques and then, the most proper technique is used for model selection and forecasting. In our analysis we use forecasting models, which fall under the category of supervised learning. So, in this case we search for the ‘‘teaching rule’’ that explains the

relationship between hedge fund indices and some given factors. The algorithms that will be used in our study, are the following:

- Linear Regression analysis with subset selection
- Lasso Regression
- Regression trees
- Random Forests

Regression analysis

Regression analysis in machine learning refers to a process where we need to find the relationship among variables. More specific, we have a dependent variable and other independent variables, and we try to analyze their relationship. So, we say that one independent variable has impact on a dependent variable if we have the following scheme: We vary the one independent variable and see that the dependent variable changes, while the other independent variables stay fixed.

Simple linear regression

It is the simplest form as its name declares, and we have to predict a quantitative response Y with only a single predictor X. We make the hypothesis of a linear relationship between X and Y, (see for example, Hastie and Tibshirani).

$$Y \approx \beta o + \beta 1 X \quad \textbf{(3.3.1)}$$

We describe the equation (3.3.1) by saying that we *regress Y on X* .In the (3.3.1) the βo and β1 are two unknown constants that represent the *intercept* and the *slope* of the model. If we increase one unit the X, we will have bj as the effect on Y. These two constants are the coefficients of the model, and we use the data to estimate them. We do not derive the Y exactly, so the (3.3.1) takes the following form:

$$Y = \beta o + \beta 1 X + \varepsilon \quad \textbf{(3.3.2)},$$

where ε is a random mean-zero error term,normally distributed.

The estimates $\hat{\beta}_0$,$\hat{\beta}_1$ from (3.3.2) are obtained using the *least squares* method. Once we have found $\hat{\beta}_0$, $\hat{\beta}_1$ we can make predictions with the following equation

$$\hat{y} = \beta\hat{o} + \hat{\beta}_1 x \quad ,$$

where $\hat{y}$ denotes a point estimate given the value of the X=x.

The $\hat{e}_i = y_i - \hat{y}_i$ are called the *residuals* of the model, and actually they represent the deviation of the real-observed value y_i , with the estimated from the regression model $\hat{y}_i$.

Multiple Linear Regression

We analyzed above the case of simple regression. But, in practice we do not have one single predictor for the response. We extend the simple linear model, so that it can include many predictors in the equation. So, the equation of multiple regression is:

$$Y=\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots. + \beta_p X_p + \varepsilon, \quad \textbf{(3.3.3),}$$

for j=1,…,p X_j is the jth predictor and the corresponding coefficient *βj* denotes the association between the predictor and the Y, so if we set constant all the other variables, βj expresses the effect on Y if we increase one unit the value of Xj .

For the estimation of the coefficients we use the same method (ordinary least squares approach) with the simple linear regression, to obtain them. Our main purpose in the multiple regression process is, to find if there are some statistical significant variables in the equation and also, which are these variables, by applying variable selection methods.

Once we have found the coefficient estimates, we continue to make predictions with our model. The accuracy of our predictions depends on how well does the model fits the data. We will introduce the tools of estimation of accuracy to the next paragraph.

Accuracy of the model

Once we have fitted our model, the next step is to quantify how good the model fits the data .The quality of the fit is measured using two related quantities: The *residual standard error* (RSE) and the *R-squared* statistic. The RSE is an estimate of the standard deviation of the ε terms from the (3.3.3) equation. More practical, it is the average amount

of how the response variable will deviate from the estimated regression line. The formula that gives the RSE is:

$$\text{RSE}=\sqrt{\frac{1}{n-2}RSS}=\sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(yi-\hat{y}_i)^2}\ ,$$

,where $\text{RSS}=\sum_{i=1}^{n}(yi-\hat{y}i)^2$

So, RSE measures the lack of fit of the model.

In fact, the R-squared statistic is mostly used for assessing the fit and measures the proportion of variance explained by the model. It can take values from 0 to 1 and the closer it is to 1 the better the fit becomes. The formula which gives us the R-squared is:

$R^2=\frac{TSS-RSS}{TSS}$, where $TSS=\sum(y_i-\bar{y})^2$, is the *total sum of squares.*

Variable Selection Methods

In linear regression the fit of the model can be improved by adding covariates in the model. By adding these covariates we minimize the bias in the expense of higher variance. Nevertheless, this technique leads to poor predictability of the model and bad interpretability of the results. So, it is preferable to choose with some criteria which variables should be included. We have two categories of variables selection methods:

- Subset selection
- Shrinkage

Subset selection

We aim to find the optimal number of covariates in a linear regression model. In this category we have the stepwise selection methods. In *forward* stepwise selection it starts with a least squares model with no covariates, adding one variable at a time based on its contribution to the model fit. The process ends when we cannot add another covariate. In *backward* stepwise selection it starts with a least squares model including all covariates, with each following model with one less variable until only the important variables are included in the model. The variable that is going to be removed is the one that contributes the least to the explanatory power of the model. At the end of the backward deletion process, there is be a set of p models, each corresponding to the best performing model in

the family of models including one covariate, two covariates, and so on. There is another one stepwise method called *step-by-step* stepwise selection. In the beginning, we start from a given model and in each step we decide which variable to include. After adding the best variable we check if we must remove another variable. We select the move that will be executed according to a criterion. At all stepwise methods the model selection. or variable insertion or exlusion criteria are the values that minimize the *Akaike Information Criterion* (AIC) or the *Bayesian Information Criterion* (BIC).For this purpose, it can also be used the adjusted R-squared.

Shrinkage Methods

Shrinkage methods reduce or shrink the values of the coefficients towards zero. The main advantage of these techniques compared to the classical methods of least squares for estimation of the coefficients, is that exhibits less variance. The two widely used shrinkage methods are the *Ridge Regression* and the *Least Absolute Shrinkage Method* (LASSO).

Least Absolute Shrinkage and Selection Operator Method (LASSO)

In high dimensions, where there are some variables in the dataset which are correlated , the traditional ordinal least squares tend to have high variance, although they have little bias. Ridge regression, Hoerl et al. (1970), was a technique proposed to reduce the problem of multicollinearity. The innovation was, that by adding some degree of bias to the regression estimates, the ridge regression reduces the standard errors. So, in this case the quantity we want to minimize is:

$$\sum_{i=1}^{n}(yi - \beta^T z_i)^2 \quad \text{s.t.} \quad \sum_{j=1}^{p}\beta j^2 \leq t \,,$$

where β are the coefficients and z are the standardized covariates

Although ridge regression is not used in practice, it led to the popular LASSO, which is also called $l_1$penalization.

Tibshirani (1996) introduced LASSO, which performs coefficient shrinkage and variable screening at the same time. In the same way as ridge regression, we want to minimize the quantity:

$$\left(y - z_\beta\right)^T \left(y - z_\beta\right) \;\; s.t. \;\; \sum_{j=1}^{p} |\beta j| \leq s$$

$$\Leftrightarrow \; \left(y - z_\beta\right)^T \left(y - z_\beta\right) + \lambda \sum_{j=1}^{p} |\beta j|,$$

λ shrinks each coefficient by a factor λ.

The *s* is called the tuning parameter, controlling the amount of shrinkage applied to the coefficient estimates. If s is big enough, then the resulting solution will be the same with the solution obtained from multiple regression using the ordinary least squares. On the other hand, if s is small enough some coefficients will be equal to 0.

Our aim is to estimate the *s*. Cross-validation is an efficient technique for estimating the tuning parameter. The data split into train data used for estimation and test data used for testing the predictive performance of the model. The most popular process is the k-fold cross-validation, where we split the data into k-folds. The k-1 are used for training and the last one for testing. Since we have fitted the model to the training data, we report the *Mean Squared Error* for the k-th fold. In the end, we select the s that minimizes the Mean Squared Error. Another way is, to run LASSO for a variety of λ values and select the value λ using the Mallows Cp, by Mallows(1973) , an index used for assessing the fit of a regression model. The type of Cp is:

$$c_P = \frac{RSS}{\hat{\sigma}^2 f^{uII}} - (n - 2p)$$

Using the plot of Cp vs s we select the *s* that minimizes the index Cp. The R package that will be used is the 'lars', by Efron, Hastie and Tibshirani (2004) .

Lasso Regression seems to outperform the Ordinal Least Squares in forecasting economic indices, according to Chan-Lau (2017). In fact, each economic index (dependent variable) may depend on a large set of economic factors (independent

variables). By the lasso screening of variables, the results occurring are quite enough interpretable, which is very important for financial forecasting problems.

Regression trees

Regression trees belong to the *tree-based* methods for regression. There are also the *classification* trees but , we will not be consumed on this category, because it involves qualitative variables for the predicted responses, and economic indices are continuous.

The tree-based methods divide the predictor space into a number of simple regions. Our aim is to make a prediction given an observation, and we use the mean or mode of the training observations in the region in which this belongs. All the splitting rules used for the segmentation of predictor space are visually represented in a tree, usually referred as a *decision tree*.

The regression tree is built as a binary tree. It begins with a first node, which is the root node and each node has two child nodes. The split at each node is determined from an explanatory variable, given from a set of explanatory variables of the dataset, which reduces the most of the deviance. While moving down at each node, we have to make a decision whether to go to the *left* or to the *right sub-branch*. If the condition we test is satisfied, then we move to the left sub-branch. Otherwise, we go to the right sub-branch. Since we have chosen the *leaf node*, we calculate the prediction with the formula:

$$m_C = \frac{1}{n_c} \sum_{i \in C} y_i \quad ,$$

where we denote n_c the total observations in the leaf node.

