Using a neural network approach for backtesting methodologies for estimating and forecasting asset risk

Christian Scherrer
Department of Statistics, Econometrics and Mathematical Finance, School of Economics and Business Engineering, University of Karlsruhe and KIT
Kollegium am Schloss, Bau II, 20.12, R210, Postfach 6980, D-76128, Karlsruhe, Germany
E-mail: scherrer@statistik.uni-karlsruhe.de, Phone +49-721-608-2042

Svetlozar T. Rachev
Chair-Professor, Chair of Statistics, Econometrics and Mathematical Finance, School of Economics and Business Engineering, University of Karlsruhe and KIT, and Department of Statistics and Applied Probability, University of California, Santa Barbara, and Chief-Scientist, FinAnalytica INC
Kollegium am Schloss, Bau II, 20.12, R210, Postfach 6980, D-76128, Karlsruhe, Germany
E-mail: rachev@statistik.uni-karlsruhe.de, Phone: +49-721-608-7535

Young Shin Kim
Department of Statistics, Econometrics and Mathematical Finance, School of Economics and Business Engineering, University of Karlsruhe and KIT
Kollegium am Schloss, Bau II, 20.12, R210, Postfach 6980, D-76128, Karlsruhe, Germany
E-mail: aaron.kim@statistik.uni-karlsruhe.de, Phone: +49-0721-608-6727

Michael Feindt
Professor, Chair of Experimentelle Teilchenphysik, University of Karlsruhe and KIT and Chief-Scientist, Phi-T GmbH
E-mail: Michael.Feindt@physik.uni-karlsruhe.de, Phone: +49-721-608-7378

Frank Fabozzi
Frank J. Fabozzi, Professor in the Practice of Finance, Yale School of Management
New Haven, CT USA
E-mail: frank.fabozzi@yale.edu, Phone +1-203-432-2421

Scherrer's research was supported by Phi-T GmbH. Rachev gratefully acknowledges research support by grants from Division of Mathematical, Life and Physical Sciences, College of Letters and Science, University of California, Santa Barbara, the Deutschen Forschungsgemeinschaft and the Deutscher Akademischer Austausch Dienst.

Abstract

In this paper we present a neural network approach for predicting a conditional probability density function (pdf) for the daily Dow Jones Industrial Average (DJIA) return. The conditional pdf is given by a user-defined amount of random numbers. We fit a classical tempered stable distribution (CTS) to the output which allows us to define a stochastic process and makes it possible to find a risk-neutral process. By investigating a large backtest (1987-2009) we compare the forecasts of two different neural network models with the performance of the normal-ARMA-GARCH, *t*-ARMA-GARCH, and CTS-ARMA-GARCH models.

Keywords: neural network, normal-ARMA-GARCH, *t*-ARMA-GARCH, CTS-ARMA-GARCH, backtest, forecast, time series
JEL Classifications:

1 Introduction

A key task of risk managers and asset managers is to estimate and forecast the risk of an asset. One of the most common approaches for doing so is the mathematical modeling of a time series with the help of a generalized autoregressive conditional heteroscedasticity (GARCH)-type model. After choosing the type of model one has to employ statistical tests to assess if the historical data can be described accurately by the selected model. The next step is to perform a backtest (i.e., a simulation) using all information available until day i to forecast the next daily return. Since the real return for an asset is known, it is possible to compare the prediction and the actual value and then decide if the model is sufficiently reliable so as to be useful in predicting future returns. This test is necessary to exclude trivial models which have enough parameters to learn a complete historical data sample but which are capable of real forecasting.

Many studies have shown that the assumption of a normal distribution for the residuals of a GARCH-model is inappropirate because asset returns are generally skewed and have a nonzero kurtosis (see, among others, Menn and Rachev (2005a) and Menn and Rachev (2005b)). An alternative to a Gaussian is the class of tempered stable distributions (CTS) and will be the distribution used in this paper. The definition of this type of distribution is given in Rosinski (2007). The CTS has been used for the residuals of a GARCH model in Kim *et al.* (2008a) and Kim *et al.* (2008b).

In this paper, we propose five models and compare their performance using a large backtest in which the daily returns of the Dow Jones Industrial Average (DJIA) are predicted from 1987 until 2009. We employ as our benchmark a normal-ARMA-GARCH and a *t*-ARMA-GARCH model to define a process for the daily returns. Additionally, we use a CTS-ARMA-GARCH model which has the advantage of describing the skewness and fat tails that has been observed for assets in numerous studies. The remaining two models are two neural network models which take into account volatility clustering, another stylized fact observed about asset prices.

Originally, neural networks were only used for classification problems. In this case, the network gets a data sample in which for an event i the input vector $\overrightarrow{x_i}$ and the target t_i are specified. The target can be a signal ($t_i = 1$) or a background ($t_i = -1$). The network is then trained with these samples; that is, the network can learn and get experience from examples and can afterwards be used to predict the probability of an event being a signal under the condition that the input vector is given as $\overrightarrow{x_i}$.

Neural networks are especially useful when there is a large number of historical events in which an input vector and a target variable are provided. They can approximate a universal function by an arbitrary grade of acuracy (see among others Irie and Miyake (1988)) and therefore it is possible to learn nonlinear correlations between input variables and target. Although neural networks were originally used for classification problems, the problem with a time series of daily returns is that we cannot define a return as a signal or a background because the returns are continuous and cannot be described by a binary decision. In Weigend and Srivastava (1995) and Feindt (2004), this problem was solved using several output nodes. While in Weigend and Srivastava (1995) they used the output nodes to model the probability density function (pdf), in Feindt (2004) they fit the cumulative distribution function (cdf). With this improvement, it is possible to predict a conditional probability density function for the return of an asset given some information at day i. The output of the neural network is given by random numbers which are generated from the pdf. These random numbers are fitted by a CTS distribution which is used to define a stochastic process. If one has defined the process, a risk-neutral process can be found so that the model can be used within the framework of arbitrage pricing theory Kim *et al.* (2008a).

The aim of this paper is to show how a neural network model can be used as an alternative to GARCH models. In Section 2 we explain the details of the neural network model and in Section 3 we explain the CTS distribution. How to construct a reasonable backtest is provided in Section 4. In Section 5 we discuss the results of our neural network model and compare them to the three ARMA-GARCH models. Section 6 concludes the paper. A brief introduction to the NeuroBayes[1] software that is used to generate our results is provided in the appendix.

2 The neural network model

The goal in designing a risk management methodology using a neural network is to be able to predict the probability density function for the daily DJIA return $f(r_i|\overrightarrow{x_i})$. To do so requires several steps which we discuss here.

