Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2010 Proceedings

International Conference on Information Systems (ICIS)

2010

ROBUST DECISION SUPPORT SYSTEMS WITH MATRIX FORECASTS AND SHARED LAYER PERCEPTRONS FOR FINANCE AND OTHER APPLICATIONS

Hans-Jörg von Mettenheim
Leibniz Universität Hannover, mettenheim@iwi.uni-hannover.de

Michael H. Breitner
Leibniz Universität Hannover, breitner@iwi.uni-hannover.de

ROBUST DECISION SUPPORT SYSTEMS WITH MATRIX FORECASTS AND SHARED LAYER PERCEPTRONS FOR FINANCE AND OTHER APPLICATIONS

Completed Research Paper

Hans-Jörg von Mettenheim
Institut für Wirtschaftsinformatik
Leibniz Universität Hannover
Germany
mettenheim@iwi.uni-hannover.de

Michael H. Breitner
Institut für Wirtschaftsinformatik
Leibniz Universität Hannover
Germany
breitner@iwi.uni-hannover.de

Abstract

The recent financial crisis showed the need for more robust decision support systems. In this paper, we introduce a novel type of recurrent artificial neural network, the shared layer perceptron, which allows forecasts that are robust by design. This is achieved by not over-fitting to a specific variable. An entire market is forecast. By training not one, but many networks, we obtain a distribution of outcomes. Further, multi-step forecasts are possible. Our system uses hidden states to model internal dynamics. This allows the network to build a memory and hardens it against external shocks. Using a single shared weight matrix offers the possibility of interpreting system output. An often cited disadvantage of neural networks, the black box character, is not an issue with our approach. We focus on two case studies: determining value at risk and transaction decision support. We also present other applications, including load forecast in electricity networks.

Keywords: Decision Support Systems (DSS), Artificial Neural Networks, Financial Forecast

Introduction and Motivation

The recent financial crisis is still showing its effects, although the economy already has started to recover. Often, "faulty" models are said to have been the cause of the financial havoc wreaked (Khandani and Lo 2007), contaminating the real economy. Generally, models are implemented using information systems, especially decision-support systems. Humans tend to trust these systems blindly (Bookstaber 2007, p. 143 ff.) and also tend to ignore situations in which the models operate outside their normal regime. The question arises: "Is it possible to build a modeling framework that adequately reflects the dynamics of a system but is still robust and simple enough to be understood by those who operate it?" In building our decision support system, we follow the methodology described by Hevner and Vanstone (Hevner et al. 2004, Hevner 2007, Vanstone and Finnie 2009, Yu et al. 2007b).

Such a model cannot be followed blindly. However, a decision support system that incorporates a model that is robust by design leads to better decisions. In this paper, we present such a model – one that is able to robustly forecast several observables of a dynamic system. Multi-step forecasts are possible and the forecast is not necessarily a single number, which will never be exact anyway. Rather, the forecast is a distribution of outcomes (Zimmermann et al. 2006, Mettenheim 2009). The basis of the model is rooted in artificial neural networks (Haykin 2009). However, this is merely a framework. The model itself can be generalized.

Our model primarily addresses perceived weaknesses of models that are in use today at financial institutions and by (financial) decision-makers (Bookstaber 2007, p. 143 ff., Khandani and Lo 2007, Laidi 2009, p. 127 ff., Turban et al. 2010). Specifically, we consider the above mentioned following points:

- Lack of robustness: Most models are built to explain a single variable. However, when used to explain another, related variable they often fail. Such a model does not seem trustworthy. Our approach automatically models all available observables.
- Monocausality: In an attempt to simplify a model, only a few variables are taken into account. While such models are easy to present and market, they may oversimplify a complex relationship. Our approach aims at modeling entire markets without increasing complexity.
- Reliance on a single number: Models that only produce a single number are easy to interpret. However, this single number may (and invariably will) be precisely wrong. This is partly mitigated by computing confidence intervals. For this it is first necessary to assume an underlying distribution. Here, another source of error looms. Our model produces a distribution of outcomes with minimal prior assumptions.
- Blind flying: Most models only produce a single-step forecast. We argue that in order to make clear decisions, multi-step forecasts are desirable.

The remainder of this paper is structured as follows. A literature review briefly introduces literature on decision support systems and forecasting, especially in the context of artificial neural networks and finance. This suggests that a nonlinear modeling technique is a sensible approach to real-world social dynamic systems. The model using a shared layer perceptron is presented.

Three case studies follow, two of which are in the domain of financial markets. The first case study models market Value at Risk (VaR), using an ensemble of networks. We chose that example, because considerable effort in the finance literature is devoted to VaR models. The key to sound risk management is an accurate VaR model. Here, the distributional features of our model are highlighted. The second case study provides a purchasing and transaction decision support system. The objective is to find a favorable entry point for trades that occur on a regular basis, such as monthly purchases. For example, a corporate treasurer will try to achieve the best possible price in a given time interval. Such trades occur in commodities, such as copper. Pension funds also often must regularly buy an asset, for example bonds. This case study emphasizes the ability to produce multi-step forecasts.