The whole sample space is partitioned into regions R1,R2,..,Rc, so the predicted response will be:

$$f(x) = \sum_{i=1}^{c} m_C I(x \in R_C) \quad ,$$

where $I = \begin{cases} 1, x \in Rc \\ 0, x \notin R_c \end{cases}$.

We calculate the sum of squared errors for a tree:

$$\sum_{C \in leaves(T)} \sum_{i \in c} (yi - m_c)^2 , \quad \textbf{(3.3.4)}$$

This method of prediction may give good predictions, but it overfits the data leading to poor results on the test set. This problem occurs because the tree may be too complex. The idea to address this situation is, to build a tree so long, as the decrease in the equation of sum of squared errors (3.3.4) exceeds some big enough threshold. This leads to a smaller tree, a *pruned tree* and the process explained above is called *tree pruning.* The package that will be used is the *rpart()* in R, and the control parameters we specify are the threshold complexity parameter *cp,* which decides how much reduction in the deviance will occur if a split is attempted, and the *minsplit*. Minsplit is just the minimum number of observations at a node for computing the split.

Tree-based methods have some advantages. The most important is, that trees are easy to present to the people. The binary style of decision trees is more closely to human decision-making than the multiple regression. They can also be presented graphically and interpreted by a person who is not an expert in machine learning algorithms. But, unfortunately their predictive accuracy is not so competitive with the standard regression approach . However, if we combine many decision trees, we can improve the predictive performance of the simple tree approach. In the next paragraph we will show how it can be accomplished with the *random forest* approach.

Random Forests

Random forests is a method of building many decision trees on bootstrapped training samples proposed by Breiman (2001). But, when we are building a tree, each time a split is required, we have to choose from a subset *m* of predictors. The basic difference from the regression tree approach is, that the best variable for the split is decided from a set of *m* predictors randomly chosen at the node, while for the regression trees we have to decide from the total explanatory variables set.

By using this improved technique we create many different trees, and in this manner we avoid the correlation between predictors, if we have a large number of them in the dataset. By using the Strong Law of Large Numbers, we have that the algorithm will always

converge and the model does not overfit. After the construction of N trees, the predicted response will be:

$$\hat{f}_{rf}^{N}(x) = \frac{1}{N}\sum_{n=1}^{N} T\left(x; \Theta n\right), \quad \textbf{(3.3.5)}$$

where Θn is the n^{th} random forest tree, and finally with (3.3.5) we calculate the averaged prediction from all N trees.

The algorithm will be carried out with the package *randomForest()* in R and the control parameters that we should pay attention are:

1)ntree: The number of trees that build the random forest. The default parameter is 500 and it is often the most preferable choice, so as to minimize the test error.

2)mtry: It is the number of predictors which will be used, to make a decision from at each split. The default choice is m=p/3, but it is often used the half value of the default or the double value of the default.

A wise choice for the 1),2) values occurs after making some trials by changing every time the default values, until we reach the best model.

Random forests technique is a very good method for prediction. It does not overfit, and the element of randomness at the step of variable selection, decorrelates the trees and gives more reliable predictions.

3.4 Models from econometric theory

3.4.1 ARCH/GARCH models

The ARCH/GARCH models (Autoregressive conditional heteroscedasticity / Generalized autoregressive conditional heteroscedasticity), are used for modelling the volatility of financial time series. As we have seen in Section 2.3, the volatility of some financial assets is non-constant over time. The ARCH models by Engle (1982), can be defined in a dynamic linear regression problem. In standard linear regression problems the error ε is assumed to be normally distributed with mean μ and constant variance σ^2 . This measures the size of error, and if it is constant we call this *homoscedasticity.* However, in the financial series the assumption of constant volatility is not valid. Assuming that we

have a time series of observations of the returns (y_t) of an asset, Haavelmo (1944) consider it as a stochastic process of random variables. So, we have a sequence of variables characterized by a joint probability distribution at each moment. The process is called *weakly stationary* if the mean and the variance of the process are constant and do not depend on time. We say then that, the terms μ_t and $\sigma^2{}_t$ are the unconditional mean and variance of the process. In finance we use past information for making predictions. So, we need the conditional distribution $f_{t2}(y|It_1)$ of the variable y_{t2} at time t_2 conditional on the information I_{t1} known at time t_1. Based on this information we compute the conditional mean and the conditional variance. We can have weakly stationary process with time-varying conditional mean and volatility models.

Engle's ARCH model allows us to define the best weights to use for forecasting the variance. To use the ARCH model we must have a weakly stationary process. The ARCH specification model for the variance is:

$$h_t = \omega + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2$$

, where a_i are estimated by the given data, and the model of the response Y_t with X_t, the vector with the predictors, is :

$$Y_t = \beta_{0+}\beta_1 x_{t-1} + \cdots + \beta_k X_{t-k} + \varepsilon_t ,$$

The form of the errors is:

$\varepsilon_t = \sqrt{h_t} z_t$, and z_t are independent standard normal variables.

A more general form of this model is the GARCH, introduced by Bollerslev (1986) . The main idea is, that the predicted variance in the next time period is a weighted-average of the long-run average variance, the variance predicted for this period, and also the new information in this period, that is captured by the most recent squared residuals. So, the notation of GARCH model for the variance with order p and q, presented as GARCH(p,q) is:

$$h_t = w + \sum_{i=1}^{q} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j} \quad \textbf{(3.4.1)}$$

As we can see from the (3.4.1) we say that the GARCH(0,q) is same with the ARCH(q) model. The orders p and q of a GARCH model can be identified from the autocorrelation and partial autocorrelation plot of squared residuals.

The main purpose of the construction of ARCH/GARCH models presented above is to understand the risk of a time series. The confidence intervals we construct for a forecast are more realistic if we take under consideration the time-varying volatility. These models capture the volatility clustering phenomenon and the fat tail characteristic of financial data. In addition, the kurtosis calculated with the moments of an ARCH or GARCH model, is always larger than 3, which means that it is larger than the kurtosis of a normal random variable. These characteristics make these models appropriate to use in our empirical applications. We will explain precisely how we will use them in the next chapter.

3.4.2 Autoregressive model

Autoregressive models with order p (AR(p)), are the simplest univariate time series models. These models attempt to predict the response variable using past values of this variable. The general form of an AR(p) is:

$$Y_t = C \quad + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t,$$

where ϕ_i are parameters of the model and $\varepsilon_t \sim N(0,\sigma^2)$.

For the analysis we use Box-Jenkins methodology (1970). The steps for modeling and forecasting are:

1)Identification step: We define the order p from the autocorrelation of the series.

2)Estimation step: We estimate the models parameters using Maximum Likelihood and Least Squares method.

3)Diagnostic step: We check if our model fits well the data, and test for the assumptions of the residuals, i.e homoscedasticity , normality and independency of errors.

4)Make forecasts: After resulting to an appropriate model, we make forecasts from the model.

3.4.3 Autoregressive Moving Average Models

Autoregressive models with order (p,q) attempt to predict the response variable using past values of this variable and past stochastic terms. It is called also ARMA(p,q). The form of an ARMA(1,1) model is:

$$y_t = \delta + \varphi_1 y_{t-1} + \vartheta_1 \varepsilon_{t-1} + \varepsilon_t. \quad \varepsilon_t \sim N(0,\sigma^2).$$

The estimation of ARMA(p,q) models can be done using *Least Squares method* and *Maximum Likelihood method.*

The orders p and q can be identified through PACF and ACF of the residuals. The estimation of the parameters of the model with Maximum Likelihood is the following:

1)We estimate the joint probability distribution, which is actually the probability of having observed this particular sample

2) The maximum likelihood estimator of ϑ is the value that makes the sample most likely to have been observed.

The next steps 3 and 4 are the same with the steps 3 and 4 of model described in 3.5 .

Chapter 4

Estimation and forecasting

4.1 One-step forecasting

In this section, we will use the Methods described in Chapter 3 for our forecasts. The algorithms that carried out the analysis are programmed in R software. We make forecasts only for the next month with the estimated models. We will present the one-step forecasts for the EH and M hedge funds for 24 months. (At the appendix we have forecasts for the other hedge funds).

Hedge fund returns (Rt)
EH
M
RVA
ED
CA
DS
EMN
MA
EM
FIA

Table 4: Response variables

Regressors
RUS-Rf
RUS(-1)-Rf(-1)
MXUS-Rf
MEM-Rf
SMB
HML
MOM
SBGC-Rf
SBWG-Rf
LHY-Rf
DEFSPR
FRBI-Rf
GSCI-Rf
VIX
Rf

Table 5: External regressors

In the beginning, we will start with *EH* and *M*. The total observations in the dataset are 189, so we have values for 189 months. First, we will estimate the forecasting model. For this purpose, the data splits into *training* and *testing* data. The model will be estimated with the training data, and the testing data will be used for assessing the predictive performance of the model. At each step of the estimation, we use all the historical information until the time *t*. So, each time it will be used the latest information related to the response variable for the forecasting model. We will compare the performance of the models, using the ARMA(1,1) model . We have checked for stationarity of the time series (Appendix, Figure 9&10), so we can fit an ARMA(1,1) model in our data.

