Measuring Intra-Daily Market Risk: A Neural Network Approach

Wei Sun

School of Economics and Business Engineering

University of Karlsruhe, KIT, and GCFD, Germany

Email: wei.sun@kit.edu

Svetlozar Rachev

School of Economics and Business Engineering

University of Karlsruhe and KIT, Germany

Department of Statistics and Applied Probability

University of California, Santa Barbara, USA

&

Chief Scientist of FinAnalytica

Email: rachev@kit.edu.

Ye Chen

School of Economics and Business Engineering

University of Karlsruhe, KIT, and GCFD, Germany

Email: ye.chen@kit.edu.

Frank J. Fabozzi

Yale School of Management, USA

Email: frank.fabozzi@yale.edu

Abstract

The value at risk (VaR) measure often relies on an assumption about the return (or price) distribution of the underlying risky assets. Different distributional assumptions may produce widely different computed VaR values. When estimating VaR using intra-daily equity returns, the question arises as to what assumption should be made about the return distribution. Because of the difficulty of decomposing trading noise, it is very hard to identify the return distribution at the tick-by-tick level. In this paper, we circumvent the difficulty of making a distributional assumption of intra-daily market fluctuations by specifying a neural network approach. With this approach, no distributional assumption regarding the return distribution is required for estimating and forecasting the VaR using intra-daily data. Using this approach, we forecast VaR using high-frequency data for the German equity market index. Our neural network forecasts, evaluated on the basis of several statistical performance measures and compared with alternative time-series models, suggest that the performance of the neural network approach in VaR computation dominates that of the commonly used time-series models.

Keywords: High Frequency, Intra-daily Data, Neural Network Approach, Value at Risk

JEL Classification: C15, C46, C52, G15

1. Introduction

Value at risk (VaR) serves as a commonly used methodology for managing market risk. VaR is defined as the lowest quantile of the potential loss over a specified time period. The advantages of VaR as a measure of risk have been well documented in the literature (see, for example, Jorion (2007)), so to save space we will not review them here. The VaR methodology has been endorsed by regulators and financial industry advisory groups, as well as being adopted by financial institutions in their risk management operations and nonfinancial corporations for risk reporting. VaR estimates have been used in making decisions regarding capital resource allocation, setting position limits, and performance evaluation [1].

The standard VaR computation (e.g., delta-normal VaR) requires that the underlying return-generating processes for the risky assets be normally distributed, where the moments are time invariant and can be estimated with historical data. Despite the increased use of the VaR methodology, it does have well-known drawbacks. VaR is not a coherent risk measure and does not provide insight into the risk beyond the quantile.[2] The empirical work by Beder (1995) clearly demonstrates how different VaR models can lead to dramatically different VaR estimates. Moreover, when employing the VaR methodology, it is possible for a decision maker, unintentionally or not, to decrease portfolio VaR while simultaneously increasing the expected losses beyond the VaR (i.e., by increasing the "tail risk" of a portfolio or position).[3] There are superior measures to VaR for measuring market risk, such as average value at risk (AVaR). This risk measure – also called conditional value at risk (CVaR) and, for continuous distributions called expected tail loss (ETL) – is a coherent risk measure that overcomes the conceptual deficiencies of VaR.[4] Even with these well-known limitations, however, VaR remains the most popular measure of market risk employed by risk managers [5].

Different arguments about the distributional assumption of the underlying risky assets have been proposed in the literature. Neftci (2000) points out that extreme events are structurally different from the return-generating process under normal market conditions. Höchstötter et al. (2005) and Rachev et al. (2005b, 2007b) make the same argument, focusing on the stylized fact that returns are heavy tailed. Brooks et al. (2005) argue that heavy tailedness might lead to an underprediction of both the size of extreme market movements and the frequency with which they occur. Sun et al. (2008) propose a methodology for computing VaR based on the fractional stable model.

With the availability of intra-daily price data (i.e., high-frequency data), researchers and practitioners have focused more attention on market microstructure issues to understand and help formulate strategies for the timing of trades. Intra-daily data have many problematic stylized facts, they are often heavy tailed and their variance is heteroscedastic with a dependence pattern (see, for example, Sun et al (2007, 2008b)). The major characteristic of intra-daily data is that they exhibit erratic arrival that contains several distinct types of noise. Such noises reflect the trading behavior and information flows in the market. Due to the difficulty of decomposing the different types of noises, an iterated process such

---

[1]Sun et al. (2008b) provide a review of the adoption of VaR for measuring market risk. A more technical discussion of market risk can be found in Khindanova and Rachev (2000), Khindanova et al. (2001), and Gamrowski and Rachev (1996).

[2]See Artzner et al. (1999).

[3]See Martin et al. (2003) and the references therein.

[4]See, for example, Acerbi and Tasche (2002) and Rachev et al. (2005a).

[5]See Dowd (2002) for the characteristics of VaR that make it appealing to risk managers

as a neural network model might be more suitable for prediction purposes. In this paper, we propose an approach for calculating VaR with high-frequency data that utilizes the neural network model. The empirical evidence we present suggests that this approach outperforms the three most popular parametric models (autoregressive model, moving average model, and autoregressive moving average-GARCH model) used in practice to estimate VaR. Our findings are consistent with several empirical studies reported in the literature [6].

We have organized the paper as follows. In Section 2, we introduce the methodology of the neural network models employed in our study for estimating and evaluating VaR. In Section 3, we specify the parametric and nonparametric methods investigated in our study utilized to compute VaR. We describe the methods to assess the relative performance of the VaR estimates in Section 4. The study's data, empirical methodology, and results are covered in Section 5. Our empirical analysis is based on high-frequency data (1-minute level) for the German DAX index. We summarize our findings in Section 6.