[1] Developed by Phi-T® Physics Information Technologies GmbH

2.1 Definition of the target

First, a target variable for the network must be defined. It is easier for the network to learn a distribution which does not have extreme outliers. Therefore, the idea is not to use directly the return as a target variable. Instead, a transformed return must be computed. That means if we would like to predict the return, we would get the density of the transformed return from the neural network. Then a back transformation from this target variable must be computed in order to obtain the return again.

The transformation just normalizes the daily returns. This is done by dividing the daily returns by volatility. So we have

$$\tilde{r}_i = \log\left(\frac{\text{close}_{i+1}}{\text{close}_i}\right)\frac{\sqrt{253}}{\sigma_i} \tag{1}$$

where σ_i is an estimate for the yearly volatility derived from the daily history of the DJIA, including the close of day i. Failure to normalize the returns means that the neural network would also have to learn the volatility clustering, which is not easy.

It is important to note that σ_i only includes information up to day i which means that no information of the future is included. If we had included information of the future in the volatility, we would have had to predict the distribution of the return and the distribution of the volatility. That means it is best to define a target which has similar properties for each and every day but which does not include more than one variable which embodies future information.

The target is then defined by

$$t_i = \tilde{r}_i \tag{2}$$

2.2 Definition of the input vector

In the second step the input vector from which the network can learn the conditional probability density function must be defined. We take into account for the daily returns of the DJIA the open, high, low, and close, and from these data we construct input variables for every trading day. For example at day i complete information of the past up to day i can be used. That means the latest information we use is given by the close, open, high, and low of day i. From the time series up to day i we construct 51 input variables for the neural network. These input variables include many well-known technical indicators which can be found in Colby (2002) and Achelis (2000). In addition, we also use variables we constructed (e.g., coefficients of wavelets or combined variables of technical indicators).[2] The input variables for the neural network models computed by using different time series algorithms of Phi-T® are:

- 14 variables from wavelet analysis
- 12 variables constructed from combinations of $high_i$, low_i, $close_i$, $open_i$, $high_{i-1}$, low_{i-1}, $close_{i-1}$, $open_{i-1}$
- 6 variables constructed from $close_i$, ..., $close_{i-k}$
- 6 different volatilities (using moving averages, exponential moving averages, Wilder's volatility)
- 4 relative strength indices defined on different time intervals
- 4 different combinations of moving averages defined on different time intervals
- 2 variables which include different stochastic oscillators
- 1 variable using Bollinger bands
- 1 variable constructed from the Moving Average Convergence/Divergence indicator
- 1 variable using Williams %R indicator

2.3 How to forecast daily returns

After defining the input vector and the target, we can train our network and try to predict future returns. We would like to have a prediction for

$$r_i = \log\left(\frac{\text{close}_{i+1}}{\text{close}_i}\right) \tag{3}$$

That means close_{i+1} is future information and close_i is known. But we also know high, low, and open of day i. That means we can use all data in our sample up

[2]The variables are used in a commercial model at Phi-T® and cannot be described in detail due to their proprietary nature.

to day i to train the network and adjust its weights. Then we compute the input vector $\overrightarrow{x_i}$. We insert this input vector into the network and get a forecast for t_i which is defined in (2).

The NeuroBayes software is able to predict random numbers of t_i, so we have to do the back transformation of (1) to get the density for r_i.

3 The classical tempered stable distribution

From the NeuroBayes software we get a prediction of the probability density function of the daily DJIA return. But without defining a stochastic process, we are not able to define a risk-neutral process. For this purpose we fit a classical tempered stable distribution to the random numbers generated by NeuroBayes. In this way, we can define a stochastic process. In Kim *et al.* (2009) the classical tempered stable distribution is reviewed including a proof that there exists an equivalent martingale measure. Consequently, we are operating in the framework of the arbitrage pricing theory (APT).

In this section we summarize what we need to understand in order to use the classical tempered stable distribution. First, let's provide the definition of a CTS:

> *An infinitely divisible random variable X is said to follow the classical tempered stable distribution if its characteristic function is given by*
>
> $$\begin{aligned}\Phi_X(u;\alpha,C_1,C_2,\lambda_+,\lambda_-,m) &= \\ \exp(ium &+ C_1\Gamma(-\alpha)((\lambda_+-iu)^\alpha-\lambda_+^\alpha) \\ &+ C_2\Gamma(-\alpha)((\lambda_-+iu)^\alpha-\lambda_-^\alpha))\end{aligned}$$
>
> *where* $C_1, C_2, \lambda_+, \lambda_- > 0$, $\alpha \in (0,2)$ *and* $m \in \mathbb{R}$. *A Lévy process induced from the CTS distribution is called a classical tempered stable process with parameters* $(C_1, C_2, \lambda_+, \lambda_-, m)$.

If

$$C = C_1 = C_2 = (\Gamma(2-\alpha)(\lambda_+^{\alpha-2}+\lambda_-^{\alpha-2}))^{-1} \tag{4}$$
$$m = -\Gamma(1-\alpha)(C_1\lambda_+^{\alpha-1}-C_2\lambda_-^{\alpha-1}) \tag{5}$$

is fulfilled then $X \sim \text{CTS}(\alpha, C_1, C_2, \lambda_+, \lambda_-, m)$ has zero mean and unit variance and we call X the *standard CTS distribution* denoted by $\text{stdCTS}(\tilde{\alpha}, \tilde{\lambda}_+, \tilde{\lambda}_-)$ where we use the tilde to symbolize the parameters of the standard CTS distibution.

If we use the definition of the characteristic function in (4) we can define the cumulants $c_n(X) := \frac{1}{i^n}\frac{d^n}{du^n}\log(E[e^{iuX}])$:

$$c_n(X) = \begin{cases} m + \Gamma(1-\alpha)(C_1\lambda_+^{\alpha-1} - C_2\lambda_-^{\alpha-1}), & \text{for } n = 1 \\ \Gamma(n-\alpha)(C_1\lambda_+^{\alpha-n} + (-1)^n C_2\lambda_-^{\alpha-n}), & \text{for } n = 2, 3, ... \end{cases} \tag{6}$$

We use NeuroBayes to obtain random numbers of the returns and normalize them by the transformation

$$\tilde{r}_i = \frac{r_i - \mu_i}{\sigma_i}$$

where i denotes the day in the time series, μ_i is the mean of the random numbers, and σ_i is the root mean square. We fit the stdCTS probability density function to the transformed random variables r_i, so we have to estimate the parameters $(\tilde{\alpha}, \tilde{\lambda}_+, \tilde{\lambda}_-)$. The fit is done using maximum likelihood maximization where the pdf is computed by Fast Fourier Transformation (FFT).

After estimating the three parameters, we get the parameters of the non-normalized CTS distribution using

$$\begin{aligned} \alpha &= \tilde{\alpha} \\ C &= \sigma^{\tilde{\alpha}}\tilde{C} \\ \lambda_+ &= \frac{\tilde{\lambda}_+}{\sigma} \\ \lambda_- &= \frac{\tilde{\lambda}_-}{\sigma} \\ m &= \mu \end{aligned} \tag{7}$$

where $\tilde{C}$ is defined in (4) using the parameters $(\tilde{\alpha}, \tilde{\lambda}_+, \tilde{\lambda}_-)$. These transformations can easily be proven by using the definition of the cumulants in (6).