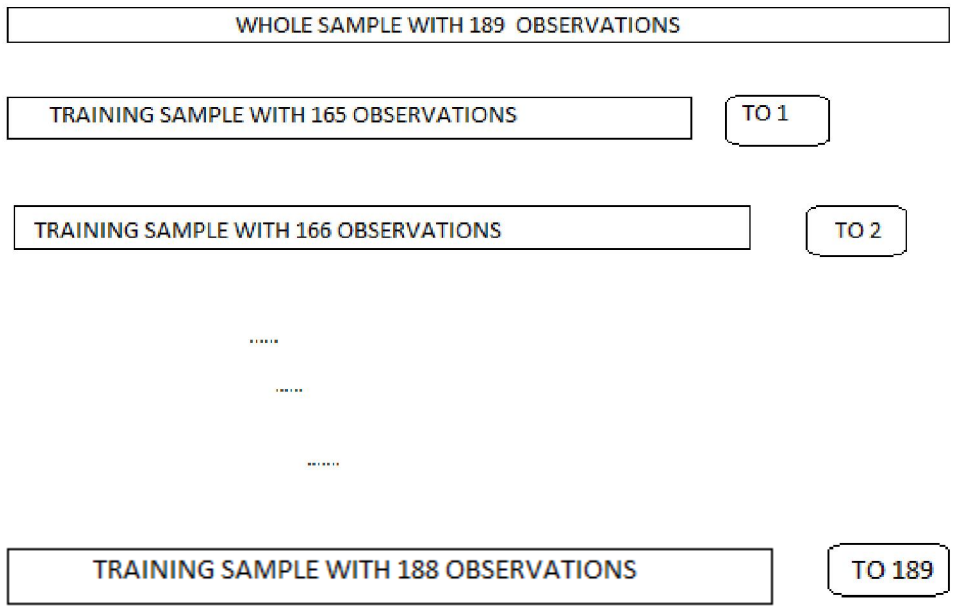


Figure 6: Diagram of the method of model estimation.

So, the mathematical form of the model can be written as:

$$y_{t+1} = f\left(X_{1,t}, X_{2,t}, \dots, X_{k,t}\right) + \varepsilon_{t+1} \quad \text{, where } \varepsilon_{t+1} \sim N(0,\sigma^2) \textbf{ (4.1.1)}$$

The process in Figure 8, shows that the training set of observations is updated at every one-step ahead forecast. The TO denotes the test observation of each step. We use the algorithms explained in Chapter 3, for estimating the mapping function $\boldsymbol{f}$ from the (4.1.1). It is obvious that $\boldsymbol{f}$ will be updated also at each forecasting step, because we have new training data. So, in the end we will have 24 different $\boldsymbol{f}$. Since we have fitted the model, the next step is the forecast. So, in the end of the whole process there will be 24 forecasts. We will compare the results from each method used, by using the *Mean Squared Error:*

$$MSE = \frac{\sum_1^{24}(y(predicted) - y(testing))^2}{24}$$

4.2 Results

We present the results from the models. There is a column named Parameter, which specifies the value of the tuning parameter for the Machine Learning algorithms. In LASSO, the default parameter is the 's' which minimizes the Mallows Cp. The default parameters for Regression Tree are minsplit=20 and cp=0.012. For the Random Forest algorithm, we tried three different values for 'mtry'. The default, the half of the default, and the double of the default. In this case the default value is 5. The results were improved by choosing the double value from the default, so the mtry=10. Variable selection in Multiple Regression is executed with stepwise step-by-step selection. The best technique is the method which minimizes the MSE, and is marked with red frame.

[2] Explained in Chapter3

Model	Parameter	MSE
Multiple Regression	**—**	**0.0002757487**
LASSO	**Default**	**0.00004141184**
Regression Tree	**Default**	**0.0002095322**
Random Forest	**mtry=double(default)**	**0.0000379776**
ARMA(1,1)	**—**	**0.0007402437**

Table 6: Predictive performance for monthly returns of EH.

Model	Parameter	MSE
Multiple Regression	**—**	**0.002308732**
LASSO	**Default**	**0.01081406**
Regression Tree	**Default**	**0.001286918**
Random Forest	**mtry=double(default)** **ntree=default**	**0.0001350695**
ARMA(1,1)	**—**	**0.0005589166**

Table 7: Predictive performance for monthly returns of M.

The performance evaluation for *EH* and *M* are summarized in Table 6 and 7 respectively. We observe that the Random Forest algorithm shows greater performance than the other 4

algorithms used for prediction in both response variables. The algorithm which performs worst for forecasting *EH* is, the Multiple Regression algorithm. On the other hand, the algorithm which performs worst for forecasting *M* is, the LASSO.

Predicted values

0.021151982 0.011592746 0.005899105 -0.018767628
0.007088150
0.011989556 -0.019269913 -0.001905539 0.026558752
0.015146763
0.023109330 0.012833557 -0.003415332 0.016846034 –
0.005628797
-0.021298694 0.027037633 0.019804843 0.030653644 0.010592472
0.009699006 -0.014704024 0.021251099 0.011968074

Table 8: Predicted values of EH using Random Forest.

Predicted values

0.0159163375 0.0126752003 0.0125137524 -0.0196749983 -0.0001103572
0.0087547497 0.0046103859 0.0203501497 0.0231378567 0.0261015291
0.0185165250 0.0207421680 -0.0001677490 0.0267101978 -0.0085580738
0.0011117294 0.0182948213 0.0183827800 0.0252751048 0.0157676740
0.0147982310 -0.0097307280 0.0093476849 0.0237738625

Table 9: Predicted values of M using Random Forest.

It is assumed that the error term follows a normal distribution. Thus, the errors should be normally distributed, with not significant autocorrelations between them and homoscedastic. So, in this step we will present the diagnostic plots for the residuals, and Tables 10 and 11 below, show the findings from the plots.

Model/Assumptions	Normality	Non-autocorrelation	Homoscedasticity
Multiple Regression	✓	✓	X
LASSO	✓	✓	X
Regression Tree	X	✓	X
Random Forest	X	✓	X
ARMA(1,1)	X	✓	✓

Table 10: Checking the assumptions from the models for forecasting EH.

Model/Assumptions	Normality	Non-autocorrelation	Homoscedasticity
Multiple Regression	X	X	X
LASSO	X	X	X
Regression Tree	✓	X	X
Random Forest	X	X	X
ARMA(1,1)	X	✓	✓

Table 11: Checking the assumptions from the models for forecasting M.

From the Table 10, it is shown that the hypothesis of normality is not satisfied only in two algorithms, Random Forest and Regression Tree. The hypothesis of uncorrelated errors is satisfied in all methods for EH, but the great problem is the heteroscedasticity. The heteroscedasticity problem is detected through the acf of squared residuals plot, and the Ljung-Box test of independence for squared residuals.

From the Table 11, in the same way we see that the normality assumption is violated at almost all models, except Regression Tree model. There is a severe problem of autocorrelation in errors and squared errors at all models.

It is obvious that the problem of serial autocorrelation does not exist for ARMA(1,1) model , because the residuals from an ARMA(p,q) procedure resemble to a white noise process.

In Section 2.3 we explained the volatility clustering phenomenon, which causes periods of short and large volatility. The models we presented above do not capture this effect, so there is incomplete information about the behaviour of the series . In the next Section, we will propose a technique to reduce this phenomenon, so as to improve the fit of our models.

4.3 GARCH-correction for volatility clustering phenomenon

In one-step forecasting the returns of the hedge funds before, we used information about the explanatory variables (factors), from the dataset given until time t . Although their performance is sufficient, the volatility of the series is not captured. Machine Learning algorithms assume constant volatility, which has been proven above that is not valid for this data. The Cochran-Orcutt (1949) procedure , is a method from econometrics for estimating a linear regression model with time series data, with autocorrelated errors. Our thought is, to combine the volatility clustering models referred in Sestion 2.3 with the models we have used for prediction.

So, the forecast will be based on these two equations:

$$\hat{Y}_{t+1} = f(X_{1t}, X_{2t}, \dots, X_{kt}) \quad \textbf{, (4.3.1)}$$

$$h_{t+1} = \alpha_1 + \alpha_2 \, \varepsilon_{t-1}^2 + \beta \, \sigma_{t-1}^2 \quad \textbf{, (4.3.2)}$$

where h_{t+1} is the model for the variance of Y_{t+1} .

In the empirical analysis, we use different models for the mean equation , and GARCH(1,1) model to capture the time-varying volatility of financial returns.

Below, we present the predicted values and the MSE for each model. We compare the predictive performance between the models and we plot the Observed and Predicted values from the best model to visualize the results.

In the end, we check the assumptions for the residuals and we expect that all the assumptions should be satisfied after the GARCH-correction for the reduction of volatility clustering.

4.4 Results for different based Regression Garch-type models

Model	Parameter	MSE
Multiple Regression	—	**0.0003780911**
LASSO	Default	**0.00004554623**
Regression Tree	Default	**0.0009160139**
Random Forest	mtry=double(default) ntree=default	**0.0003466563**
ARMA(1,1)	—	**0.0007402437**

Table 12: Predictive performance for monthly returns of EH , using the GARCH-correction.

Model	Parameter	MSE
Multiple Regression	—	**0.002002053**
LASSO	Default	**0.002082637**
Regression Tree	Default	**0.001975184**
Random Forest	mtry=double(default) ntree=default	**0.0008205279**
ARMA(1,1)	—	**0.001741812**

Table 13: Predictive performance for monthly returns of M , using the GARCH-correction.

From the Table 12 and 13 we see the comparison between the predictive performance of the methods after using the GARCH-correction technique. We observe that the best model for forecasting EH is the LASSO. The worst performance is observed at Regression Tree algorithm. The algorithm which shows greatest performance for forecasting M, is Random Forest. The models which perform the least well for forecasting M , are the Multiple Regression and the LASSO.

Below, we use the best model, which is LASSO for *EH* and Random Forest for *M* . In the x-axis we have the Year parameter, which extends from 27/2/2004 to 30/12/2005. The observed values are the actual values from the dataset from 27/2/2004 to 30/12/2005. The observed are shown using blue curve, and the predicted using the red curve, presented in the y-axis both of them. With this graphical representation, it is expected that we have a configuration, that the predicted values approach enough the real values in terms of magnitude as well as direction.