2. Neural Networks in Modeling Market Fluctuation

A neural network (NN) is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases, a NN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. In contrast to the k-nearest neighbor method, neural network is a type of *eager learning*. In more practical terms, neural networks are nonlinear statistical data modeling tools used to characterize highly complex and convoluted relationships between inputs and outputs or to find correlation patterns in financial data. In finance, NN models have been used for various tasks. Kantardzic (2003) identify several asset management firms that employ NN technology for data mining.

Like the linear and polynomial approximation methods, a neural network relates a set of input variables x_i, $i = 1, \ldots, k$ to a set of one or more output variables y_j, $j = 1, \ldots, k$. Neural networks are essentially mathematical models defining a function $f : X \to Y$. Each type of NN model corresponds to a class of such functions. The difference between a NN model and other approximation methods is that NN take advantage of one or more hidden layers, in which the input variables are transformed by a special function known as a logistic or logsigmoid transformation; that is, the function $f(x)$ is a composition of other functions $g_i(x)$ that can further be defined as a composition of other functions. Functions $f(x)$ and $g_i(x)$ are composed of a set of elementary computational units called neurons, [7] which are connected through weighted connections. These units are organized in layers so that every neuron in a layer is exclusively connected to the neurons of the proceding layer and the subsequent layer. Every neuron represents an autonomous computational unit and receives inputs as a series of signals that dictate its activation. All the input signals reach the neuron simultaneously and the neurons can receive more than one input signal. Following the activation dictated from input singals, the neurons produce the output signals. Every input signal is associated with a connection weight which determines the relative importance of the input signals in generating the final impulse transmitted by the neuron.

---

[6]See, for example, Bolland and Connor (1997).

[7]Sometimes such elementary computational units are called nodes, neurodes, units, or processing elements (PEs).

Formally, the algorithm mentioned above can be expressed as follows:

$$n_{k,t} = w_{k,0} + \sum_{i=1}^{i^*} w_{k,i}\, x_{i,t} \tag{1}$$

$$N_{k,t} = G(n_{k,t}) \tag{2}$$

$$y_t = \gamma_0 + \sum_{k=1}^{k^*} \gamma_k\, N_{k,t} \tag{3}$$

where $G(\cdot)$ represents the activation function and $N_{k,t}$ stands for the neurons. In these system, there are i^* input variables x and k^* neurons. A linear combination of these input variables observed at time t, i.e., $x_{i,t}$, $i = 1, \ldots, i^*$, with the coefficient vector (i.e., a set of input weights) $w_{k,i}$, $i = 1, \ldots, i^*$, and a constant term $w_{k,0}$ form the variable $n_{k,t}$. This variable $n_{k,t}$ is transformed by the activation function $G(\cdot)$ to a neuron $N_{k,t}$ at time (or observation) t. The set of k^* neurons at time (or observation) index t are combined in a linear way with the coefficient vector γ_k, $k = 1, \ldots, k^*$ and taken with a constant term γ_0 to form the output value y_t at time t. In defining a NN model, the activiation function $G(\cdot)$ is typically one of the elements to specify. Giudici (2003) summarizes three commonly employed types of activiation functions: linear, stepwise, and signoidal.

The neurons of a NN model are organized in layers. There are three types of layers: input, output, and hidden. The input layer receives information only from the external information, i.e., an explanatory variable x_i. There is no any calculation performed for the input layer. It only transmits information to the next level. The output layer only produces the final results sent by the network to outside of the system, i.e., response variable y_j. Between the input and output layers there can be one or more intermediate layers, called hidden layers, so named because these layers are not directly connected with the external information. Giudici points out that the architecture of a NN model refers to the network's organization: (1) the number of layers, (2) the number of neurons belonging to each layer, (3) the manners in which the neurons are connected, and (4) the direction of flow for the computation.

Different information flows lead to different types of network. The NN model can by divided into to types based on the information flow: feedforward networks and recurrent networks. In the feedforward network, the information moves in only one direction: from the input layer through the hidden layer and to the output layer. There are no cycles or loops in this type of network. Equations (1)-(3) describe the feedforward networks. In contrast to feedforward networks, recurrent networks are models with bi-directional information flow which allow the neurons to depend not only on the input variable x_i but also on their own lagged values $n_{k,t-p}$ at order p. McNelis (2005) shows that the recurrent network builds "memory" in the evolution of the neurons. Replacing equation (1) with the following equation, the system of a recurrent network can be formed,

$$n_{k,t} = w_{k,0} + \sum_{i=1}^{i^*} w_{k,i}\, x_{i,t} + \sum_{k=1}^{k^*} \phi_k\, n_{k,t-p} \tag{4}$$

McNelis notes that the recurrent network has an indirect feedback effect from the lagged unsquashed neurons to the current neurons, not a direct feedback from lagged neurons to the level of output.

A NN model modifies its interconnection weights by applying a set of learning (training) samples. The learning process leads to parameters of a network which represent implicitly stored knowledge from

the data. More generally, given a specific task to solve and a class of functions F, learning means using a set of observations (learning/training samples) in order to find $f^* \in F$ which solves the task in an optimal sense. This entails defining a *cost function* $C : F \to \Re$ such that, for the optimal solution f^*, $C(f^*) \leq C(f) \; \forall f \in F$; that is, no solution has a cost less than that of the optimal solution. Since NN models learn from data, the cost function must be a function of the observations (see, for example, Kantardzic 2003).

In this paper, we focus on the multilayer perceptron model, which is a feedforward network with several hidden layers, and one input layer totally interconnected with one output layer. The reason we choose this model is that multilayer perceptrons have optimal properties. Specifically, when a sufficiently large number of nodes in the hidden layer is given, a simple neural network structure is capable of serving as an approximation of any functional form with arbitrary accuracy. This is known as the principle of universal approximation; that is, the rate of convergence is independent of the problem's dimension. Gençay (1994) reports that the results from his bifurcation analysis indicate that feedforward networks can successfully approximate the qualitative changes in the dynamcis of the time series data under investigation. His results confirm that feedforward networks can filter a realistic amount of noise quite satisfactorily.