A section presenting other applications follows: this shows that the shared layer perceptron is not limited to forecasting financial markets. The last case study deals with electricity load forecasting. We present an application to forecast the load of combined heat and power plants (CHP). Accurately forecasting this load is very important to achieving a higher share of renewable energy in today's power grids. CHP can be used to balance the irregular electricity generation of wind and solar power. The aim is to centrally control the CHPs so that they can be switched on and off on demand. The challenge here is to not interfere with the interests of the CHP owner, who mainly uses the device to produce warm water, and for whom electricity is only a by-product. Finally, other applications are mentioned. Conclusions and an outlook wrap up the paper.

Literature Review

Considering the evolution of non-linear dynamical systems to improve decisions is certainly not a new idea. See the following: Balestrassi et al. (2009), Crone et al. (2006), Fulcher et al. (2006), Fusai and Roncoroni (2008), Marzi and Turnbull (2007), McNelis (2005), Turban et al. (2010), all of whom provide excellent introductory studies, as well as pointers to more extensive references, especially in the domain of *financial* decision support systems. Nagarajan et al. (2005), Powell (2007), Samarasinghe (2007), Wu and Brynjolfsson (2009) adequately describe the challenges that arise when dealing with forecasting problems. Katsanos (2008), Laidi (2009) emphasize the aspects of complex relationships in the systems one tries to model. In several cases, hybrid decision support systems are presented in which neural networks are combined with other techniques (Andreou et al. 2008, Baesens et al. 2001, Kwon and Moon 2007). Fuzzy approaches are often used (Ang and Quek 2006, Bekiros and Georgoutsos 2007, Lee and Wong 2007) and computational challenges in training are discussed (Mettenheim and Breitner, 2005, 2006).

A significant amount of decision support literature in finance is dedicated to derivative (mostly options) pricing. Accurately pricing derivatives is very important for decisions on risk and hedging purposes. We use approaches presented by Breitner et al. (2007), Gencay and Gibson (2007), Mettenheim and Breitner (2006). Dunis et al. (2005) also provide work on volatility forecasting. Further applications include stocks (Ang and Quek 2006, Chang et al. 2009, Gavrishchaka and Banerjee 2006, Kwon and Moon 2007, Lindemann et al. 2005b, Vanstone and Finnie 2008), foreign exchange (Huang et al. 2007, Huang et al. 2006, Lee and Wong 2007) and credit risk (Baesens et al. 2001). Bekiros and Georgoutsos (2007) present a comparative study using fuzzy methods, among others.

Methodology

The shared layer perceptron is a type of artificial recurrent neural network, see Figure 1. It allows very general models by following a simple state transition formula (Mettenheim 2009, Zimmerman et al. 2006, Zimmerman 2010):

$$s_{t+1} = \tanh(W * s_t)$$

This means that the system state at time $t+1$ is computed by multiplying the state at time t with a single shared weight matrix W and applying the nonlinear hyperbolic tangent to the result vector. The denomination shared layer perceptron is chosen as an analogy to the standard multi-layer perceptron neural network. In our model, there are as many layers as there are time steps, but each layer contains the same weights. The number of variables does not increase when adding layers, that is, time steps. The salient feature of the shared layer perceptron is that state vector s contains observable time series and so-called hidden states. These hidden states act as memory (Schaefer et al. 2007). They can be considered a "heat-bath", absorbing all other influences. The hidden states allow the system to develop its own internal dynamics. This hardens the system against external shocks: the system response to a shock is not purely governed by the new input variables, but also by the accumulated memory. Each output (that is forecast) can act as an input for the next time step. This leads to a *matrix forecast*. All observable time series are obtained for several time steps into the future in a matrix scheme. The number of usable future steps depends on the system under consideration and past learning data.

The model presented here incorporates a special view of the world. First, the various time series in a dynamic system are not independent. For example, markets move together: the main stock indices are not independent of one another, interest rates have an influence, there are currency exchange rates, commodities, and so on. The shared layer perceptron acknowledges this view of the world by providing forecasts, not only for one variable, but for all variables in the system. Second, there is no such thing as a *relevant* or an *irrelevant* variable. (A stock index we want to forecast, for example, does not suddenly become more relevant.) The shared layer perceptron accommodates this view through the *hidden* states, which allow other dynamics to develop.

The topology of the shared layer perceptron leads to robust systems by design. The weight matrix W is not optimized for a single output variable, but for all variables in the system. We carry out tests for each time series. Note that adding more variables (observables) leads to better convergence in the sense of lower training error; see Figure 2. The figure shows that with, for example, ten variables in the system, the achievable training error after 1000 function calls to the optimizer is more than double that of the error achievable with 20 variables. This is at first surprising, because we are fitting more time series with the same number of parameters. It means that the shared layer perceptron is able to isolate dependencies between the time series.