4 ARMA-GARCH models

In this section we describe the ARMA-GARCH models which we use in the backtest to compare with the neural network models. In an ARMA(p, q)-

GARCH(r, s) model the log-returns are assumed to have the following dynamic:

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{i=1}^{q} b_i \epsilon_{t-i} + \epsilon_t \sigma_t$$

where $y_t = \log(r_t)$ and the residuals ϵ_t are independent and identically distributed and defined by

$$\epsilon_t^2 = \alpha_0 + \sum_{i=1}^{r} \alpha_i \sigma_{t-i}^2 \epsilon_{t-i}^2 + \sum_{i=1}^{s} \beta_i \epsilon_{t-i}^2$$

In the *t*-ARMA-GARCH model, the residuals follow a Student *t*-distribution while in the normal-ARMA-GARCH model they are assumed to be normally distributed. In the case of the CTS-ARMA-GARCH model, we use the classical tempered stable distribution which was discussed in Section 3.

When working with the CTS-ARMA-GARCH models, the first step is to fit the normal-ARMA-GARCH model to the time series to obtain the parameters a_i, b_i, α_0, α_i, and β_i. In the second step, we fit the standard classical tempered distribution to estimate the parameters $\tilde{\alpha}$, $\tilde{\lambda}_+$, and $\tilde{\lambda}_-$. Maximum likelihood estimation is used to obtain the fits.

5 How to define a reasonable backtest

In backtesting our model, we encounter the problem of forecasting a probability density function for one event while having only one actual realization of that event. The forecasted density is dependent on the input vector which we insert in the neural network which again depends on day i. That means that the density itself is also dependent on day i.

One generally accepted possibility to check the forecasted densities is to count exceedences of the return with respect to a specific quantile. In this case the most common choices for the quantile are 1% or 5%. The idea is then that in an infinitely large sample, the exceedences of the actual return with respect to the 1% quantile of the forecasted distribution should converge to the actual probability (i.e., to 1%). But if we only looked at this quantile, we would fail to be using most of the statistics. So it would be much more reasonable to look at many quantiles.

When counting the exceedences what we basically do is forecast the density of an event and measure its actual return of it. Then the cdf of the actual return which is defined in the interval $[0, 1]$ can be computed. The pdf is dependent on the event but the cdf for the actual returns z_i should always be a uniform distribution independent of the chosen event; that is,

$$z_i = \int_{-\infty}^{r_{actual}^{(i)}} dr \; f(r|\overrightarrow{x_i}) \quad z_i \in [0, 1] \tag{8}$$

Therefore, it is reasonable to check if the cdf of the actual event results in a uniform distribution (if all quantiles of our forecasted distribution are correct). For a further discussion, see Campbell (2006) and the references therein.

So far we explained how we can test if our density is correct, but we still have not quantified how we really measure whether the cdf of the actual returns is uniform and how we can compare different models. For this purpose, we turn to the Kolmogorov-Smirnov and the Anderson-Darling tests. The basic idea in both tests is the same. One involves computing the theoretical cdf of the actual returns and the empirical cdf and then takes the maximal distance between them. This distance is a random variable on which one can decide if the model is accepted or rejected. The Anderson-Darling test works in the same way but one assigns a higher weight to the correctness of the tails.

The same idea is improved upon in a methodology suggested by RiskMetrics (see Zumbach (2006)). The methodology involves first introducing the variable

$$\delta(z) = \text{cdf}_{emp.}(z) - z \qquad z \in [0, 1] \tag{9}$$

which is the difference between the empirical and the theoretical cdfs. If the model is correct, $\delta(z)$ should converge to 0 for all values of z for an infinitely large sample. So one possibility to compare different models is to plot $\delta(z)$ with the better model being the one with the smaller absolute values for the deltas.

The second suggestion by RiskMetrics is to construct a scalar from the computed deltas which is a measure for the correctness of the entire cdf. For this purpose the variable d_p is introduced where p is a parameter with which different weight can be given to the tails

$$d_p = \int_0^1 dz \mid \delta_p(z) \mid \tag{10}$$

and δ_p is given by

$$\delta_p(z) = \delta(z)(p+1)2^p \mid z - \frac{1}{2} \mid^p \tag{11}$$

If $p = 0$, this is just the integral over all absolute values of $\delta(z)$. The larger the computed p, the greater the importance of describing the tails accurately.

6 Results of the backtest and comparison with different GARCH models

In this section, we present the results of the backtest. First, we downloaded all available data (1930-2009) for the time series for the Dow Jones Industrial Index (DJIA) from www.finance.yahoo.com. For the forecast we chose the years from 1987 until 2009. Our selection of 1987 was because it included Black Monday (October 19, 1987), the largest one day decline in the DJIA in stock market history since 1929.

We use all data up to the last day in 1986 and train our neural network to adjust the parameters. We predict the pdf for the first return in 1987 and compute the cdf of the actual return as described in the previous section. The next step is to predict the second return. In principle, we now have one more event in the historical data (the first return in 1987). That means we could train the network again with the same data sample as in the first training plus this event. Since the computational effort would be quite large if we did a new training for each trading day, we decided to compromise by doing a new training after one month has passed. That means to predict January 1987, we use a training which includes data from 1930 until December 1986. After this month, we train the network again with data from 1930 until the end of January 1987 and so on.

The backtest is done for the following five models:

1. *normal-ARMA-GARCH*: ARMA(1,1)-GARCH(1,1) model with standard normal distributed innovations
2. *t-ARMA-GARCH*: ARMA(1,1)-GARCH(1,1) model with *t*-distributed innovations
3. *EWMA-CTS-nn*: Neural network with exponentially weighted moving average volatility in the definition of the target and a fit of a CTS-distribution to the output of the network
4. *GARCH-CTS-nn*: Neural network with historical volatility from normal-ARMA-GARCH in the definition of the target and a fit of a CTS-distribution to the output of the network
5. *CTS-ARMA-GARCH*: ARMA(1,1)-GARCH(1,1) model with classical tempered stable distributed innovations

In the GARCH-models, the computational costs are low and the estimation of the parameters is done for every trading day.

In Section 5 we explained that the distribution of the cdf of the actual returns should be a uniform distribution if the model were acceptable. Therefore, we compare these plots for our different models. In Figure 1 we can see the results for the normal-ARMA-GARCH model and the *t*-ARMA-GARCH model. As expected, the cdf of the tails of the actual returns is much heavier than that predicted by the normal distribution. In particular, the large losses are not described very well. In the central area there are more events than described by the normal distribution.