3. Estimation of VaR

Given $\alpha \in (0, 1]$, R is random loss of an investment in certain period, VaR of a random variable R [8] at level of α is the absolute value of the worst loss not to be exceeded with a probability of at least α. More formally, if α-quantile of the loss $L = -R$ is $q_\alpha(L) = \inf\{r \in \Re : P[L \leq r] \geq \alpha\}$, the VaR at confidence level α of R is $VaR_\alpha(R) = q_\alpha(L)$. Arztner et al (1999) proposed the concept of a coherent risk measure. A risk measure is called coherent if it is monotonous, positively homogeneous, translation invariant, and subadditive. Rockafellar and Uryasev (2002) demonstrate that CVaR or AVaR is coherent and can be expressed as $CVaR_\alpha(L) = E(L|L > VaR_\alpha(R))$.[9] That is, for a loss L with $E(|L|) < \infty$ and distribution function F_L, the conditional value at risk at confidence level $\alpha \in (0, 1)$ is defined as

$$CVaR_\alpha(-R) = AVaR_\alpha(L) = \frac{1}{1-\alpha} \int_\alpha^1 q_u(F_{-R}) du = \frac{1}{1-\alpha} \int_\alpha^1 VaR_u(F_{-R}) du \qquad (5)$$

where $q_u(F_{-R})$ is the quantile function of F_L.

3.1 Non-parametric approach of VaR estimation

Because VaR is the quantile of loss distribution of a risky asset, estimation of VaR requires the estimation of the loss distribution. The kernel estimator is the basic methodology used to estimate density (see Silverman (1986)). If a random variable X has density $f(x)$, then

$$f(x) = \lim_{a \to 0} \frac{1}{2a} P(x - a < X < x + a) \qquad (6)$$

---

[8] R represents the return, while $L = -R$ is the loss.

[9] See also Rachev et al. (2007) for additional references.

By counting the proportion of sampling observations falling in the interval of $(x-a, x+a)$, the probability $P(x - a < X < x + a)$ can be estimated for any given a. Defining kernel function K for

$$\int_{-\infty}^{\infty} K(x)d(x) = 1 \tag{7}$$

in which, $K(x)$ is usually regarded as a symmetric probability density function, for example, the normal desity. The kernel estimator is defined by

$$\tilde{f}(x) = \frac{1}{na} \sum_{i=1}^{n} K\left(\frac{x - X_i}{a}\right) \tag{8}$$

where a is the window width and n is the sample size. The kernel estimator can be looked as a sum of bumps placed at the observations X_i. Kernel function $K(x)$ determines the shape of the bumps and the window width a determines the width of bumps.

For evaluating the quality of the estimate, the mean-square error (MSE) is defined as:

$$MSE_x(\tilde{f}) = E\left(\tilde{f}(x) - f(x)\right)^2 = \left(E\tilde{f}(x) - f(x)\right)^2 + var\left(\tilde{f}(x)\right) \tag{9}$$

The global closeness of the fit of $\tilde{f}(x)$ to $f(x)$ is found by intergrating the MSE over x; the mean integrated square error (MISE) is defined as

$$MISE_x(\tilde{f}) = E\int \left(\tilde{f}(x) - f(x)\right)^2 dx = \int \left(E\tilde{f}(x) - f(x)\right)^2 dx + \int var\left(\tilde{f}(x)\right) dx \tag{10}$$

Given a symmetric kernel function K, $\int tK(t)dt = 0$ and $\int t^2 K(t)dt = k_2 \neq 0$, Silverman (1986) shows that the approximation of MISE is:

$$\frac{1}{4}a^4 k_2^2 \int f''(x)^2 dx + \frac{1}{na} \int K(t)^2 dt$$

It is clear the bias in the estimation of $f(x)$ depends on the window width. The optimal window width a_{opt} can be chosen by minimizing the MISE. Silverman (1986) shows that

$$a_{opt} = n^{-1/5} k_2^{-2/5} \left(\int f''(x)^2 dx\right)^{-1/5} \left(\int K(t)^2 dt\right)^{1/5} \tag{11}$$

and also shows that the optimal solution is given by the Epanechnikov kernel $K_E(x)$:

$$K_E(x) = \begin{cases} \frac{3}{4\sqrt{5}}(1 - \frac{x^2}{5}), & -\sqrt{5} \leq x \leq \sqrt{5} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

A drawback of the kernel estimator is its inefficiency in dealing with long-tailed distributions. Since across the whole sample, window width is fixed, a good degree of smoothing over the cernter of the distribution will often leave spurious noise in the tails (see Silverman 1986 and Dowd 2005). Silverman (1986) offers some solutions to this drawback such as nearest neighbor method and vairable kernel method. For the former method, the window width placed on an observation depends on the distance between that observation and its nearest neighbors. For the kernel estimator, the density $f(x)$ is estimated using:

$$\tilde{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{ah_{i,k}} K\left(\frac{x - X_i}{ah_{i,k}}\right) \tag{13}$$

where $h_{i,k}$ is the distance between X_i and the kth nearest of the other data points. The window width of the kernel placed on X_i is proportional to $h_{i,k}$. Therefore, flatter kernels will be placed on more sparse data.

3.2 Parametric approach of VaR estimation

Parametric approach of VaR estimation is based on the assumption that financial returns R_t are a function of two components μ_t and ε_t. That is, $R_t = f(\mu_t, \varepsilon_t)$. R_t can be regarded as a function of ε_t conditional on a given μ_t; typically, this function takes a simple linear form $R_t = \mu_t + \varepsilon_t = \mu_t + \sigma_t u_t$. Usually μ_t is called the location component and σ_t the scale component. u_t is a i.i.d random variable that follows a probability density function f_u. VaR based on information up to time t is

$$\tilde{VaR}_t := q_\alpha(R_t) = -\tilde{\mu}_t - \tilde{\sigma} \, q_\alpha(u) \tag{14}$$

where $q_\alpha(u)$ is the α-quantile implied by f_u.