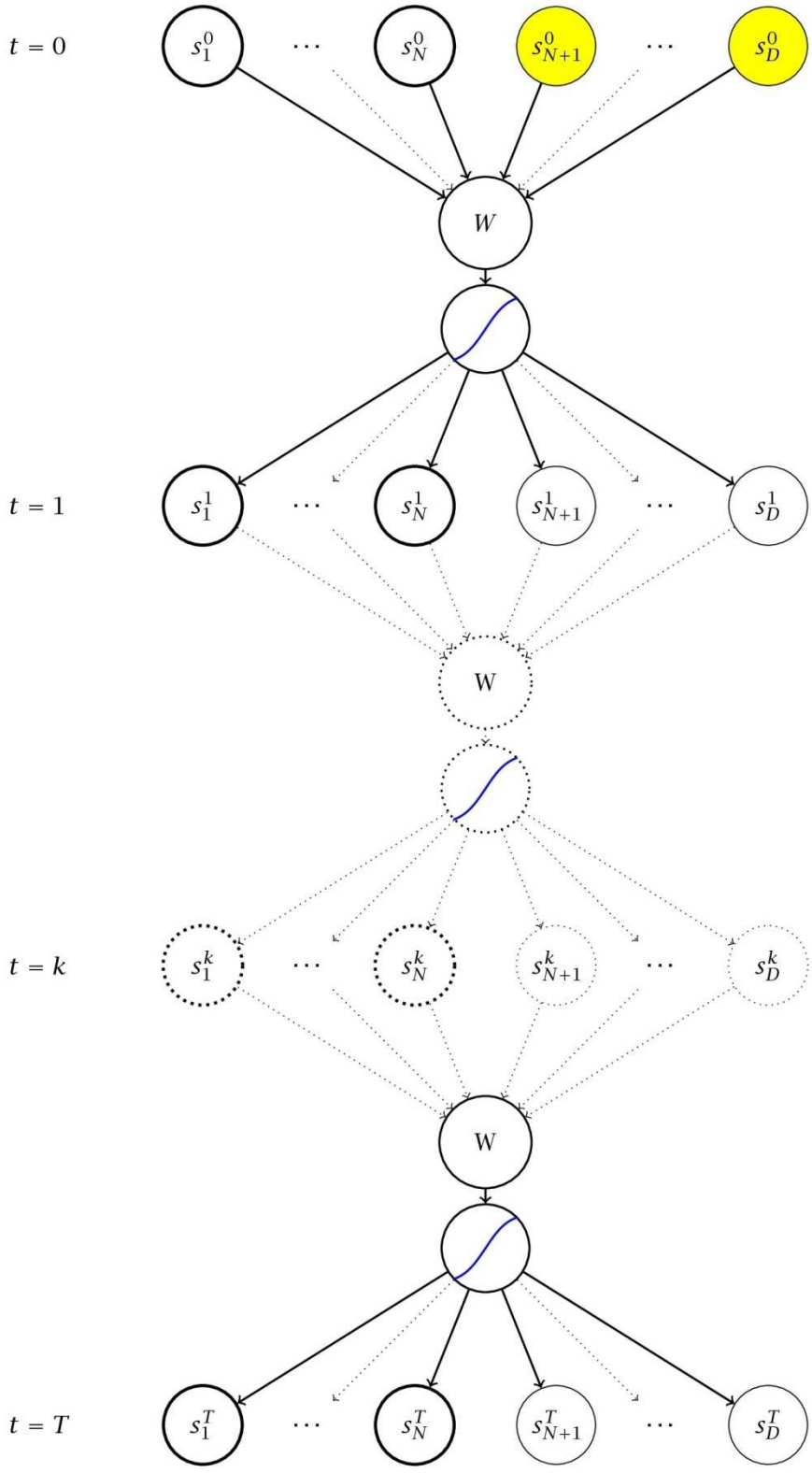

Figure 1. The shared layer perceptron for robust matrix forecasts. A shared weight matrix W simplifies training; hidden states (thin circles) enable creation of a memory.

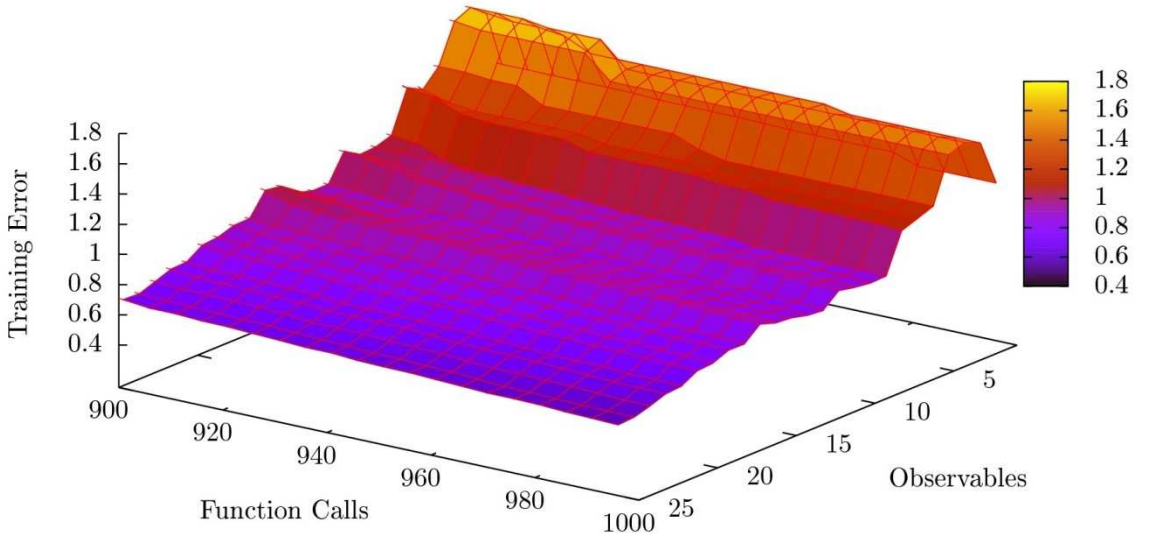

Figure 2. Convergence: the best achievable training error after 1000 function calls improves when more related observables are added. The dataset from the financial case studies is used.