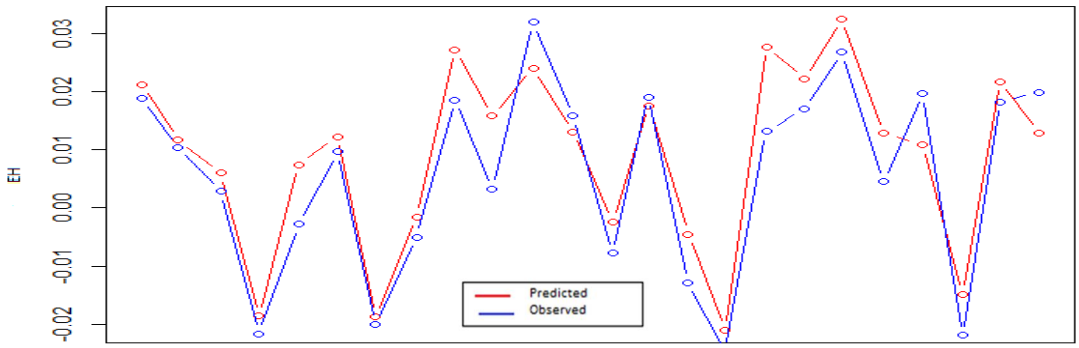


Figure 7: Predicted and Observed monthly returns of EH presented with red and blue curves respectively using the best model (LASSO).

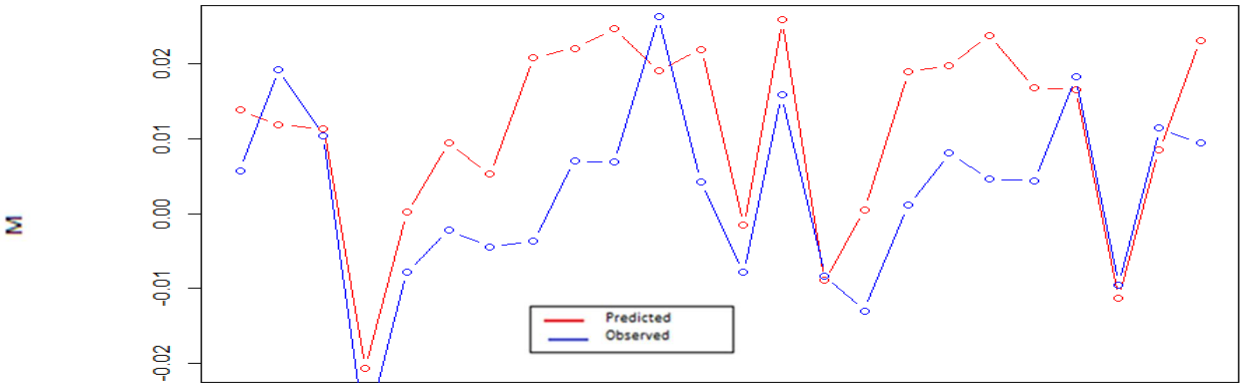


Figure 8: Predicted and Observed monthly returns of M presented with red and blue curves respectively using the best model (Random Forest).

We continue by checking the hypothesis of the residuals in the models.

Model/Assumptions	Normality	Non-Autocorrelation	Homoscedasticity
Multiple Regression	✓	✓	✓
LASSO	✓	✓	✓
Regression Tree	✓	✓	✓
Random Forest	✓	✓	✓
ARMA(1,1)	✓	✓	✓

Table 14: Checking the assumptions from the models for forecasting EH

Model/Assumptions	Normality	Non-Autocorrelation	Homoscedasticity
Multiple Regression	✓	X	✓
LASSO	✓	X	✓
Regression Tree	✓	X	✓
Random Forest	X	✓	✓
ARMA(1,1)	X	✓	✓

Table 15: Checking the assumptions from the models for forecasting M.

So, the results of the assumptions are summarized in Table 14 and 15, where we discover some interesting facts. The technique we have followed has performed enough good at the models for forecasting EH, while there is still space for improvement at the models for forecasting M. The model which we have chosen for prediction for M, is Random Forest. Though, the hypothesis of normality of residuals is violated. The heteroscedasticity effect is absent in all models, and especially in EH, all the assumptions for the residuals are satisfied. On the other hand, some hypotheses are not satisfied at models for M.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Chapter 5

Conclusions and Further Research

In this Chapter we will summarize the results from this thesis. The subject of this thesis targeted to investigate if the use of Machine Learning algorithms could outperform the traditional algorithms for financial forecasting. Continuing , we discussed about the data and their properties. Our response variables, the hedge fund indices have some properties common with other economic features(for example bonds, stocks, exchange currency pairs etc.).

Most of the distributions of the response variables do not follow the normal distribution. Specifically, from the distributions in Figure 27 the RVA,FIA,EM,ED,CA,DS exhibit high kyrtosis, which means that there is a severe fat tail phenomenon in their distribution. Also, after the observation of ACF plots we observe that the autocorrelation plots based on 20 lags do not show significant autocorrelation. So, the explanation of this is, that the value of the next observation does not depend from the value of the previous observation. Though, there is a big problem of heteroscedasticity in all response variables, which can be detected through ACF plots of squared response variables. This phenomenon is crucial for planning our models, because the time-varying variance models must be considered.

Continuing in forecasting models at Chapter 4, the results have shown that EH monthly returns can be forecasted best with Random Forest model if we do not take account the time-varying variance, with accuracy MSE **0.0000379776**. LASSO performs also good enough with MSE **0.00004141184**. The Regression tree and Multiple Regression model with stepwise selection gives almost same results, slightly worse than the other methods. ARMA(1,1) model gives the biggest MSE **0.0007402437**, that means that it is the least good model for forecasting EH.

The results of the residual diagnostics tests for each model had showed heteroskedasticity problem, so we tried to minimize this problem by using GARCH-type

models. In the end, the models for forecasting EH after the correction procedure did not show problem in residuals diagnostics.

The best model for forecasting EH is the LASSO model with MSE **0.00004554623 ,** but the Random Forest is very competitive also, giving an MSE **0.0003466563.** In Figure 24 (Appendix), we present the LASSO coefficients. As we have explained, LASSO is a technique of shrinkage and screening of variables. The variables that are not important in our model have coefficients equal to zero. The R-squared of the model is 79,1 %, which means that a large enough piece of variance is explained from the model. The predicted values are presented in Table 8 and the plot of actual vs predicted values is found in Figure 7 . The plot of Figure 7 indicates that, the forecasts from the model we have chosen, follow the *direction* and the *actual value* of the indices.

The results on M monthly returns show that, Random Forest model performs better in forecasting the returns. The MSE from this model is **0.0008205279.** ARMA(1,1) model is the second better model, but it cannot approximate good enough the direction of the indices as Random Forest can. The importance of variables can be discovered using the *Mean Decrease Impurity (Figure 25)*. As we have explained in Chapter 3, Random Forest consists of many regression trees. Every node in the regression trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure which defines the optimal condition of choice, is called impurity. In the case of regression trees, the measure we will use is the variance. So, the variable selection is implemented by calculating the impurity decrease from each variable. From the above explanation, we conclude that the important variables for forecasting M are**, MEM-Rf, SBGC-Rf, MXUS-Rf, RUS-Rf, LHY-Rf, SBWG-Rf.** The predicted values are presented in Table 9 and the plot of actual vs predicted values is found in Figure 8. We observe that the predicted values approximate perfectly the direction, and sufficiently the magnitude of the actual values.

In addition, we have tried to fit these models to the other response variables. The results of the predictive performance are shown in Tables 14-21 (Appendix). So, we have that **RVA,ED,FIA** are best forecasted with ARMA(1,1) model. **M, EMN,EM** give better results with Random Forest model. Finally, the **EH,CA,DS,MA** are forecasted precisely with LASSO model.

Completing our analysis, the main result from this study is, that the LASSO and Random Forest models outperform the Multiple Linear Regression model. By screening the not important variables using LASSO method, statistical models are more stable [3] . LASSO type estimators may have bias, but the great advantage of this method is that can handle problems of high dimensionality, many of which arise from financial contexts. It offers also interpretable results, so as to be presented to people.

On the other hand, the Random Forest algorithm can be used for forecasting, improving the predictive performance related to Regression Tree algorithm. But, it takes more time to execute than the LASSO, and it is sensitive if we change the tuning parameters. Though, it is easy to interpret because the tree approaches can be easily understood by the people.

Finally, the main findings from this thesis are:

1) Different hedge fund indices are affected significantly from different risk factors.

2) Even if there are some algorithms which perform better than others, there is not a universally best algorithm for all hedge fund indices.

For further research, some other amendments could be tried. For example, we could include lagged response variables in each forecasting model. In addition, we could try to run other more complex machine learning algorithms such as, Artificial Neural Networks, Supporting Vector Machines. According to the research in this area, these algorithms show well predictive performance, but their results are not easy to interpret.

[3] Jorge A. Chan-Lau(2017), Lasso Regressions and Forecasting Models in Applied Stress Testing.

APPENDIX

Stationarity test

```
> kpss.test(my_data$EH,null = c("Level", "Trend"))

        KPSS Test for Level Stationarity

data:  my_data$EH
KPSS Level = 0.49011, Truncation lag parameter = 3, p-value = 0.04389
```

Figure 9: Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for stationarity of time series EH.

```
> kpss.test(my_data$M,null = c("Level", "Trend"), lshort = TRUE)

        KPSS Test for Level Stationarity

data:  my_data$M
KPSS Level = 0.66713, Truncation lag parameter = 3, p-value = 0.01653
```

Figure 10: Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for stationarity of time series M.