Unconditional parametric approaches set μ_t and σ_t as constants, and the returns R_t are i.i.d random variables with density $\sigma^{-1} f_u(\sigma^{-1}(R_t - \mu))$. Conditional parametric approaches set the location component and the scale component as functions not constants. Typical time-varying conditional location setting is the $\mathrm{ARMA}(r, m)$ processes. That is, for the conditional mean equation is

$$\mu_t = \alpha_0 + \sum_{i=1}^{r} \alpha_i \, R_{t-i} + \sum_{j=1}^{m} \beta_j \varepsilon_{t-j}. \tag{15}$$

The typical time-varying conditional variance setting is a $\mathrm{GARCH}(p,q)$ process; that is,

$$\sigma_t^2 = \kappa + \sum_{i=1}^{p} \gamma_i \, \sigma_{t-i}^2 + \sum_{j=1}^{q} \theta_j \, \varepsilon_{t-j}^2 \tag{16}$$

$\varepsilon_t = \sigma_t u_t$ where u_t are $N(0, 1)$.

4. Evaluation of VaR estimators

Backtesting is the usual method to evaluate the VaR estimators and its forecasting quality. It can be performed for in-sample estimation evaluation and for out-of-sample interval forecasting evaluation. The backtesting is based on the indicator function I_t which is defined as $I_t(\alpha) = I(r_t < -VaR_t(\alpha))$. The indicator function shows violations of the quantiles of the loss distribution. The process $\{I_t\}_{t \in T}$ is a process of i.i.d Bernoulli variables with violation probability $1 - \alpha$. Christoffersen (1998) shows that evaluating the accuracy of VaR can be reduced to checking whether the number of violations is correct on average and the pattern of violations is consistent with i.i.d processes. In other words, an accurate VaR measure should satisfy both the unconditional coverage property and the independent property. The unconditional coverage property means that the probability of realization of a loss in excess of the estimated $VaR_t(\alpha)$ must be exactly $\alpha\%$, that is, $P(I_t(\alpha) = 1) = \alpha$. The independent property means that previous VaR violations do not presage future VaR violations.

Kupiec (1995) proposes a frequency of failures test that checks how many times an estimated VaR is violated in a given time period. If the observed frequency of failures of the estimated VaR differs significantly from $\alpha \times 100\%$, the underlying risk measure is less reliable. Using a sample of T observations, Kupiec's test statistic K is derived from

$$K = 2[\ell(\tilde{\alpha}; I_1, I_2, ..., I_T) - \ell(\alpha; I_1, I_2, ..., I_T)] \sim \chi_1^2 \tag{17}$$

where $\ell(\cdot)$ is log binomial likelihood, the observed frequency of violations is $\tilde{\alpha} = I(\alpha)/T$, and $I(\alpha) = \sum_{t=1}^{T} I_t(\alpha)$ is the number of violations in the sample. Then equation (17) can be transferred as:

$$
\begin{aligned}
K &= 2\log\left((1-\tilde{\alpha})^{T-I(\alpha)}\,\tilde{\alpha}^{I(\alpha)}\right) - 2\log\left((1-\alpha)^{T-I(\alpha)}\,\alpha^{I(\alpha)}\right) \\
&= 2\log\left(\left(\frac{1-\tilde{\alpha}}{1-\alpha}\right)^{T-I(\alpha)}\left(\frac{\tilde{\alpha}}{\alpha}\right)^{I(\alpha)}\right)
\end{aligned}
\tag{18}
$$

Note that if the frequency of violations from estimated VaR $\tilde{\alpha} \times 100\%$ is exactly equal the preset $\alpha \times 100\%$, the test statistic is zero; otherwise, the test statistic shows the over-estimation or under-estimation of VaR. Kupiec (1995) also proposes the Wald variant z of the likelihood ratio statistic, which is

$$
z = \frac{\sqrt{T}(\tilde{\alpha} - \alpha)}{\sqrt{\alpha(1-\alpha)}} \sim N(0,1)
\tag{19}
$$

If there is no violation, the K statistic cannot cover such case since $\log 0$ is undefined.

The shortcoming of the backtesting proposed by Kupiec (1995) is that it does not focus on the independence property. In order to detect violations of the independence property of an estimated VaR measure, say, $I_t(\alpha)$, Christoffersen (1998) suggests the Markov test. The Markov test detects whether the likelihood of a VaR violation depends on another VaR violation occurred on the previous time period. The assumption behind the Markov test is that if the VaR estimate accurately reflects the underlying risk situation, then (1) the chance of violating the current period's VaR should not depend on whether or not the previous period's VaR was violated and (2) the chance of violating the current period's VaR should not influence next period's violation.

Christoffersen (1998) defines I_t as a binary first-order Markov chain with transition probability matrix Θ, that is:

$$
\Theta = \begin{bmatrix} 1-\theta_{01} & \theta_{01} \\ 1-\theta_{11} & \theta_{11} \end{bmatrix}
$$

and $\theta_{ij} = P(I_t = j | I_{t-1} = i)$. Defining n_{ij} as the number of transitions from state i to state j, $n_{ij} = \sum_{t=2}^{T} I(I_t = i | I_{t-1} = j)$, the Christoffersen test statistic C is

$$
C = 2[\ell(\tilde{\Theta}; I_2, I_3, ..., I_T | I_1) - \ell(\tilde{\theta}_2; I_2, I_3, ..., I_T | I_1)] \sim \chi_1^2
\tag{20}
$$

where ℓ is the log binomial likelihood function and under the null hypothesis of testing independence, $\theta_{01} = \theta_{11} = \theta_2$. For

$$
\tilde{\theta}_{01} = \frac{n_{01}}{n_{00} + n_{11}}; \ \tilde{\theta}_{11} = \frac{n_{01}}{n_{10} + n_{11}}; \ \tilde{\theta}_2 = \frac{n_{01} + n_{11}}{n_{00} + n_{11} + n_{10} + n_{01}},
$$

$$
C = 2\log\left((1-\tilde{\theta}_{01})^{n_{00}}\,\tilde{\theta}_{01}^{\,n_{01}}(1-\tilde{\theta}_{11})^{n_{10}}\,\tilde{\theta}_{11}^{\,n_{11}}\right) - 2\log\left((1-\tilde{\theta}_2)^{(n_{00}+n_{11})}\,\tilde{\theta}_2^{\,(n_{00}+n_{11})}\right)
\tag{21}
$$