By adequately adjusting the weights it is possible to generate additional explanatory power. This only works if there is a (non-linear) relation between the time series. It does not help when trying to model unrelated time series with the same shared layer perceptron.

Using a *shared* weight matrix improves convergence and robustness, as the number of free parameters in the system is reduced (Mettenheim and Breitner 2005, Steil 2006). When using the shared layer perceptron for decision support, it is also interesting to see not only a point forecast, but also a distribution of possible outcomes (Lindemann et al. 2005b, Miazhynskaia et al. 2008, Tsangari 2007). This is achieved by training several hundred shared layer perceptrons. The resulting expert topology is called an *ensemble*. Each perceptron accurately represents history according to one error criterion. But due to different random initializations of the weight matrix, their forecasts may (and invariably will) differ; see the case study in Figure 3. Having a distribution increases robustness of decisions. The decision maker notices forecast uncertainty when the variance of the distribution becomes large. Then one can use the forecast with additional risk precautions.

The simple state transition equation of the shared layer perceptron alleviates one of the currently perceived disadvantages of neural networks, that is, the black box character. In a typical neural network, several layers (matrices) are concatenated one after the other. Interpreting the result is difficult, often impossible. Interpretability of results is an important feature of robust decision support systems. Using our approach, there is only a single weight matrix. The elements of the matrix can be directly interpreted as influences from past time steps on future time steps. Individual contributions of time series become obvious. This is comparable to a cross-correlation analysis in linear models. A straightforward graphical representation using, for example, a Hinton graph becomes possible. This further simplifies the decision-making process based on the forecast. A decision support system is useful when it not only produces reliable decisions, but also provides rationales for the decision.

A limitation of the presented methodology is that the time series under consideration have to be manually selected. An automatic procedure is currently not implemented. Theoretical arguments from linear systems are helpful. Computing auto-correlation and cross-correlation measures indicate correlated time series. As the shared layer perceptron is capable of mapping every linear correlation (additionally to non-linear correlations) linearly correlated time series are theoretically good candidates for inclusion. This approach is further refined in a principal component analysis or in a stepwise neural regression (Dunis and Chen 2005). These methods aim at maximizing the explained variance of considered time series. Selection of the training dataset has to be considered carefully. The same theory as for linear systems applies: past data is selected such that explanatory power on the validation dataset is maximized. Selecting too much data leads to the inclusion of unnecessary (outdated) information with no relevance for present dynamics. Further theoretical research is warranted in this area.

Modeling Market Value at Risk

VaR is a common measure among finance academics (Hull 2008) and managers (Chan 2009). We use VaR computation as our first application because accurately determining VaR is a key challenge in modern applied finance. The above references devote a considerable amount of space to the presentation of various VaR models alone.

VaR describes the maximum loss to a portfolio that is to be expected within the next *N* days at a confidence level of *X* percent. In the following examples, VaR is calculated for the next ten days at a confidence level of 95 percent. VaR is also a measure for financial regulators: reserve capital requirements are based on VaR. VaR is therefore an important risk management measure, see also Setiono et al. (2006). If the VaR is estimated too high, too much reserve capital is required that could have been used elsewhere. If the VaR is estimated too low, sanctions by the regulator follow. Additionally, internal risk measures may be exceeded and – in the worst case – an institution may be forced into bankruptcy. Estimating VaR is basically a matter of correctly forecasting the probable path of an asset or a portfolio of assets in the next few days. However, current methods are generally linear and not flexible (Chan 2009, Hull 2008). One often used estimator is historical simulation. Historical simulation simply looks at the worst returns from the last 500 days, for example, and takes an appropriate percentile of their distribution.

To model value at risk in the shared layer perceptron framework we require several time series of input data. This data is shown in Table 1. As no commonly agreed upon theory of the financial markets exists, time series from several different instrument types and regions are chosen. The choice follows the methodology from Dunis and Chen (2005), Dunis et al. (2006), and Dunis et al. (2008).

Selection is based on key variables of worldwide financial markets. These include the major stock indices from different regions of the world, relevant interest rates, the most common exchange rates, and commodities. To facilitate reproduction of our study, mnemonics as used by Thomson Reuters Datastream are also provided in Table 1. These mnemonics are used in the following.

To conserve space, we do not detail the reasons behind the inclusion of each variable. Instead we confine ourselves to a general motivation. Stock indices from Europe and North America are widely followed. Arguably they are the most relevant indicators of a financial system. We include the South Korean Kospi index because it is generally considered to be a leading indicator. This is partly due to South Korea's important position in ship building. Including short- and long-term interest rates allows us to build a yield curve, i.e., long-term rate minus short-term rate. In the following, yield curves are denoted by the suffix "yc" in the mnemonic, e.g. USyc for United States yield curve. Yield curves for different countries are used in the following examples. Widely followed interest rates include the German Bund and 10-year US Treasuries. The market for exchange rates is the most liquid worldwide. However, most liquidity comes from speculation and only 5 % comes from transactions in the real economy. Finally, commodities are also considered leading indicators. The Commodity Research Bureau (CRB) index averages several commodities, mostly traded in the United States. The Baltic Exchange Dry Index measures bulk freight rates and is considered a long term leading indicator.