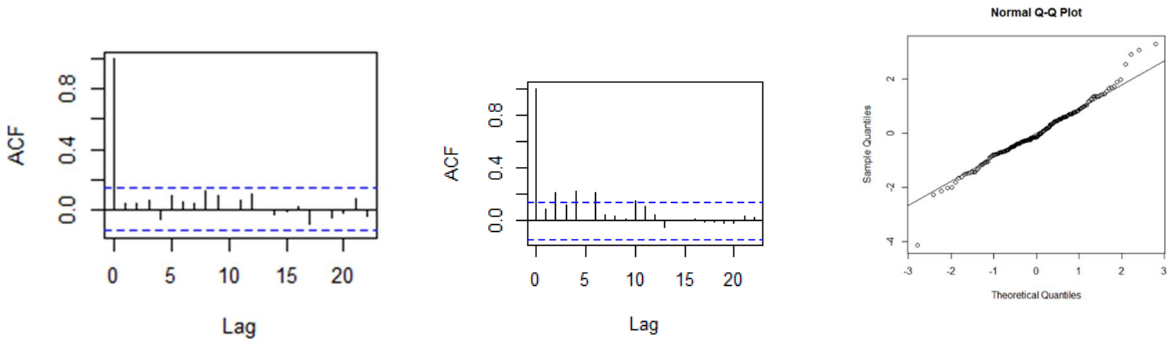


Figure 11: Residual diagnostics for multiple Regression model for EH.

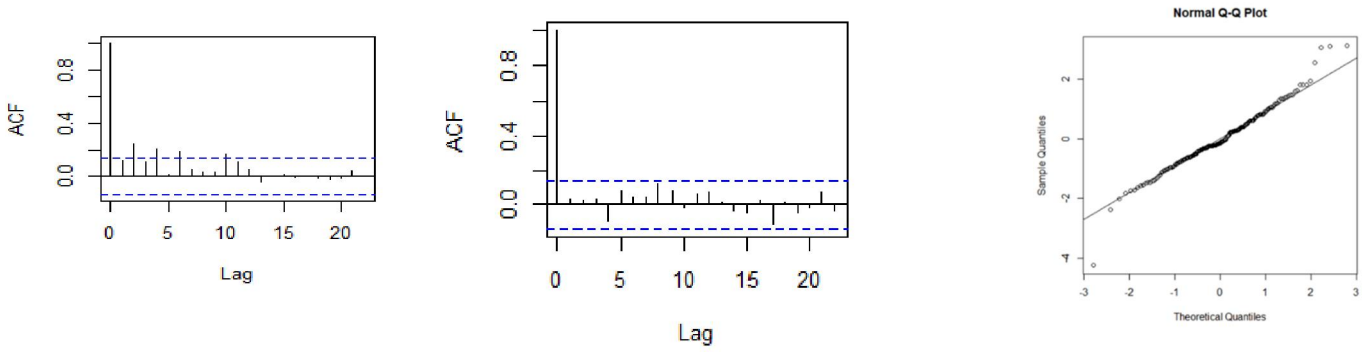


Figure 12: Residual diagnostics for LASSO model for EH.

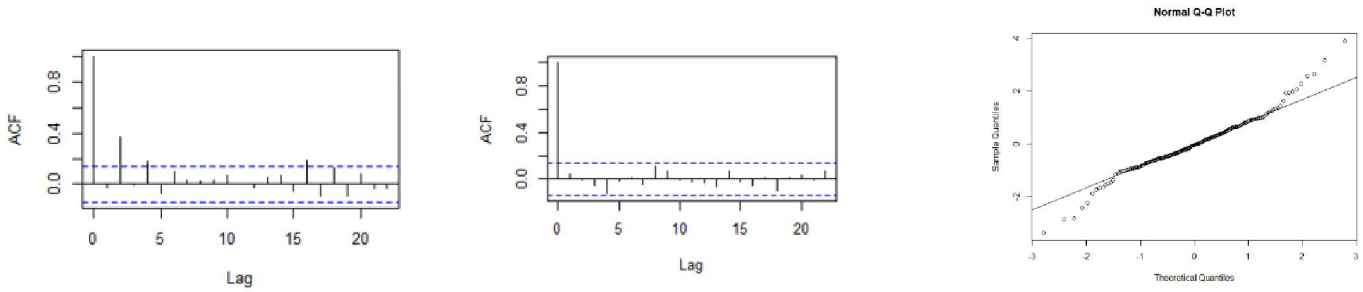


Figure 13: Residual diagnostics for Regression tree model for EH.

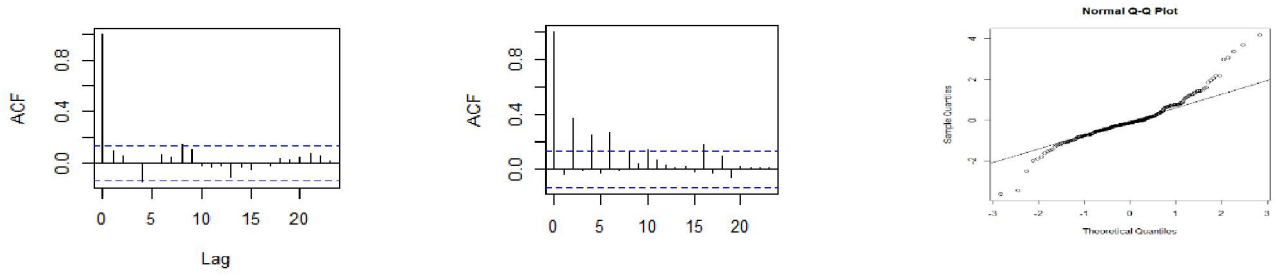


Figure 14: Residual diagnostics for Random Forest model for EH.

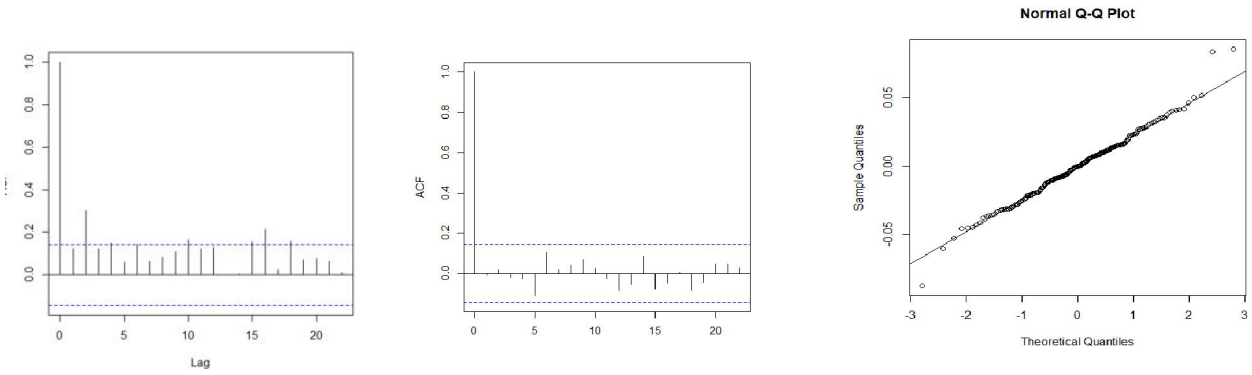


Figure 15: Residual diagnostics for ARMA(1,1) model for EH.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 13.454, df = 20, p-value = 0.8571

Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 28.234, df = 20, p-value = 0.104

Figure 16: Box-Ljung test of independence for the residuals and squared residuals from Multiple Regression model for EH.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 12.431, df = 20, p-value = 0.9004

Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 29.226, df = 20, p-value = 0.0834

Figure 17: Box-Ljung test of independence for the residuals and squared residuals from LASSO model for EH.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 16.04, df = 20, p-value = 0.7142

Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 19.695, df = 20, p-value = 0.4772

Figure 18: Box-Ljung test of independence for the residuals and squared residuals from Regression Tree model for EH.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 12.856, df = 20, p-value = 0.8835
Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 30.992, df = 20, p-value = 0.0553

Figure 19: Box-Ljung test of independence for the residuals and squared residuals from Random Forest model for EH.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 42.782, df = 20, p-value = 0.002184

> Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 30.194, df = 20, p-value = 0.06677

Figure 20: Box-Ljung test of independence for the residuals and squared residuals from Multiple Regression model for M.

> Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 38.145, df = 20, p-value = 0.0085

Box.test(res^2,lag=20)

Box-Pierce test

data: res^2
X-squared = 30.229, df = 20, p-value = 0.06623

Figure 21: Box-Ljung test of independence for the residuals and squared residuals from LASSO model for M.

Box.test(res,lag=20)

Box-Pierce test

data: res
X-squared = 35.029, df = 20, p-value = 0.01995

> Box.test(res^2,lag=20)

Box-Pierce test

data: res^2

X-squared = 25.472, df = 20, p-value = 0.184

Figure 22: Box-Ljung test of independence for the residuals and squared residuals from Regression Tree model for M.

> Box.test(res,lag=20)

Box-Pierce test

data: res

X-squared = 32.92, df = 20, p-value = 0.03443

Box.test(res^2,lag=20)

Box-Pierce test

data: res^2

X-squared = 33.276, df = 20, p-value = 0.05146

Figure 23: Box-Ljung test of independence for the residuals and squared residuals from Random Forest model for M.

```
> coef(lassomodel,s=which.min(rescp$Cp),mode='step')
        RUS-Rf RUS(-1)-Rf(-1)         MXUS-Rf          MEM-Rf             SMB
    0.34784296     0.03192846     0.04720441     0.05164335     0.26644074
           HML            MOM        SBGC-Rf        SBWG-Rf         LHY-Rf
   -0.08352262     0.03366479     0.16257890    -0.07616480     0.00000000
        DEFSPR        FRBI-Rf        GSCI--Rf            VIX             Rf
    0.00000000     0.04621042     0.06823961     0.08976772     1.87149658
```

Figure 24: Coefficients from LASSO model for forecasting EH after GARCH-correction.