Christoffersen (1998) also suggests combining test statistic K and C in order to check unconditional coverage and independence together, that is

$$
LR = 2[\ell(\tilde{\Theta}; I_2, I_3, ..., I_T | I_1) - \ell(\alpha; I_2, I_3, ..., I_T | I_1)] \sim \chi_2^2
\tag{22}
$$

5. Empirical analysis

In this section, we describe the data and methodology employed in our study and present the empirical results.

5.1 The data

We study a time series sampled at one-minute frequency level of the German DAX index in 2003.[10] Since the data are not equally spaced, we aggregate the data to the equally spaced time series. The aggregation algorithm is based on the linear interpolation [11] introduced by Wasserfallen and Zimmermann (1995). That is, given a non-equally spaced series with times t_i and values $\varphi_i = \varphi(t_i)$, the index i identifies the irregularly spaced sequence. The target homogeneous time series is given at times $t_0 + j\Delta t$ with fixed time interval Δt starting at t_0. The index j identifies the regularly spaced sequence. The time $t_0 + j\Delta t$ is bounded by two times t_i of the irregularly spaced series, $I = \max(i \,|\, t_i \leq t_0 + j\Delta t)$ and $t_I \leq t_0 + j\Delta t > t_{I+1}$. Data are interpolated between t_I and t_{I+1}. The linear interpolation shows that

$$\varphi(t_0 + j\Delta t) = \varphi_I + \frac{t_0 + j\Delta t - t_I}{t_{I+1} - t_I} (\varphi_{I+1} - \varphi_I).$$

In order to transform the data to return series, we use the continuous compounding transformation as follows:

$$y_t = \log \frac{X_t}{X_{t-1}} = \log X_t - \log X_{t-1}$$

where y_t is the return series and X_t is the normal time series of the DAX index in price level. Figure 1 illustrates the time series of the DAX index level for a random portion of 200 intra-daily observations for 2003 while Figure 2 shows the corresponding return series. Later we will see that the pattern shown in Figure 2 is forecasted better by using the NN model than the three time series models.

5.2 The Methodology

We apply three time series model and the neural network model to compute one-step ahead VaR value. Three time series models applied in this section are the autoregressive model which we denote by AR(1), the moving average model that we denote by MA(1), and the ARMA(1,1)-GARCH(1,1) model. The NN model used is the feedforward network model with 10 hidden layers.

The AR(1) model is

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

and the MA(1) model is

$$y_t = \alpha_0 + \beta_1 \varepsilon_{t-1},$$

where α_0 is a constant and ε_t is assumed to be a white noise series.

---

[10]The DAX index is a stock market index whose components include 30 blue chip German stocks that are traded on the Frankfurt Stock Exchange.

[11]Dacorogna et al. (2001) point out that linear interpolation relies on the forward point of time and Müller et al. (1990) suggests that linear interpolation is an appropriate method for stochastic processes with i.i.d. increments.

The ARMA(1,1)-GARCH(1,1) model is:

$$y_t = a_0 + a_1 y_{t-1} + \varepsilon_t + b_1 \varepsilon_{t-1}.$$

$$\sigma_t^2 = b_0 + \gamma_1 \sigma_{t-1}^2 + \theta_1 \varepsilon_{t-1}^2.$$

where a_0 and b_0 are constants.

The feedforward network model is

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-10}, \theta) + \varepsilon_t$$

where ε_t is distributed with zero mean and one variance. Explicitly, the function $f(\cdot)$ is:

$$
\begin{aligned}
y_t &= f(y_t, \theta) + \varepsilon_t \\
&= F\left(\alpha_0 + \sum_{j=1}^{10} \alpha_i \, G(\theta_{i0} + \sum_{j=1}^{10} \theta_{ij} y_{j,t})\right) + \varepsilon_t
\end{aligned}
$$

where $F(\cdot)$ and $G(\cdot)$ are known activation functions. The back-propagation algorithm is applied for the model building (see, Gençay and Liu (1997)).

Cross-validation has been used for model assessment in our study. We apply the method introduced by Sun et al. (2007). We split the data into two parts, the training set and the forecasting set. We define N as the length of the sample. The sub-sample series used for the in-sample (training) analysis is randomly selected by a moving window of length T ($1 \leq T \leq N$). Replacement is allowed in the sampling. Letting T_F denote the length of the forecasting set, we perform one-step ahead out-of-sample forecasting ($1 \leq T \leq T + T_F \leq N$). In the empirical analysis, sub-sample length (i.e., the window length) of T was chosen for the training period and T_F for the out-of-sample forecasting (see Sun et al. (2007)). The training set contains 2,000 observations ($T = 2,000$) and the forecasting set contains 200 observations ($T_F = 200$), and we gathered totally 230 sub-samples. The comparison is done for the four models only based on the forecasting set. The reason we only focus on the forecasting set is that in practice, what banks do in risk management is to control future loss based on historical observations. Therefore, only the forecasting power of the four models used for the VaR computation is of interest.