As is common for financial time series, all variables are non-normally distributed. This is confirmed by the Jarque-Bera statistic at the 99-percent confidence level. Most series exhibit slight skew and significant kurtosis, i.e., fat tails. The series are at least bimodal and in some cases multi-modal. Furthermore, the level series are non-stationary, that is, they have non-constant mean and variance over time. This is confirmed by the augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) test statistics. For this reason, the series are transformed into returns, as is common in financial time series analyses (Lindemann et al. 2005a). The transformed series are stationary, as shown in Table 2. All return series are non-normal and fat tailed.

We use the time series of Table 1 to train 300 shared layer perceptrons. Our data spans ten years of daily observations from July 1999 to June 2009, obtained from Thomson Reuters Datastream. We produce rolling window forecasts, that is, we use 440 days of training data to generate VaR forecasts for the next 110 days, which means for day 441 to day 550. A sample result for the FTSE100 stock index is shown in Figure 3. This type of computation is produced for every time series. Figure 3 shows the worst 10-day (i.e. most *negative*) returns, in order of magnitude for each time step. For a single shared layer perceptron, the worst 10-day return is obtained by constructing the portfolio path in a multi-step forecast and taking the lowest value. Therefore, even if the portfolio rises continuously, the "worst" return is at best 1.0. For our VaR analysis, we are interested in the lowest percentiles of the worst returns distribution, i.e. the network number 15 for a 95 percent VaR (used for the example) and the network number 3 for a 99 percent VaR (not shown here). This means we "cut" through the lower end of the graph in Figure 3.

The resulting VaR can be found on Figure 4. The thick blue line corresponds to the target. Two objectives are met. First, we want our forecast to be as close as possible to the blue line. Second, we do not want our forecast to violate the target. This means we want to avoid real returns being *worse* than our forecasts. From Figure 4 we note that the benchmark historical simulation (red line) never violates the target. However, historical simulation is an almost constant estimator. It does not represent the dynamics in the underlying system. Similarly, it is often far away from the target, because observations from the distant past still have a significant influence on historical VaR. On the other hand, the forecast using the shared layer perceptron (green line) models the dynamics appropriately. We notice violations on five occasions. However, over a time span of 110 days, five violations are within the confidence interval of 95 percent. Additionally, we provide a forecast obtained by smoothing the original forecast using a moving average of ten days. This has the effect of avoiding violations, but some dynamics are lost.

Table 3 shows results for all assets under consideration. Yield curves were not included, because to really trade on this, one has to construct a derived portfolio that tracks the effect of yield curve changes. This can be realized, for example, with a swap. It unnecessarily complicates the exposure without adding anything important.

Table 1. Time Series Used in Application, Ordered by Instrument Type

Name	Instrument	Region	Mnemonic
FTSE 100 Index	Equities	United Kingdom	FTSE100
DAX 30 Index	Equities	Germany	DAXINDX
CAC 40 Index	Equities	France	FRCAC40
FTSE MIB	Equities	Italy	FTSEMIB
Dow Jones Euro Stoxx 50	Equities	Europe	DJES50I
S&P 500 Index	Equities	United States	S&PCOMP
NASDAQ 100 Index	Equities	United States	NASA100
Nikkei 225 Index	Equities	Japan	JAPDOWA
Kospi Index	Equities	South Korea	KORCOMP
Three Month LIBOR	Interest rate	United Kingdom	ECUKL3M
Twelve Month LIBOR	Interest rate	United Kingdom	BBGBP12
Germany Three Months	Interest rate	Germany	ECWGM3M
France Three Months	Interest rate	France	ECFFR3M
Italy Three Months	Interest rate	Italy	ECITL3M
EURIBOR Three Months	Interest rate	Euro area	ECEUR3M
Eurodollars Three Months	Interest rate	United States (Europe)	ECUS$3M
Benchmark Bond Three Months	Interest rate	Japan	ECJAP3M
Benchmark Bond 10 Years	Interest rate	United Kingdom	UKMBRYD
Bund Future 10 Years	Interest rate	Germany	BDBRYLD
Benchmark Bond 10 Years	Interest rate	France	FRBRYLD
Benchmark Bond 10 Years	Interest rate	Italy	IBRYLD
US Treasuries 10 Years	Interest rate	United States	USBD10Y
Benchmark Bond 10 Years	Interest rate	Japan	JPBRYLD
US Dollar to Great British Pound	Exchange rate	United Kingdom	USDOLLR
US Dollar to Swiss Franc	Exchange rate	Switzerland	SWISFUS
US Dollar to Euro	Exchange rate	Euro area	USEURSP
Yen to US Dollar	Exchange rate	Japan	JAPAYEN$
Gold Bullion	Commodity	United Kingdom (world)	GOLDBLN
Brent Crude Oil	Commodity	Europe (world)	OILBREN
CRB Index	Commodities	United States (world)	NYFECRB
Baltic Exchange Dry Index	Commodities	World	BALTICF

Table 3 shows the sum of squared deviations (SSD), which is obtained by accumulating squared *errors*, i.e., deviations, in the forecast. The mean SSD is obtained by dividing SSD by the number of days under consideration. The smaller these numbers, the better the forecast approximates the true value. This is important to lower the employed risk capital but also to avoid violations. The number of violations is in line with what can be expected from the relevant confidence levels and is not shown here.