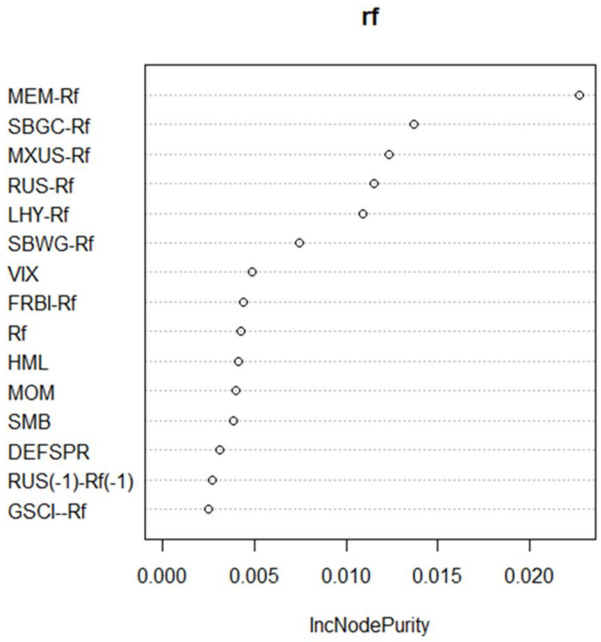
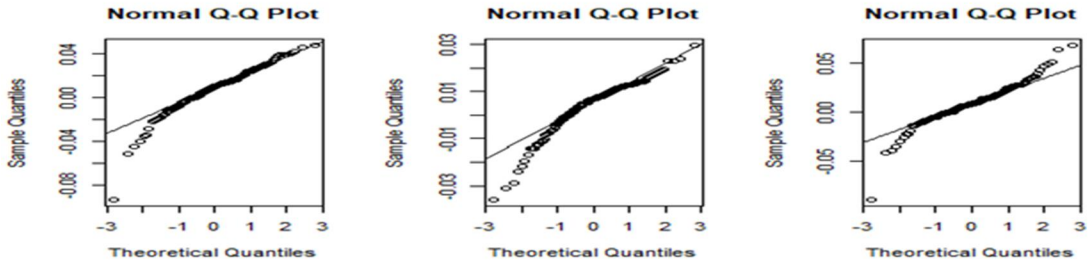


Figure 25: Variable importance of factors in Random Forest model for M after GARCH-correction.



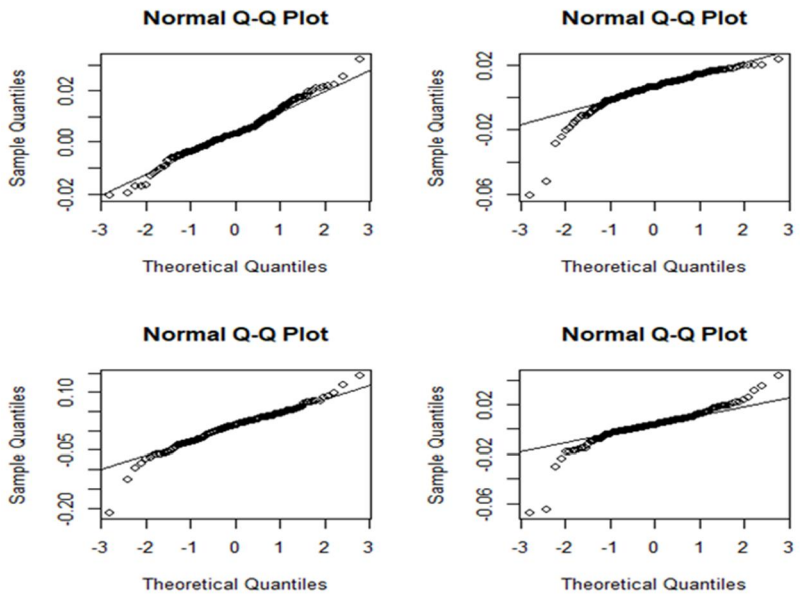


Figure 26: Normal quantile plots of the returns of RVA,ED,CA,DS,EMN,MA,EM,FIA.

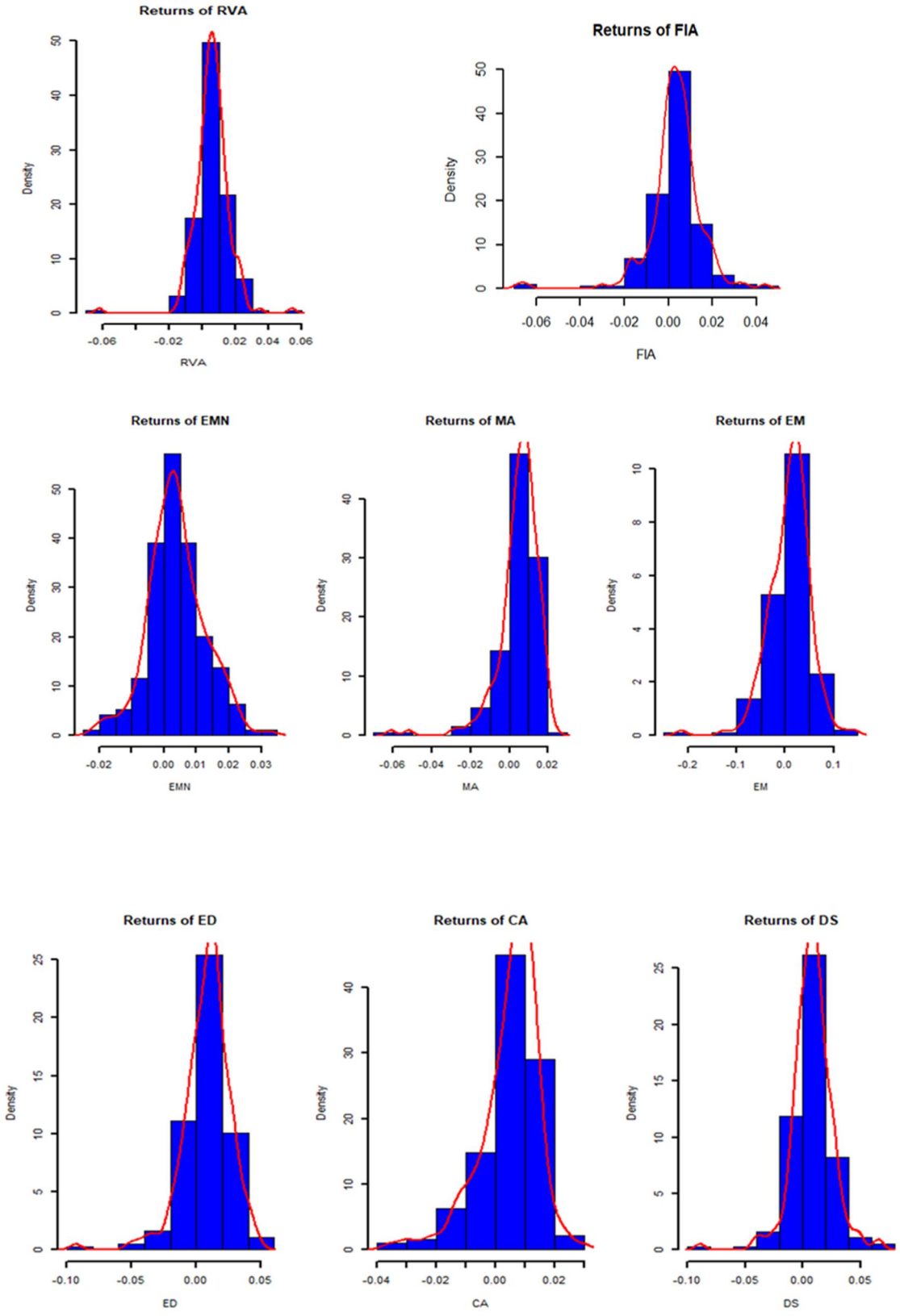


Figure 27: Histograms of hedge fund returns RVA,ED,CA,DS,EMN,MA,EM,FIA with a normal distribution curve overlaid.