In case of the *t*-ARMA-GARCH model, the tails (especially the right tails) seem to be too fat while the central area is underestimated as observed for the normal-ARMA-GARCH model. The missing property of both models is skewness. But assets exhibit the typical property of being skewed to the left.

The results of the neural network models are presented in Figure 2. We can see immediately that the distributions are quite uniform and that the central area and the fat tails are described well. The asymmetry which can be seen in the plots of the two GARCH models studied does not appear in the neural network models.

In case of the CTS-ARMA-GARCH model (see Figure 3), the cumulative distribution of the actual return is also quite uniform and the tails and the asymmetry are described well.

The quantile-quantile-plots of the cdf distribution of the actual return are presented in the Figures 4, 5, and 6. Again we can see that the asymmetry is a very important missing property of the normal-ARMA-GARCH and the *t*-ARMA-GARCH model.

Next we compare the δ_p's, defined in (11) to quantify the forecasting abilities of the different models. For $p = 0$, no special weight is given to the tails and we end up with Figure 7. These are just the differences between the theoretical and empirical cdfs. Again we see that the normal-ARMA-GARCH and the *t*-ARMA-GARCH models are comparable while both neural network models and the CTS-ARMA-GARCH model are much better because the absolute differences between the theoretical and the empirical cdf are much smaller.

To investigate the tail properties of the models, we follow RiskMetrics and plot δ_p defined in (11) for $p = 32$ (see Figures 8 and 9). Again we see that the left tail of the normal-ARMA-GARCH model is not heavy enough while the *t*-ARMA-GARCH model has a left tail that is too heavy. For the left tail the CTS-ARMA-GARCH model is the best one while the right tail is described best by the GARCH-CTS-nn model.

In order to have a scalar which characterizes the complete model we followed again RiskMetrics and computed d_0 and d_{32} as defined in (10). The results are reported in Table 1. Note that d_0 and d_{32} are both random numbers which include a statistical uncertainty. Again, we see that both neural network models and the CTS-ARMA-GARCH model clearly outperform the normal-ARMA-GARCH and the *t*-ARMA-GARCH models. The central area is similar in both neural network models and the CTS-ARMA-GARCH model, while the tails are described best in the GARCH-CTS-nn model and the CTS-ARMA-GARCH model. The GARCH-CTS-nn model is even slightly better than the EWMA-CTS-nn model.

In order to see how well the models perform in different time intervals, we computed d_0 and d_{32} for every year. The results are summarized in Table 2 and Table 3. The normal-ARMA-GARCH model is the worst performing model for the estimation of the tails while the *t*-ARMA-GARCH does not perform well in the central area. From this analysis we see that in general one should use advanced neural network models as well as advanced ARMA-GARCH models (such as CTS-ARMA-GARCH) to forecast the risk of an asset.

The markets in the years 2007 and 2008 were dominated by the financial crisis popularly referred to as the subprime mortgage crisis. Therefore, we plot the 1% value-at-risk (VaR) measure of all models in Figures 10 to 14. Especially in September and October 2008, the tail losses increased dramatically. The black dots symbolize the VaR violations. Again we see that the tails in the normal-ARMA-GARCH model are too thin while they are extremely fat in the *t*-ARMA-GARCH model.

We summarize these violations in Table 4. The numbers of violations that are bolded are consistent with the 95% confidence interval of the Kupiec test (Kupiec (1995)). In this test we would reject the normal-ARMA-GARCH model for both years and both neural network models for the year 2007, while we would accept the *t*-ARMA-GARCH and the CTS-ARMA-GARCH model. [3]

[3] A natural question to ask is why the neural network models were not able to outperform the CTS-ARMA-GARCH model. To answer this question, one has to understand how the NeuroBayes software is able to create a pdf without assuming an analytical function. In principle, NeuroBayes reconstructs the cdf from a neural network with 20 output nodes. But using 20 nodes means that it is necessary to approximate the cdf between the nodes, adding further uncertainty to the model.

7 Conclusions

Forecasting time series is one of the most important tasks in finance. One of the most common approaches for forecasting is the application of GARCH models in which one has to explicitly assume a probability density function for the residuals explicitly. This is not necessary when employing neural network models since the network can learn the probability density function based on historical data. In order to make sure that modeling is with the APT framework, we fitted a classical tempered stable distribution to the output of the neural network.

In a large backtest with a time span of more than 20 years, our results show that our neural network model is able to forecast the time series of the DJIA with amazing precision and that it can outperform a *t*-ARMA-GARCH and a normal-ARMA-GARCH model. Our findings suggest that the neural network produces a similar prediction in the central area and in the tails, and the results are comparable to the results of a CTS-ARMA-GARCH model.

The forecasting abilities of a model is dependent on the time horizon which is used for the backtest. There is no universal model which performs well over every time period. When forecasting the risk of an asset one should use different advanced neural network models as well as advanced ARMA-GARCH models (such as CTS-ARMA-GARCH).

A The NeuroBayes software

The NeuroBayes[4] software is based on the idea of neural networks. We summarize the most important ideas of the package:

- The input variables are fitted by robust splines to regularize statistical irrelevant ourliers. Here the user has to assign a so called preprocessing flag to every input variable. The flag specifies the variable being discrete or continuous and the type of fit which is used. The fitted variables are used in the neural network.
- NeuroBayes uses the variable which has the highest correlation to the target and rotates the remaining variables in a way which ensures that this part of the information is removed from the rest of the variables. Then it proceeds stepwise with the rest of the variables. Without this algorithm the same information would be used again and again which would have no statistical relevance and would lead to overtraining.
- The pdf is constructed from a neural network with 20 output nodes which are used for a classification problem. This introduces a discretization error but it has the advantage of using a nonparametrical pdf.

For further information, especially about the density construction, see Feindt (2004).

[4] Developed by Phi-T® Physics Information Technologies GmbH

References

Achelis, S. B. (2000). *Technical Analysis from A to Z*. McGraw-Hill.

Campbell, S. D. (2006). A review of backtesting and backtesting procedures. *Journal of Risk*.

Colby, R. W. (2002). *The Encyclopedia Of Technical Market Indicators*. McGraw-Hill.

Feindt, M. (2004). A Neural Bayesian Estimator for Conditional Probability Densities, http://www.citebase.org/abstract?id=oai:arXiv.org:physics/0402093.

Irie, B. and Miyake, S. (1988). Capabilities of three-layered perceptrons, Proceedings of the IEEE International Conference of Neural Networks.

Kim, Y. S., Rachev, S. T., Bianchi, M. L., and Fabozzi, F. J. (2008a). Financial market models with Lévy processes and time-varying volatility. *Journal of Banking and Finance*, *32*(7), 1363–1378.