We note that over a time span of 110 days, the smoothed forecast (which is produced by using the ten day moving average) always beats historical simulation. The pure forecast is often better (in 14 of the 17 time series). Even when we consider a time span of eight years, the smoothed forecast still beats the benchmark for 12 out of 17 time series. This is achieved without ever *retraining* the model. This means that the shared layer perceptron is obviously able to use its built-in memory to adapt to situations not directly observed in training data.

Our objective is to find a measure that reproduces the *real* encountered risk accurately. We are not interested in conservative values. Conservative models like the benchmark historical simulation are useful and necessary in their own right, especially because they are widely accepted. Financial institutions are, however, also interested in the minimum value of the portfolio that will likely occur during the next ten days. Our view is that for a financial institution involved in real investment or trading, a model that adequately represents the portfolio over the next ten days (while putting up with the occasional violation that may occur) is more useful than a very conservative model that obliges the institution to hold a much higher amount of risk capital in reserve.

Table 2. Stationarity analysis for returns: all returns are stationary. This is confirmed by both the augmented Dickey-Fuller (ADF) and the Phillips-Perron (PP) test statistic at the 99% significance level. Stationarity analysis for the level series shows that levels are not stationary. Stationarity translates into a mean of zero and is helpful in neural network training.

Mnemonic	ADF	p-value	PP	p-value
FTSE100	-14.51	0.01	-55.08	0.01
DAXINDX	-13.66	0.01	-53.27	0.01
FRCAC40	-14.16	0.01	-54.00	0.01
FTSEMIB	-13.17	0.01	-52.08	0.01
DJES50I	-14.14	0.01	-53.98	0.01
SPCOMP	-13.82	0.01	-56.52	0.01
NASA100	-12.89	0.01	-55.84	0.01
JAPDOWA	-13.79	0.01	-51.94	0.01
KORCOMP	-14.46	0.01	-50.23	0.01
BBGBP12	-10.15	0.01	-44.59	0.01
ECEUR3M	-9.11	0.01	-69.35	0.01
UKyc	-13.68	0.01	-58.84	0.01
GERyc	-13.17	0.01	-54.00	0.01
FRyc	-13.10	0.01	-56.06	0.01
ITyc	-12.79	0.01	-53.65	0.01
USyc	-11.52	0.01	-54.66	0.01
JAPyc	-13.35	0.01	-59.54	0.01
USDOLLR	-13.78	0.01	-47.98	0.01
SWISFUS	-13.74	0.01	-54.53	0.01
USEURSP	-13.36	0.01	-49.85	0.01
JAPAYEUSD	-12.97	0.01	-51.98	0.01
GOLDBLN	-14.35	0.01	-51.16	0.01
OILBREN	-12.44	0.01	-51.81	0.01
NYFECRB	-13.81	0.01	-50.28	0.01
BALTICF	-9.14	0.01	-14.13	0.01

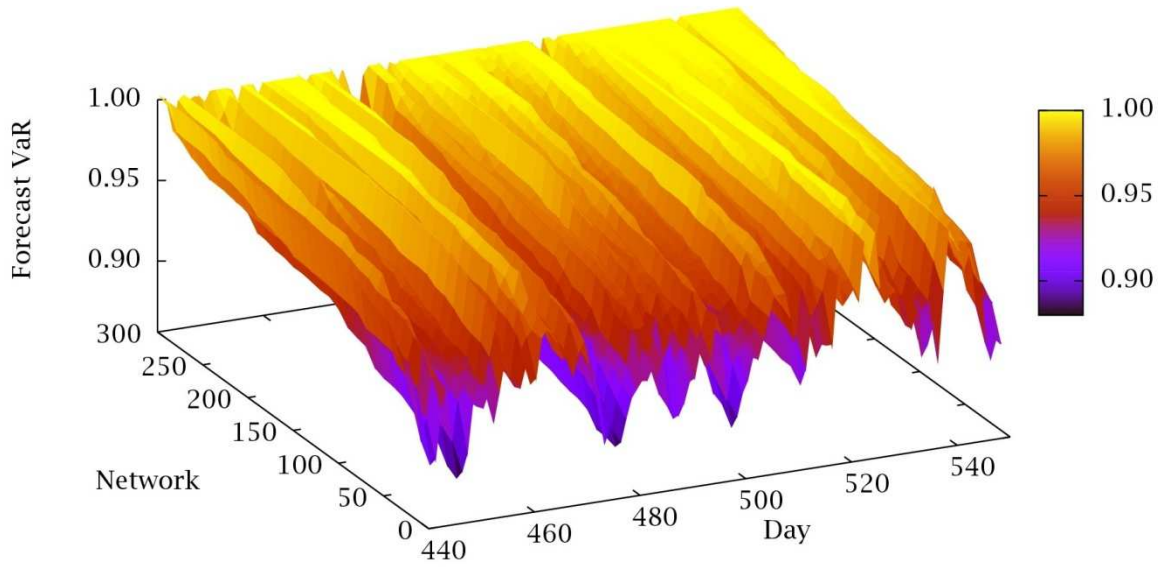

Figure 3. 10-day value at risk forecasts by shared layer perceptron ensemble for the FTSE100 stock index. Forecasts are ordered by magnitude. To determine value at risk we cut through the lower end of the return distribution. For example to determine the 95 percent forecast VaR we cut through the distribution at network 15 (because 15 corresponds to five percent of the entire ensemble of 300 networks). This method is used to produce Figure 4. Here, a forecast over 110 future time steps is presented. 440 daily observations of past data are used for training.