We first use the training set (2,000 observations) to establish the models (i.e., to estimate the parameters). After the model has been estimated, we use it to predict (in this case, 200 observations). We compute the VaR values based on the predicted value.

We use the following goodness of fit tests suggested in the model assessment literature: the Anderson-Daring (AD) distance, the Cramer Von Mises (CVM) distance, the Kolmogorov-Simirnov (KS) distance, the Kuiper distance (K), and the MSE. Specifically, these criterion are defined as follows:

$$AD = \sup_{x \in \Re} \frac{|M_s(x) - \tilde{M}(x)|}{\sqrt{\tilde{M}(x)(1 - \tilde{M}(x))}},$$

$$CVM = \int_{-\infty}^{\infty} \left(M_s(x) - \tilde{M}(x)\right)^2 d\tilde{M}(x),$$

$$KS = \sup_{x \in \Re} |M_s(x) - \tilde{M}(x)|,$$

and

$$K = \sup_{x \in \Re} \Big(M_s(x) - \tilde{M}(x) \Big) + \sup_{x \in \Re} \Big(\tilde{M}(x) - M_s(x) \Big),$$

where $F_s(x)$ denotes the empirical sample distribution and $\tilde{F}(x)$ is the estimated distribution function of the computed VaR values.

5.3 The Results

The mean computed 95% VaR values for the four models as well as the actual VaR computed from the actual observations (i.e., the forecasting set) by using a nonparametric method are summarized below:

	Actual	AR model	GARCH model	MA model	NN model
$VaR_{95\%}$	3.3662×10^{-4}	1.6300	1.7031	2.1449	3.8605×10^{-4}

As can be seen, only the VaR value computed using the NN model is close to the observed (actual) value. As we mentioned in Section 4, the VaR value depends on the underlying data-generating process. In other words, the more the predicted data generated by the underlying model is close to the actual forecasting set, the more the VaR value computed by the underlying model is close to the actual VaR value.

In order to compare the VaR values computed from the predicted data set and that from the forecasting set, we use the tests described above. The results, reported in Table 1, indicate that the NN model outperforms the three time series models based on the smaller values in all the computed test values for the NN model. The VaR values computed by the NN model are close to the actual VaR value from the actual observations under different statistical criterion.

The four panels in Figure 3 show one example of 200 forecasted values for the four models. A comparison of Figure 2 with the four panels in Figure 3, clearly suggests that the NN model looks closer in shape to the actual observations.

Table 2 shows the results for the two tests to evaluate VaR. The Kupiec test shows the violation ratio. The NN model is the closest one to a 5% confidence level. The Christoffensen test rejects the null hypothesis of dependent violation for the four models investigated.

6. Conclusion

There is considerable interest in computing VaR for market risk management. In order to estimate market risk at the intra-daily level, we propose a neural network approach for computing high-frequency VaR. We investigate the one-step ahead risk forecast for the German DAX index using three time series models (i.e., the AR model, the MA model, and the ARMA-GARCH model) commonly utilized in risk management and a neural network model. The statistical tests we employ clearly find that the neural network model outperforms the three time series models.

Why do the underlying three time series models dramatically underperform the NN model? The reason is that time series models have several drawbacks. First, standard time series models require the residuals follow a normal distribution. As we noted earlier, the preponderance of the empirical evidence from real-world financial markets clearly rejects the hypothesis that returns are normally distributed. Second, time series models do not have a memory while the NN model can memorize the dynamics of the observation and save it with hidden layers. The normal distribution for a random variable cannot capture the memory effect contained in the observations. Because neither the model structure nor the error term can capture the memory effect, the accuracy of the model is reduced. Third, the NN model allows us to use very accurate computing methods, i.e., a recursive method referred to as backpropogation, while the maximum likelihood estimation for the time series models is less accurate in its applicable algorithm.

Acknowledgment

References

[1] Acerbi, C. and Tasche, D., 'On the coherence of expected shortfall', *Journal of Banking and Finance*, Vol. 26, 2002, pp. 1487-1503.

[2] Artzner, P., Delbaen, F., Eber, J. and Heath, D., 'Coherent measures of risk', *Mathematical Finance*, Vol. 9, 1999, pp. 203-228.

[3] Beder, T., 'VAR: seductive but dangerous', *Financial Analysts Journal*, Vol. 51, 1995, pp. 12-24.

[4] Beltratti, A. and Morana, C., 'Computing value at risk with high frequency data', *Journal of Empirical Finance*, Vol. 6(5), 1999, pp. 431-455.

[5] Berkelaar, A., Cumperayot, P. and Kouwenberg, R., 'The effect of VaR based risk management on asset prices and the volatility smile', *European Financial Management*, Vol. 8(2), 2002, pp. 139-164.

[6] Berkowitz, J. and O'Brien, J., 'How accurate are value-at-risk models at commercial banks?', *Journal of Finance*, Vol. 57(3), 2002, pp. 1093-1911.

[7] Bolland, P.J., and Connor, J., 'A constrained neural network Kalman filter for price estimation in high frequency financial data', *International Journal of Neural Systems*, Vol. 8, No. 4, 1997, pp.399-415

[8] Brooks, C., Clare, A. D., Molle, J. and Persand, G., 'A comparison of extreme value theory approaches for determining value at risk', *Journal of Empirical Finance*, Vol. 12, 2005, pp. 339-352.

[9] Campbell, S.D., 'A review of backtesting and backtesting procedures', *Finance and Economic Discussion Series*, 2005, Federal Reserve Board, Washington, D.C.

[10] Christoffersen, P., 'Evaluating interval forecasts', *International Economic Review*, Vol. 39, 1998, pp. 841-862.

[11] Dowd, K., *Measuring market risk* (Chichester: John Wiley, & Sons, 2002).

[12] Gamrowski, B. and Rachev, S., 'Testing the validity of value at risk measures', In C. Heyde et al. (eds) *Applied Probability*, 1996.