Kim, Y. S., Rachev, S. T., Bianchi, M. L., and Fabozzi, F. J. (2009). Tempered stable and tempered infinitely divisible GARCH models. Technical report, Chair of Econometrics, Statistics and Mathematical Finance School of Economics and Business Engineering University of Karlsruhe (https://www.statistik.uni-karlsruhe.de/download/doc secure1/RDTS-GARCH-MathFin.pdf).

Kim, Y. S., Rachev, S. T., Chung, D. M., and Bianchi, M. L. (2008b). The modified tempered stable distribution, GARCH-models and option pricing. *Forthcoming in Probability and Mathematical Statistics*.

Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*.

Menn, C. and Rachev, S. T. (2005a). A garch option pricing model with [alpha]-stable innovations. *European Journal of Operational Research*, *163*(1), 201–209.

Menn, C. and Rachev, S. T. (2005b). Smoothly truncated stable distributions, GARCH-models, and option pricing. Technical report, Chair of Econometrics, Statistics and Mathematical Finance School of Economics and Business Engineering University of Karlsruhe (http://www.statistik.uni-karlsruhe.de/download/tr_smoothly_truncated.pdf).

Rosinski, J. (2007). Tempering stable processes. *Stochastic Processes and Their Applications*.

Weigend, A. S. and Srivastava, A. N. (1995). Predicting conditional probability distributions: a connectionist approach. *International Journal of Neural Systems*, *6*(2), 109–118.

Zumbach, G. (2006). Backtesting Risk Methodologies from One Day to one Year, Tech. Report, Risk Metrics Group.

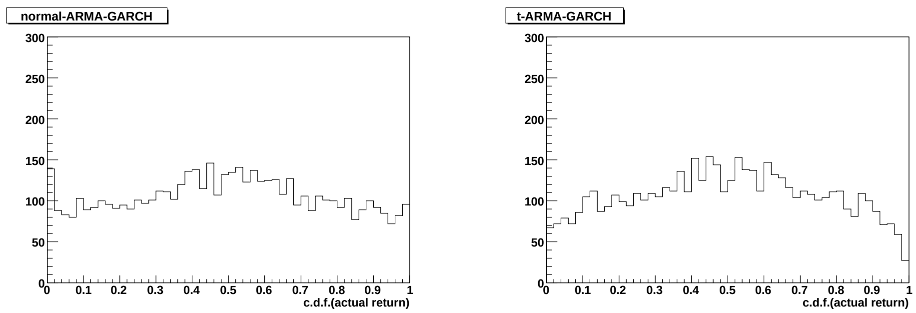

Figure 1: Cumulative distribution function of the actual return using the normal-ARMA-GARCH model and the *t*-ARMA-GARCH model.

This figure reports the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a normal-ARMA-GARCH model (left) and a *t*-ARMA-GARCH model (right).

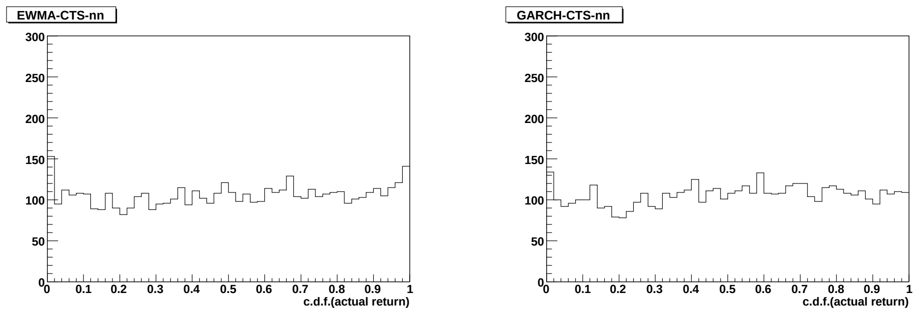

Figure 2: Cumulative distribution function of the actual return using the models EWMA-CTS-nn and GARCH-CTS-nn.

This figure reports the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a neural network model with an exponentially weighted moving average definition for the volatility (left) and a neural network model using a volatility computed by a normal-ARMA-GARCH model (right). The outputs of the neural network models are fitted by a classical tempered stable distribution.

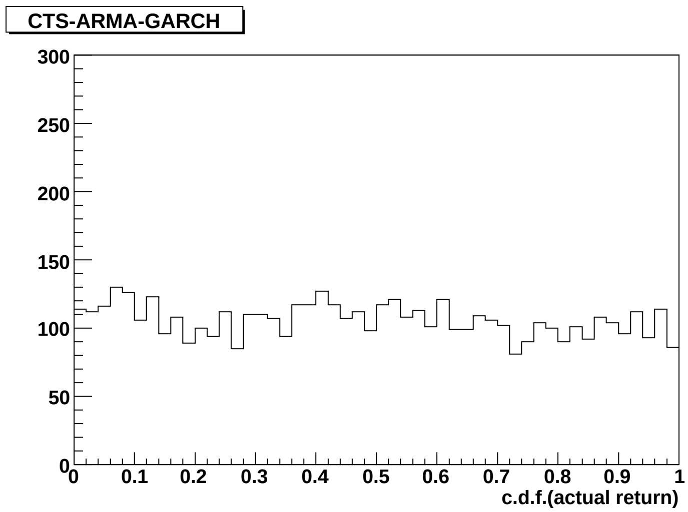

Figure 3: Cumulative distribution function of the actual return using the CTS-ARMA-GARCH model.

This figure reports the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a CTS-ARMA-GARCH model.

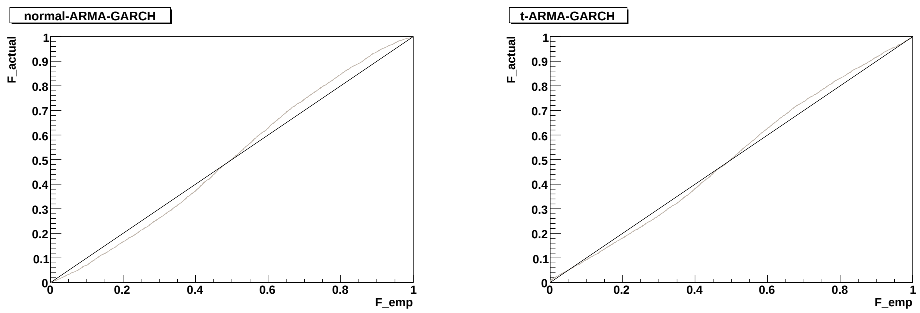

Figure 4: Quantile-Quantile-plot of the cumulative distribution function of the actual return using the normal-ARMA-GARCH and the *t*-ARMA-GARCH model.

This figure reports the Quantile-Quantile-plot of the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a normal-ARMA-GARCH model (left) and a *t*-ARMA-GARCH model (right).