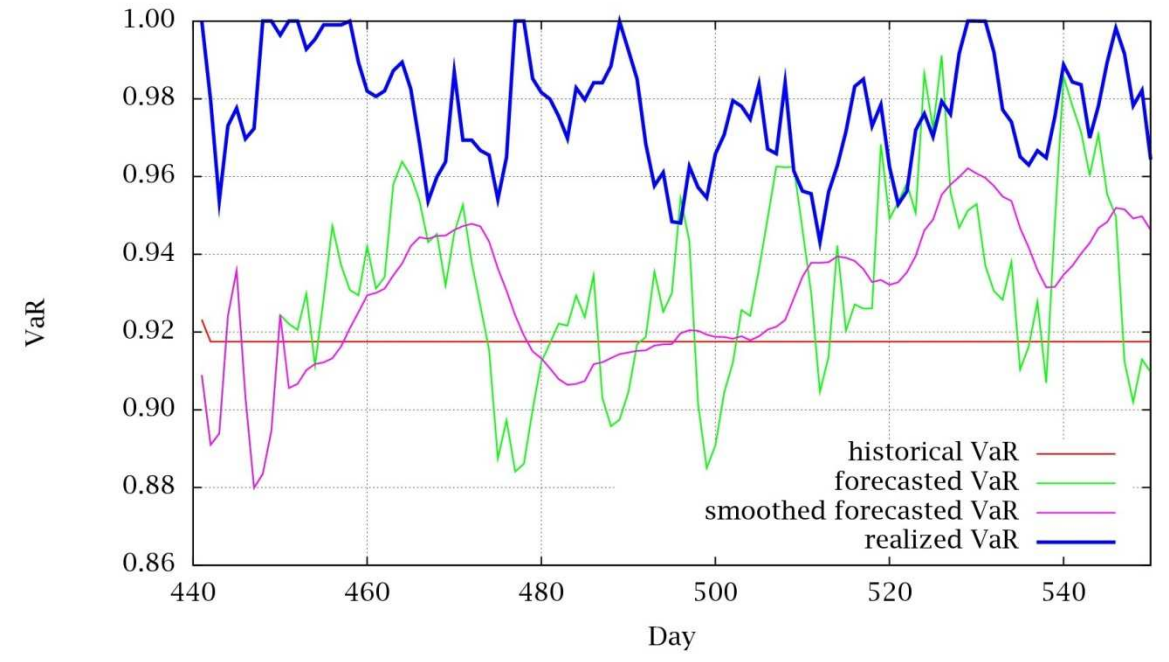

Figure 4. Comparison of historical, forecasted and smoothed forecasted VaR for the FTSE100, 95 percent confidence interval, 10-days ahead. The forecasts are obtained by cutting through the distribution as described in Figure 3. Consider that the smoothed forecasted VaR is more flexible than the benchmark historical VaR which almost never changes in the time span under consideration.

Table 3. 10 days, 95 percent VaR benchmark measures. The sum of squared deviations (SSD) is always less for the smoothed neural network forecast (using a ten day moving average) as compared to the benchmark historical simulation when considering a forecasting time span of 110 days (excesses are highlighted in red). Only over a time span of eight years do we have five out of a 17 time series in which the deviations of the smoothed neural network forecast exceed the benchmark.
Legend: hist = historical simulation, fc = shared layer perceptron forecast, smooth = smoothed shared layer forecast