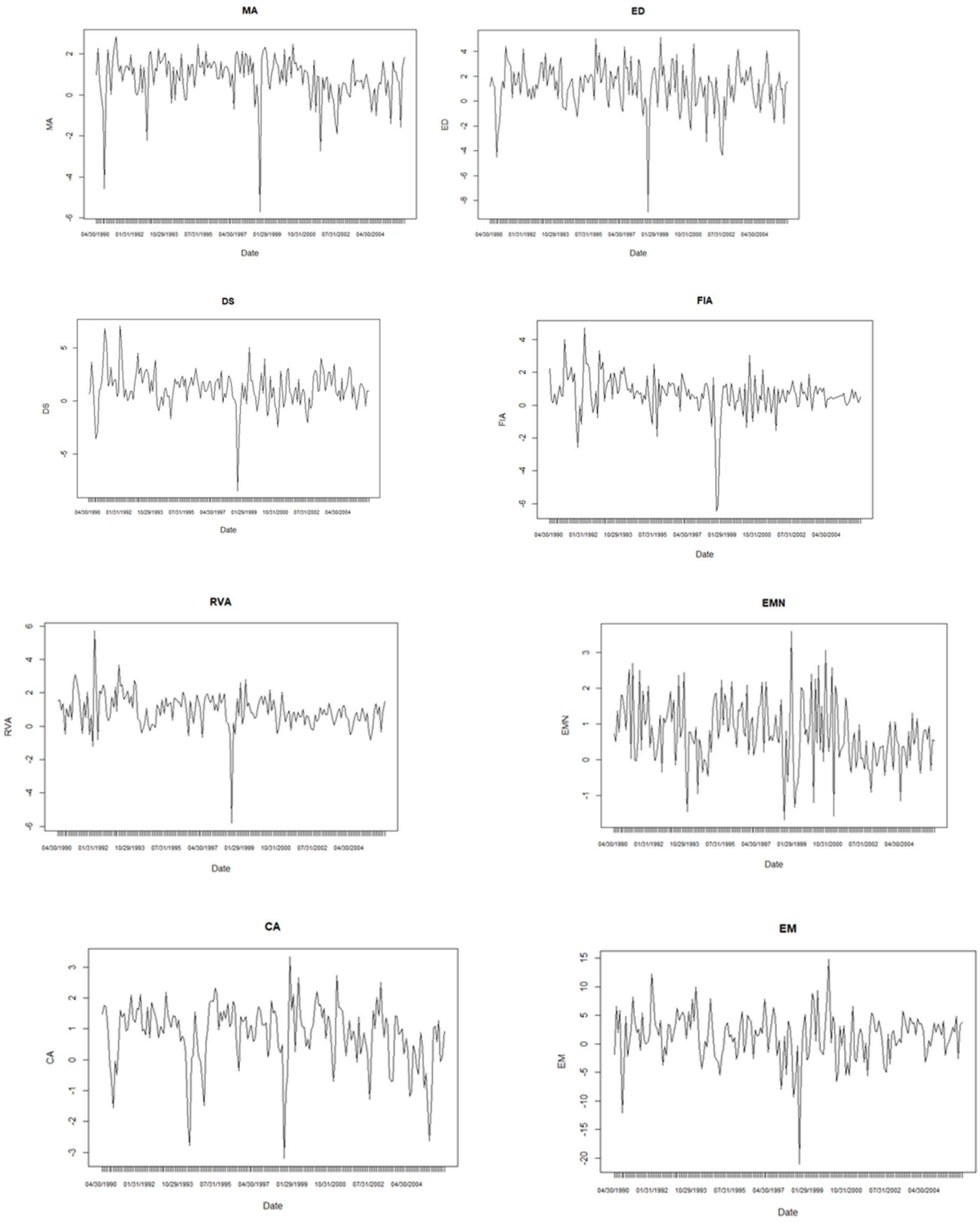


Figure 28: Time series plots of hedge fund returns RVA,ED,CA,DS,EMN,MA,EM,FIA .

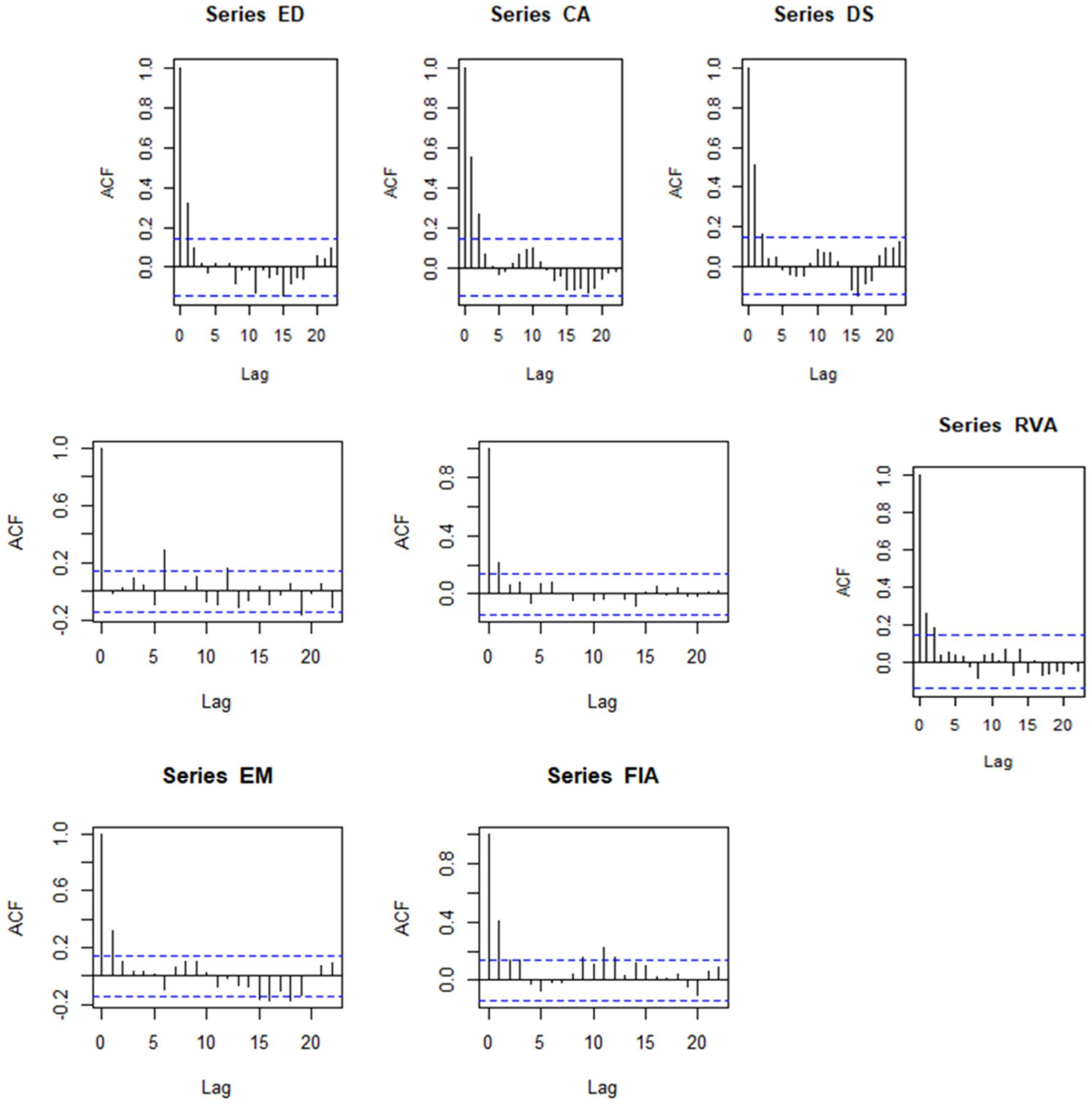


Figure 29: Autocorrelation plots of hedge fund returns RVA,ED,CA,DS,EMN,MA,EM,FIA.

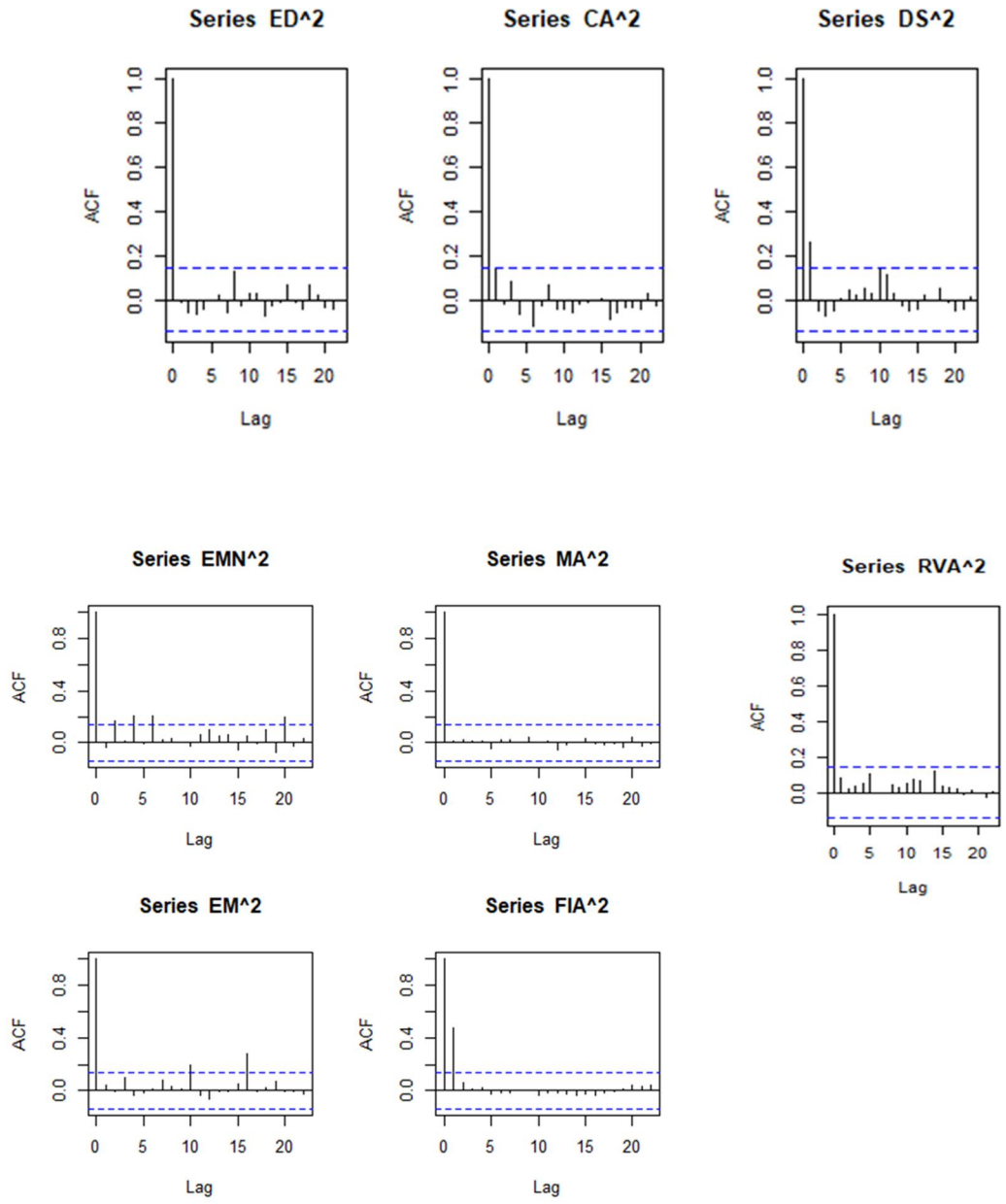


Figure 30: Autocorrelation plots of squared hedge fund returns RVA,ED,CA,DS,EMN,MA,EM,FIA.