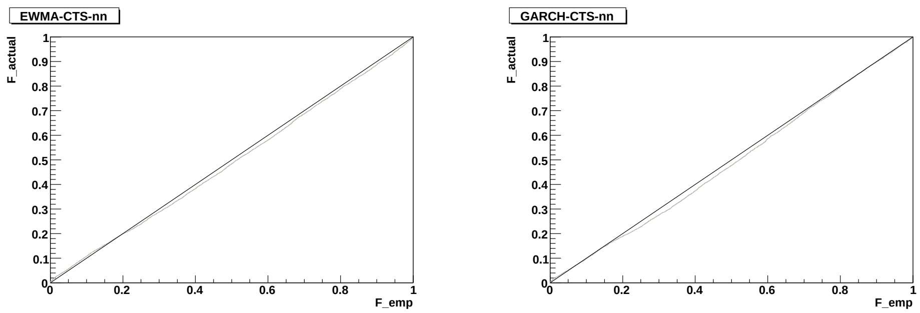

Figure 5: Quantile-Quantile-plot of the cumulative distribution function of the actual return using the EWMA-CTS-nn and the GARCH-CTS-nn model.

This figure reports the Quantile-Quantile-plot of the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a neural network model with an exponentially weighted moving average definition for the volatility (left) and a neural network model using a volatility computed by a normal-ARMA-GARCH model (right). The outputs of the neural network models are fitted by a classical tempered stable distribution.

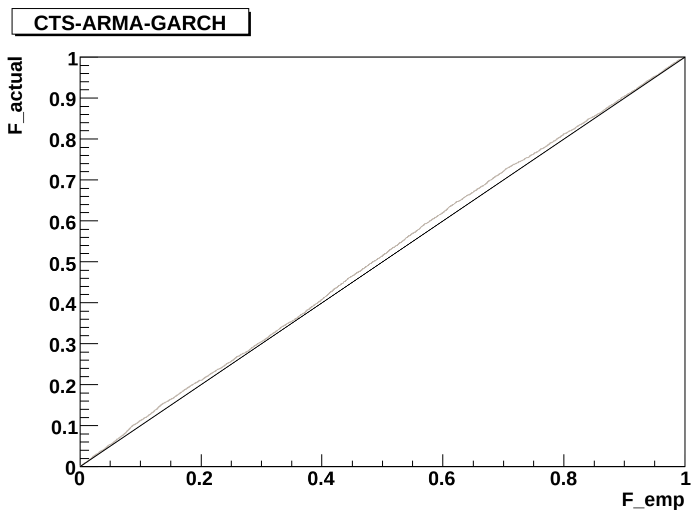

Figure 6: Quantile-Quantile-plot of the cumulative distribution function of the actual return using the CTS-ARMA-GARCH model.

This figure reports the Quantile-Quantile-plot of the cumulative distribution function of the actual returns of the Dow Jones Industrial Average (1987-2009) using the forecasted probability density function of a CTS-ARMA-GARCH model.

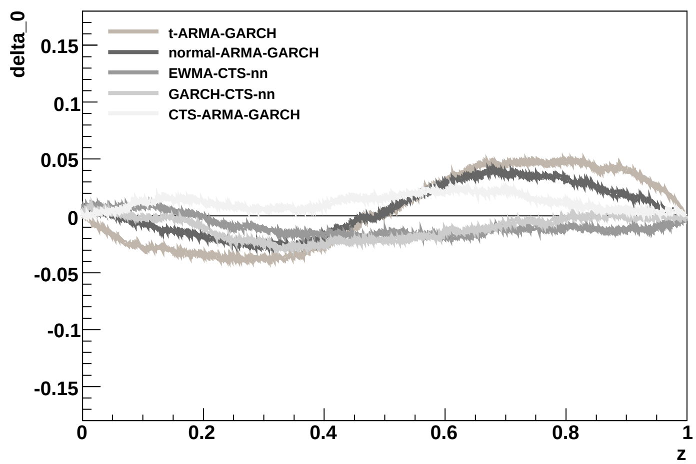

Figure 7: Difference between empirical cdf and theoretical cdf for all models (giving no special weight to the tails).

This figure reports the difference between the theoretical and the empirical cumulative distribution functions of the actual returns of the Dow Jones Industrial Average (1987-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility using a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution.

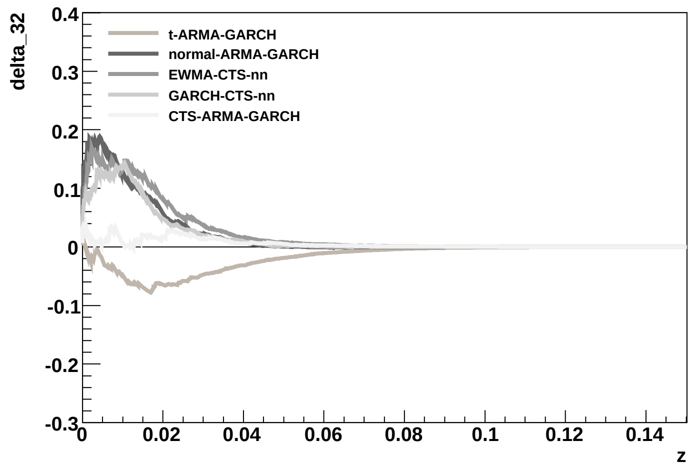

Figure 8: Difference between empirical cdf and theoretical cdf for all models (giving more weight to the tails).

This figure reports the difference between the theoretical and the empirical cumulative distribution functions of the actual returns of the Dow Jones Industrial Average (1987-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. A higher weight is given to the left tails. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility using a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution.

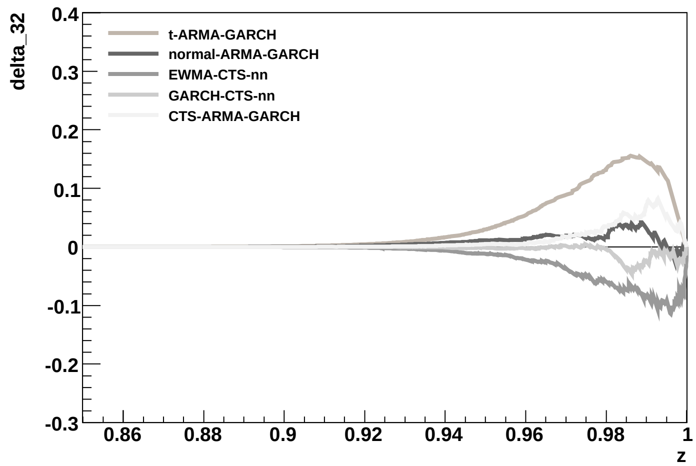

Figure 9: Difference between empirical cdf and theoretical cdf for all models (giving more weight to the tails).

This figure reports the difference between the theoretical and the empirical cumulative distribution functions of the actual returns of the Dow Jones Industrial Average (1987-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. A higher weight is given to the right tails. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility using a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution.