[13] Gatti, S., Rigamonti, A., Saita, F. and Senati, M, 'Measuring value-at-risk in project finance transactions', *European Financial Management*, Vol. 13(1), 2007, pp. 135-158.

[14] Gençay, R., 'Nonlinear prediction of noisy time series with feedforward networks', *Physics Letters A*, Vol. 187, 1994, pp. 387-403.

[15] Gençay, R., Liu, T., 'Nonlinear modelling and prediction with feedforward and recurrent networks', *Physica D*, Vol. 108, 1997, pp. 119-134.

[16] Hallerbach, W.G. and Menkveld, A.J., 'Analysing perceived downside risk: the component value-at-risk framework', *European Financial Management*, Vol. 10(4), 2004, pp. 567-591.

[17] Höchstötter, M., Rachev, S., Fabozzi, F., 'Distributional analysis of the stocks comprising the DAX 30', *Probability and Mathematical Statistics*, Vol. 25(2), 2005, pp. 363-383.

[18] Khindanova, I. and Rachev, S., 'Value at Risk: Recent advances', In: G. Anastassiou (ed) *Handbook of Analytic-Computational Methods in Applied Mathematics*, pp. 801-858. (Boca Raton: CRC Press, 2000).

[19] Khindanova, I., Rachev, S. and Schwartz, E., 'Stable modeling of value at risk', *Mathematical and Computer Modelling*, Vol. 34, 2001, pp. 1223-1259.

[20] Kuester, K., Mittnik, S., and Paolella, 'Value-at-Risk prediction:a comparison fo alternative strategies', *Journal of Financial Econometrics*, Vol. 4(1), 2006, pp. 53-89.

[21] Kupiec, P., 'Techniques for verifying the accuracy of risk management models', *Journal of Derivatives*, Vol. 3, 1995, pp. 73-84.

[22] Martin, D., Rachev, S. and Siboulet F., 'Phi-Alpha optimal protfolios and extreme risk management', *Wilmott Magazine of Finance*, November, 2003, pp. 70-83.

[23] Mittnik, S. and Rachev, S., 'Modeling asset returns with alternative stable models', *Econometric Reviews*, Vol. 12, 1993a, pp. 261-330.

[24] Mittnik, S. and Paolella, M., 'Conditional density and value-at-risk prediction of Asian currency exchange rates', *Journal of Forecasting*, Vol. 19(4), 2000, pp. 313-333.

[25] Neftci, S. N., 'Value at risk calculation, extreme events, and tail estimation', *The Journal of Derivatives*, Vol. 7, 2000, pp. 23-37.

[26] Rachev, S. and Mittnik, S., *Stable Paretian Models in Finance* (New York: Wiley, 2000).

[27] Rachev, S., C. Menn and F. Fabozzi, *Fat-Tailed and Skewed Asset Return Distributions* (New Jersey: Wiley, 2005a).

[28] Rachev, S., Stoyanov, S., Biglova, S. and Fabozzi, F. 'An empirical examination of daily stock return distributions for U.S. stocks', In D. Baier, R. Decker, and L. Schmidt-Thieme (eds.), *Data Analysis and Decision Support, Springer Series in Studies in Classification, Data Analysis, and Knowledge Organization*, Berlin: Springer-Verlag, 2005b, pp. 269-281.

[29] Rachev, S., Mittnik, S., Fabozzi, F., Focardi, S. and Jasic, T., *Financial econometrics: from basics to advanced modelling techniques* (Hoboken New Jersey: Wiley, 2007a).

[30] Rachev, S.,Stoyanov, S. and Fabozzi, F., (2007), *Advanced Stochastic Models, Risk Assessment, and Portfolio Optimization: The Ideal Risk, Uncertainty, and Performance Measures*, John Wiley & Sons. Inc: Hoboken.

[31] Sun, W., Rachev, S. and Fabozzi, F., 'Fratals or i.i.d.: evidence of long-range dependence and heavy tailedness form modeling German equity market volatility', *Journal of Economics and Business*, vol. 59, 2007, pp. 575-595.

[32] Sun, W., Rachev, S. and Fabozzi, F., 'Long-Range Dependence, Fractal Processes, and Intra-daily Data', In: Seese, D., Weinhardt, C., Schlottmann, F., (eds.) *Handbook on Information Technology in Finance* (Berlin: Springer, 2008).

[33] Sun, W., Rachev, S., and Fabozzi, F., 'A New Approach of Using Lévy Processes for Determining High-Frequency Value-at-Risk Predictions, *European Financial Management*, 2008b, forthcoming.

Table 1: Model assessment with statistical goodness-of-fit tests.

Test	Model AR	GARCH	MA	NN
KS	0.5276	0.5276	0.5377	0.0704
AD	7.3909	7.3910	7.5331	0.2148
CVM	8.3543	8.3732	8.4394	0.0089
Kuiper	0.9899	1.0000	1.0000	0.1156
MSE	0.7395	0.8656	0.9891	2.5066×10^{-4}

Table 2: Model assessment in respect of VaR computation.