Model	Parameter	MSE
Multiple Regression	—	**0.00048878**
LASSO	Default	**0.00041579**
Regression Tree	Default	**0.0002150527**
Random Forest	ntree=double(default)	**0.0002130655**
ARMA(1,1)	—	**0.0001168077**

Table 16: Predictive performance for monthly returns of RVA.

Model	Parameter	MSE
Multiple Regression	—	**0.0002568932**
LASSO	Default	**0.00004813217**
Regression Tree	Default	**0.0003187425**
Random Forest	ntree=default mtry=double(default)	**0.0002854698**
ARMA(1,1)	—	**0.00002826163**

Table 17: Predictive performance for monthly returns of ED.

Model	Parameter	MSE
Multiple Regression	—	**0.001687723**
LASSO	Default	**0.0001387342**
Regression Tree	Default	**0.001131879**
Random Forest	ntree=default mtry=double(default)	**0.001239145**
ARMA(1,1)	—	**0.0002998144**

Table 18: Predictive performance for monthly returns of CA.

Model	Parameter	MSE
Multiple Regression	—	**0.00002607**
LASSO	Default	**0.000007653**
Regression Tree	Default	**0.000030393**
Random Forest	ntree=default mtry=double(default)	**0.001349126**
ARMA(1,1)	—	**0.000029164**

Table 19: Predictive performance for monthly returns of DS.

Model	Parameter	MSE
Multiple Regression	—	**0.00001353652**
LASSO	Default	**0.002300635**
Regression Tree	Default	**0.00002436165**
Random Forest	ntree=default mtry=double(default)	**0.000008797805**
ARMA(1,1)	—	**0.00003846817**

Table 20: Predictive performance for monthly returns of EMN.

Model	Parameter	MSE
Multiple Regression	—	**0.000102590**
LASSO	Default	**0.00003150365**
Regression Tree	Default	**0.0003974161**
Random Forest	ntree=default mtry=double(default)	**0.0002654665**
ARMA(1,1)	—	**0.00008950412**

Table 21: Predictive performance for monthly returns of MA.

Model	Parameter	MSE
Multiple Regression	—	**0.0008091724**
LASSO	Default	**0.0001357999**
Regression Tree	Default	**0.0005079354**
Random Forest	ntree=default mtry=double(default)	**0.00004603417**
ARMA(1,1)	—	**0.000163776**

Table 22: Predictive performance for monthly returns of EM.

Model	Parameter	MSE
Multiple Regression	—	**0.00006072966**
LASSO	Default	**0.00001793125**
Regression Tree	Default	**0.0001393455**
Random Forest	ntree=default mtry=double(default)	**0.0001332506**
ARMA(1,1)	—	**0.000004272356**

Table 23: Predictive performance for monthly returns of FIA.

References

Papers

1) Akaike(1973), Introduction to Akaike (1973) Information Theory and an Extension of the Maximum Likelihood Principle

2) Alan O.Sykes, An Introduction to Regression Analysis, The Inaugural Coase Lecture, Chicago Working Paper in Law & Economics.

3) A.Seru (2016), Machine Learning and Applications in Finance and Macroeconomics, University of Chicago.

4)Ayon Dey(2016), Machine Learning Algorithms: A Review, Department of CSE, Gautam Buddha University,Greater Noida, Uttar Pradesh, India, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 1174-1179.

5)Arindam Chaudhuri (2012), Forecasting Financial Time Series using Multiple Regression, Multi Layer Perception, Radial Basis Function and Adaptive Neuro Fuzzy Inference System Models: A Comparative Analysis, Department of Computer Science Engineering, NIIT University.

6)B. Efron (1979), Bootstrap Methods: Another Look at the Jackknife, The Annals of Statistics, Volume 7, Number 1

7)Bollerslev (1986) , GENERALIZED AUTOREGRESSIVE CONDITIONAL HETEROSKEDASTICITY , University of California at San Diego.

8)D. Cochrane and G. H. Orcutt (1949), Application of Least Squares Regression to Relationships Containing Auto- Correlated Error Terms, Journal of the American Statistical Association, Vol. 44, No. 245 (Mar., 1949), pp. 32-61

9) D. Giannikis, I.Vrontos(2011), A Bayesian approach to detecting nonlinear risk exposures in hedge fund strategies. Journal of Banking and Finance 35 1399-1414.

10)Efron,Hastie and Tibshirani(2004), Least angle regression
Annals of Stats. Volume 32, Number 2 (2004), 407-499.

11) E.Soulas, D.Shasha (2013), Online Machine Learning Algorithms for Currency Exchange Prediction, NYU CS Technical Report.

12)Fischer, Thomas; Krauss, Christopher; Treichel, Alex (2018), Machine learning for time series forecasting - a simulation study, FAU Discussion Papers in Economics, No. 02/2018.

13)G.Bontempi, Y. Le Borgne, S. Taieb(2013), Machine Learning Strategies for Time Series Forecasting, Chapter in Lecture Notes in Business Information Processing.

14)G. Udny Yule(1926), Why do we Sometimes get Nonsense-Correlations between Time-Series?--A Study in Sampling and the Nature of Time-Series, Journal of the Royal Statistical Society Vol. 89, No. 1 (Jan., 1926), pp. 1-63

15)Gregory F. Lawler and Vlada Limic, Random Walk: A Modern Introduction

16)Guerard J., Introduction to Financial Forecasting in Investment Analysis.

17)Guerard J, Schwarz E.(2007), Quantitative Corporate Finance

18)Hoerl and Kennard(1970),Ridge Regression:Applications to non-orthogonal problems.

19)Hristos Tyralis * ID and Georgia Papacharalampous ID (2017), Variable Selection in Time Series Forecasting Using Random Forests, Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens.

20)Haavelmo,M.T.(1944).The probability approach in econometrics.Econometrica12(Supplement): 1–115.

21)Herman Wold(1938), Reviewed Work: A Study in Analysis of Stationary Time Series, Journal of the Royal Statistical Society, Vol. 102, No. 2 (1939), pp. 295-298

22)Jorge A. Chan-Lau(2017), Lasso Regressions and Forecasting Models in Applied Stress Testing , IMF Working Paper Institute for Capacity and Development

23)Leo Breiman (2001),Random Forests, Dept. of Statistics, University of California.

24)Mallows(1973),Some comments on Cp,Technometrics,p.661-675

25)Paul Thagard(1990), Philosophy and Machine Learning, CANADIAN JOURNAL OF PHILOSOPHY 261 Volume 20, Number 2, pp. 261-276.

26) Q.Qifeng, P.Beling (2016), Decision analytics and machine learning in economic and financial systems.

27) Robert F. Engle(2012), ARCH/GARCH Models in Applied Financial Econometrics

28) R.Savona (2013), Hedge Fund Systemic Risk Signals, SYRTO Working paper n.6.

29)Robert F. Engle (1982), Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation, Econometrica, Vol. 50, No. 4, pp. 987-1007

30)Spyridon D.Vrontos, Ioannis D.Vrontos,Daniel Giamouridis (2008), Hedge fund pricing and Model Uncertainty,Journal of Banking and Finance, p. 741-753.

31) Schwarz(1978), Estimating the Dimension of a Model, Annals of Statistics, Volume 6, Number 2 (1978), 461-464.

32)Thomas Hellstrom et. al(1998), Predicting the Stock Market, Center of Mathematical Modeling (CMM) Department of Mathematics and Physics Malardalen University.

33) Tibshirani(1996), Regression shrinkage and selection via the lasso, ,Department of Statistics Stanford University.

34)Turan G.Bali, Stephen J. Brown, Mustafa Onur Caglayan(2011), Do hedge funds' exposures to risk factors predict their future returns?, Journal of Financial Economics 101, 36-68.

35)Yigit Atilgan , Turan G. Bali , K. Ozgur Demirtas(2013) The performance of hedge fund indices Borsa _Istanbul Review 13 30e52.

Books and Websites

1)**https://www.hedgefundresearch.com/**

2)Ioannis Vrontos, Time Series and Forecasting Methods, MSc Data Science, Department of Informatics.

3)Narendra Bhogavalli(2018),Understanding Machine Learning, http://www.sqlservercentral.com/articles/machine+learning/167368/

4)Jason Brownlee, Master Machine Learning Algorithms, Discover How They Work and Implement Them From Scratch

5)James, Witten, Hastie, Tibshirani, Introduction to Statistical Learning

6)Ioannis Ntzoufras, Advanced Data Analysis,Postgraduate Course in Statistics

7)I. Vrontos, Stationary ARMA and Box-Jenkins methodology,Dept. of Informatics.

8) http://www.gfoa.org/financial-forecasting-budget-preparation-process/

9) Tom Mitchell (1997), Machine Learning , McGraw Hill


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