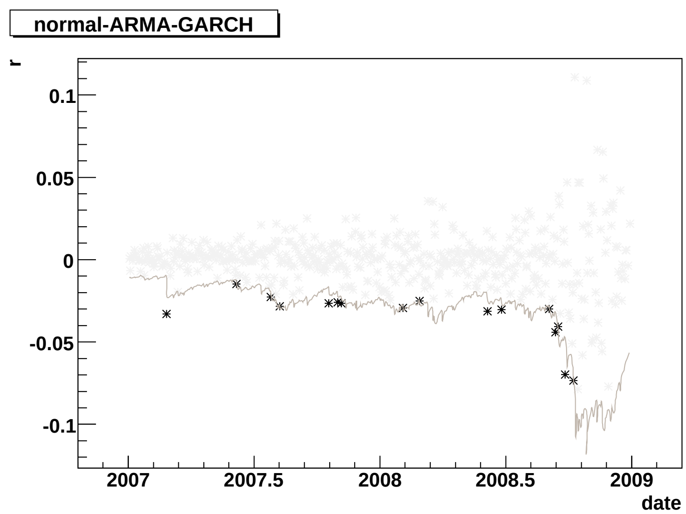

Figure 10: Value at risk for the normal-ARMA-GARCH model.

This figure reports the estimation of value at risk for the return of the Dow Jones Industrial Average (2007-2009) using a normal-ARMA-GARCH model. The black dots symbolize the VaR violations.

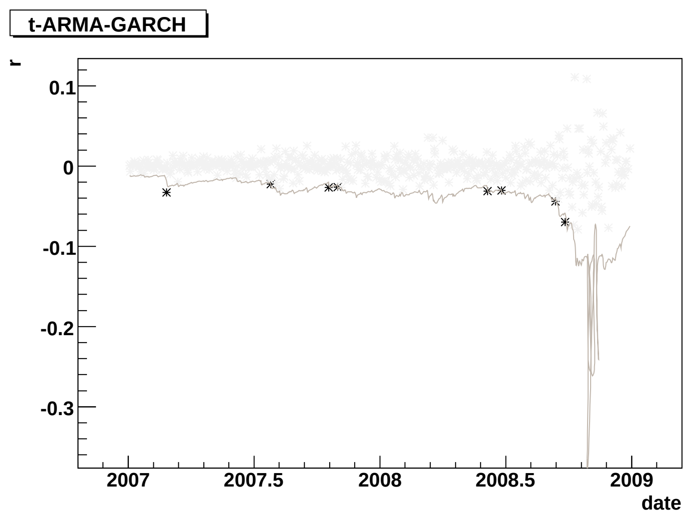

Figure 11: Value at risk for the *t*-ARMA-GARCH model.

This figure reports the estimation of value at risk for the return of the Dow Jones Industrial Average (2007-2009) using a *t*-ARMA-GARCH-model. The black dots symbolize the VaR violations.

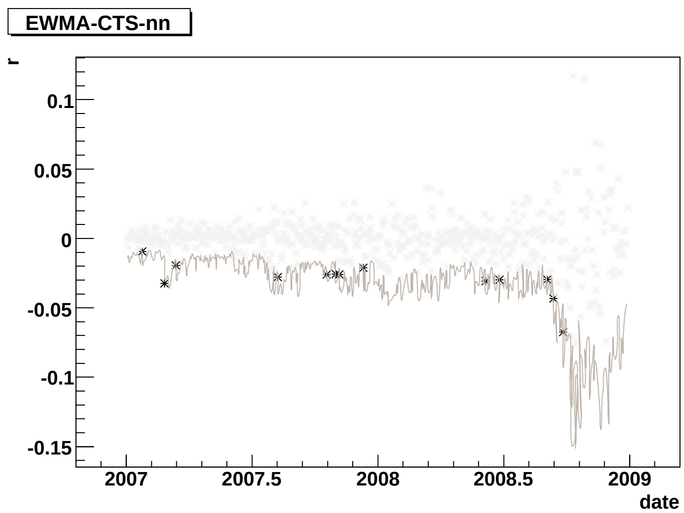

Figure 12: Value at risk for the EWMA-CTS-nn model.

This figure reports the estimation of value at risk for the return of the Dow Jones Industrial Average (2007-2009) employing a neural network model using an exponentially weighted moving average defition for the volatility. The outputs of the neural network are fitted by a classical tempered stable distribution. The black dots symbolize the VaR violations.

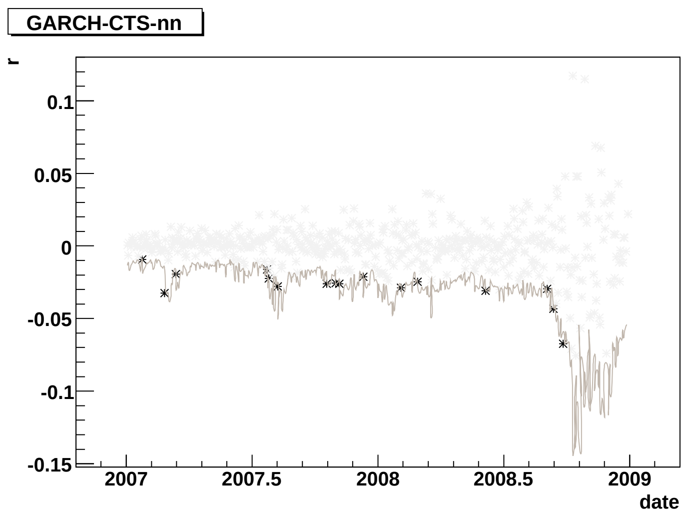

Figure 13: Value at risk for the GARCH-CTS-nn model.

This figure reports the estimation of value at risk for the return of the Dow Jones Industrial Average (2007-2009) using a neural network model with a volatility estimation of a normal-ARMA-GARCH model. The outputs of the neural network are fitted by a classical tempered stable distribution. The black dots symbolize the VaR violations.

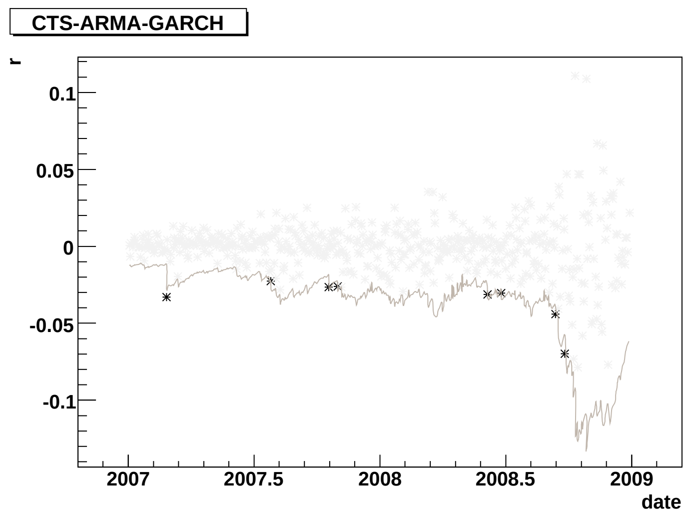

Figure 14: Value at risk for the CTS-ARMA-GARCH model.