| Test	Model			
	AR	GARCH	MA	NN
Kupiec Test	0.7660	0.6900	0.8810	0.5750
Christoffersen Test	1.2586	0.2847	0.3104	0.1412
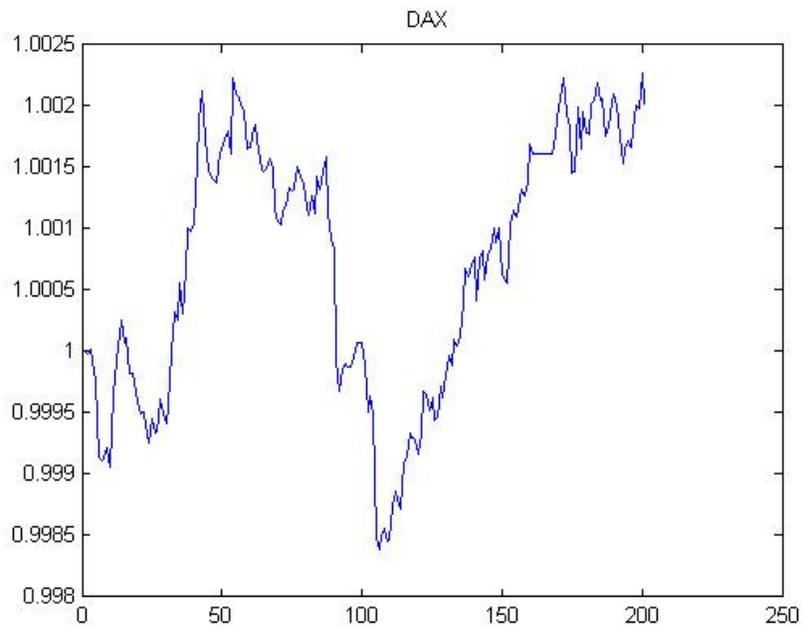
Figure 1: DAX index in level
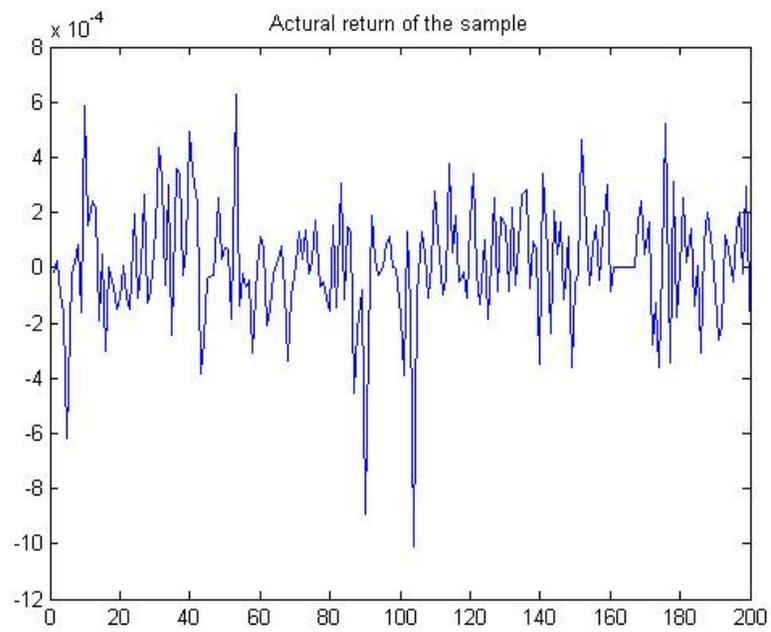
Figure 2: DAX index in return
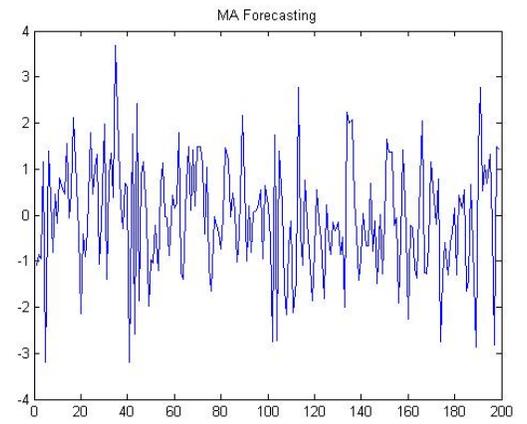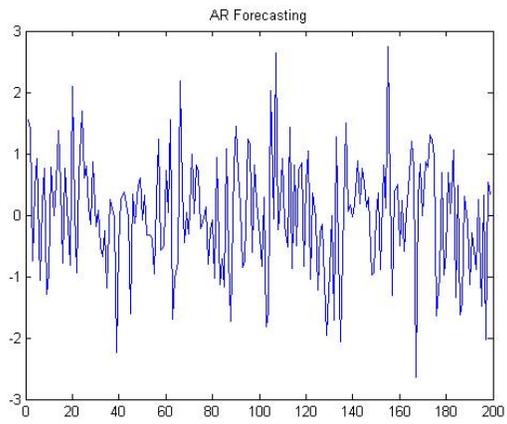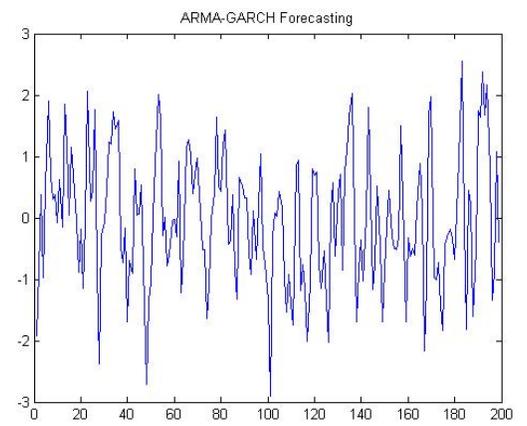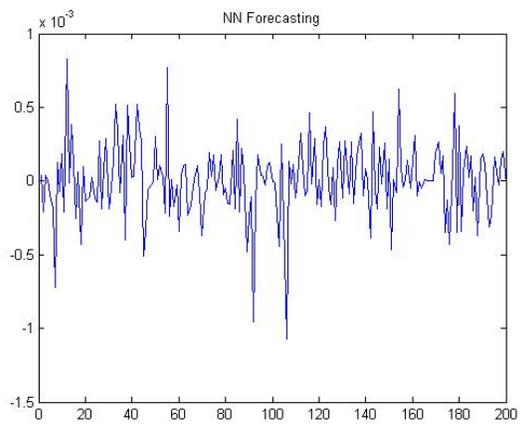
Figure 3: Model forecasting 200 returns using three time series models and the NN model.