This figure reports the estimation of value at risk for the return of the Dow Jones Industrial Average (2007-2009) using a CTS-ARMA-GARCH-model. The black dots symbolize the VaR violations.

	normal-ARMA-GARCH	*t*-ARMA-GARCH	EWMA-CTS-nn	GARCH-CTS-nn	CTS-ARMA-GARCH
d_0	0.0191	0.0300	0.0128	0.0125	**0.0101**
d_{32}	0.0042	0.0071	0.0068	0.0036	**0.0022**

Table 1: Comparison of the performance of all models.

This table reports the performance of forecasting the return of the Dow Jones Industrial Average (2007-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility with a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution. In d_0 the forecasting abilities in the central area is tested while d_{32} measures the forecasts of the tails. The numbers of the best model are bolded.

time span	normal-ARMA-GARCH	t-ARMA-GARCH	EWMA-CTS-nn	GARCH-CTS-nn	CTS-ARMA-GARCH
1986-1987	**0.028216**	0.034336	0.041713	0.040498	0.035234
1987-1988	**0.022644**	0.030760	0.040212	0.040457	0.028297
1988-1989	0.038154	0.048520	0.027009	**0.018101**	0.020179
1989-1990	0.038065	0.058315	0.061605	0.046072	**0.026360**
1990-1991	0.032053	0.036725	0.017759	**0.014751**	0.021959
1991-1992	0.045607	0.058237	0.017688	0.019752	**0.016277**
1992-1993	0.054623	0.066664	**0.016332**	0.025612	0.028545
1993-1994	0.059484	0.066465	**0.026723**	0.043880	0.032445
1994-1995	0.016809	0.032303	**0.009143**	0.015154	0.016853
1995-1996	0.043439	0.048629	0.052760	0.051693	**0.026296**
1996-1997	0.022529	0.026427	0.035784	0.043713	**0.013983**
1997-1998	0.018159	**0.013125**	0.025526	0.041042	0.022264
1998-1999	**0.008944**	0.014877	0.012507	0.027169	0.018269
1999-2000	0.016726	0.020272	0.018360	**0.016350**	0.026765
2000-2001	0.024323	0.026312	0.027842	**0.018941**	0.028146
2001-2002	0.022391	0.026269	0.017002	**0.012332**	0.026585
2002-2003	0.041965	0.040762	0.043573	**0.035881**	0.051660
2003-2004	0.025904	0.034855	0.030652	0.025739	**0.013507**
2004-2005	0.033400	0.042242	0.017000	**0.016907**	0.032626
2005-2006	0.018641	0.023739	0.014251	**0.012594**	0.029929
2006-2007	0.032855	0.038259	0.029995	**0.022319**	0.025798
2007-2008	0.033232	0.034138	0.031309	0.033099	**0.024182**
2008-2009	0.034417	0.034999	0.026763	**0.025185**	0.042336

Table 2: Comparison of d_0 for all models depending on the specific time interval.

This table reports the performance of forecasting the return of the Dow Jones Industrial Average (2007-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility with a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution. In d_0 the forecasting abilities in the central area is tested. The numbers of the best model for a specific time interval are bolded.

time span	normal-ARMA-GARCH	t-ARMA-GARCH	EWMA-CTS-nn	GARCH-CTS-nn	CTS-ARMA-GARCH
1986-1987	0.007395	**0.006087**	0.016646	0.015032	0.008058
1987-1988	0.013496	**0.003051**	0.015146	0.015595	0.009155
1988-1989	0.006286	0.007846	0.021596	0.011474	**0.003703**
1989-1990	0.004091	0.008857	0.026887	0.017027	**0.002509**
1990-1991	0.008461	0.006354	0.008743	**0.005080**	0.007492
1991-1992	0.003730	0.010248	0.008351	0.006005	**0.002315**
1992-1993	0.009839	0.006559	**0.003821**	0.005877	0.007939
1993-1994	0.008969	0.004607	**0.003085**	0.007423	0.007329
1994-1995	0.006938	0.005760	0.007138	0.007500	**0.005153**
1995-1996	0.005370	0.008457	**0.003842**	0.006845	0.005599
1996-1997	0.006429	**0.006213**	0.010610	0.007321	0.008976
1997-1998	0.011416	**0.005615**	0.013973	0.013154	0.006351
1998-1999	0.008785	**0.004317**	0.005139	0.007104	0.004926
1999-2000	0.006034	**0.002746**	0.005220	0.003298	0.004898
2000-2001	0.011860	0.006267	**0.002795**	0.005773	0.007767
2001-2002	0.004968	0.007266	0.003313	0.003556	**0.003199**
2002-2003	0.007920	**0.002561**	0.009977	0.008295	0.007073
2003-2004	0.005898	0.007637	**0.002653**	0.003421	0.002725
2004-2005	0.006470	0.006107	**0.002867**	0.003966	0.003668
2005-2006	0.004693	0.006181	**0.003081**	0.003736	0.003474
2006-2007	0.003923	0.006061	0.004354	0.004146	**0.003134**
2007-2008	0.012549	0.009672	0.010819	0.015471	**0.008874**
2008-2009	0.014378	**0.006122**	0.013072	0.013401	0.008955

Table 3: Comparison of d_{32} for all models depending on the specific time interval.

This table reports the performance of forecasting the return of the Dow Jones Industrial Average (2007-2009) using a normal-ARMA-GARCH, a t-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility with a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution. In d_{32} the forecasting abilities of the tails is tested. The numbers of the best model for a specific time interval are bolded.

	normal-ARMA-GARCH	*t*-ARMA-GARCH	EWMA-CTS-nn	GARCH-CTS-nn	CTS-ARMA-GARCH
2007	7	**4**	8	10	**4**
2008	9	**4**	**5**	**6**	**4**

Table 4: Comparison of the value at risk violations in the years 2007 and 2008.

This table reports the value at Risk violations forecasting the return of the Dow Jones Industrial Average (2007-2009) using a normal-ARMA-GARCH, a *t*-ARMA-GARCH, a CTS-ARMA-GARCH, and two neural network models. The EWMA-CTS-nn model uses an exponentially weighted moving average for the volatility while the GARCH-CTS-nn model estimates the volatility with a normal-ARMA-GARCH model. The outputs of the neural network models are fitted with a classical tempered stable distribution. The numbers of the models accepted by Kupiec's test are bolded.