Mnemonic	Time	SSD			Mean SSD		
		hist	fc	smooth	hist	fc	smooth
FTSE100	110 days	0.414	0.336	0.318	0.00376	0.00305	0.00289
	8 years	17.546	9.053	8.083	0.00817	0.00421	0.00376
DAXINDX	110 days	0.594	0.55	0.473	0.0054	0.005	0.0043
	8 years	30.215	13.089	11.733	0.01407	0.00609	0.00546
FRCAC40	110 days	0.389	0.42	0.352	0.00354	0.00382	0.0032
	8 years	23.68	10.487	9.03	0.01102	0.00488	0.0042
FTSEMIB	110 days	0.446	0.463	0.398	0.00405	0.00421	0.00361
	8 years	26.245	12.943	11.586	0.01222	0.00603	0.00539
DJES50I	110 days	0.362	0.104	0.081	0.00329	0.00095	0.00073
	8 years	24.213	3.784	3.084	0.01127	0.00176	0.00144
SPCOMP	110 days	0.645	0.544	0.52	0.00586	0.00495	0.00473
	8 years	14.941	12.158	10.988	0.00696	0.00566	0.00512
NASA100	110 days	3.21	2.241	2.217	0.02919	0.02038	0.02015
	8 years	25.722	75.732	70.908	0.01198	0.03526	0.03301
JAPDOWA	110 days	0.674	0.064	0.084	0.00613	0.00058	0.00076
	8 years	21.521	4.299	3.619	0.01002	0.002	0.00168
KORCOMP	110 days	2.018	1.744	1.664	0.01835	0.01586	0.01513
	8 years	29.087	61.269	56.884	0.01354	0.02852	0.02648
USDOLLR	110 days	0.118	0.083	0.077	0.00107	0.00076	0.0007
	8 years	2.871	2.251	2.046	0.00134	0.00105	0.00095
SWISFUS	110 days	0.131	0.032	0.027	0.00119	0.00029	0.00024
	8 years	2.192	1.026	0.825	0.00102	0.00048	0.00038
USEURSP	110 days	0.176	0.174	0.156	0.0016	0.00158	0.00142
	8 years	2.723	5.642	5.163	0.00127	0.00263	0.0024
JAPAYEUSD	110 days	0.118	0.12	0.102	0.00108	0.00109	0.00093
	8 years	3.184	4.694	4.281	0.00148	0.00219	0.00199
GOLDBLN	110 days	0.409	0.202	0.18	0.00372	0.00184	0.00163
	8 years	10.69	3.927	3.321	0.00498	0.00183	0.00155
OILBREN	110 days	3.776	2.7	2.359	0.03433	0.02455	0.02145
	8 years	55.391	67.252	62.115	0.02579	0.03131	0.02892
NYFECRB	110 days	0.16	0.131	0.124	0.00145	0.00119	0.00112
	8 years	6.57	3.226	2.947	0.00306	0.0015	0.00137
BALTICF	110 days	0.439	0.17	0.159	0.00399	0.00155	0.00145
	8 Years	79.787	9.257	8.629	0.03715	0.00431	0.00402

Consider that additional optimizations have not been conducted. For example, we can optimize the smoothing period or use additional validation data to optimize the number of violations. Doing this separately for each asset can considerably improve forecasts for individual assets. However, this goes against the spirit of a robust model. We prefer to use a model that has been demonstrated to work well for *all* assets without curve fitting to a specific one. In the context of a decision support system in which only a few assets (in contrast to an entire market) are of interest, there may be a good reason to improve forecasts of these assets at the expense of others. The shared layer perceptron can accommodate this via weightings in the error function. Some of our experiments show that performance then degrades in the long run, rendering continuous retraining mandatory.

To summarize, key benefits of our VaR forecasting approach are:

- Multi-step forecasts offer a complete view of the value path of the portfolio.
- A single model can be used for different confidence levels. Once the ensemble of shared layer perceptrons is trained, the user gets every percentile of the distribution simultaneously: they only need choose a suitable point in the distribution of returns. This alleviates one of the deficiencies of standard VaR. It only answers the question: What will most likely *not* happen to the portfolio within the next ten days? It does not answer the question: If the worst happens, how bad will it be?
- A single model is used for all assets. If it works simultaneously over a broad range of assets, a person will have more confidence in using it. The shared layer perceptron works well for all inputs by design.
- The model produces very good results over a short time span and still achieves good results over a significantly longer time span without retraining. This means that performance degrades gracefully. There is no point in creating catastrophically wrong forecasts.

As a follow-up to the results in the present paper we carry out a benchmarking of our method against advanced Copula-GARCH models in Breitner et al. (2010) using exactly the same network topology and VaR calculation method. In total we test 18 different models on a weighted portfolio of benchmark equity indices. The test period includes 20 years of data. We retrain the neural networks every six months. The results further validate our approach: while the networks do not beat each benchmark model in terms of violations or deviations the results are, again, consistent. The neural network model is the only one for which conditional coverage and independence cannot be rejected at all significance levels. The various GARCH models offer results which fluctuate significantly, ranging from satisfactory to unacceptable. The test includes only one neural network model which was not tuned in any way. On the other hand it is only with the benefit of hindsight that we can estimate the best GARCH model.

Further research will especially address the question of retraining: How often should we retrain the networks? Retraining often is desirable because the model can adapt to the newest data. However, unnecessary training is a waste of computational resources. Another limitation of the current approach is that the extreme percentiles of the shared layer perceptron distributions are considered. Only a limited number of networks is available. The very important theoretical question concerning the probability of outliers is not addressed here. This problem is partly mitigated by training more perceptrons.

Purchasing and Transaction Decision Support

This section outlines a typical decision problem for a corporate treasurer: regularly, for example monthly, some kind of transaction has to be undertaken. This may be an investment in equities for a pension plan or a fixed income placement. It may be a regular foreign exchange transaction to pay monthly costs in another currency. Or it can be the monthly supply of some commodity, such as fuel or metal. This case often happens in different industries. What all these cases have in common is that the treasurer must choose an appropriate time for the transaction. This is, of course, the day on which the price is the most favorable. We want to buy equities, foreign currencies and commodities at the *lowest* price within some time frame. Here, a multi-step forecast proves useful, because it gives an idea of probable price movements.

In the following, we look at a 20-day multi-step forecast for different assets. The benchmark to evaluate the quality of market timing is the realized potential (RP). RP is a number between 0 and 1, where 0 indicates that the transaction is forecast to take place at the worst possible moment. RP = 1 indicates that the user gets the best possible price. To simplify, we now assume that we are buying and that the *best* price is equal to the *lowest* achievable price.

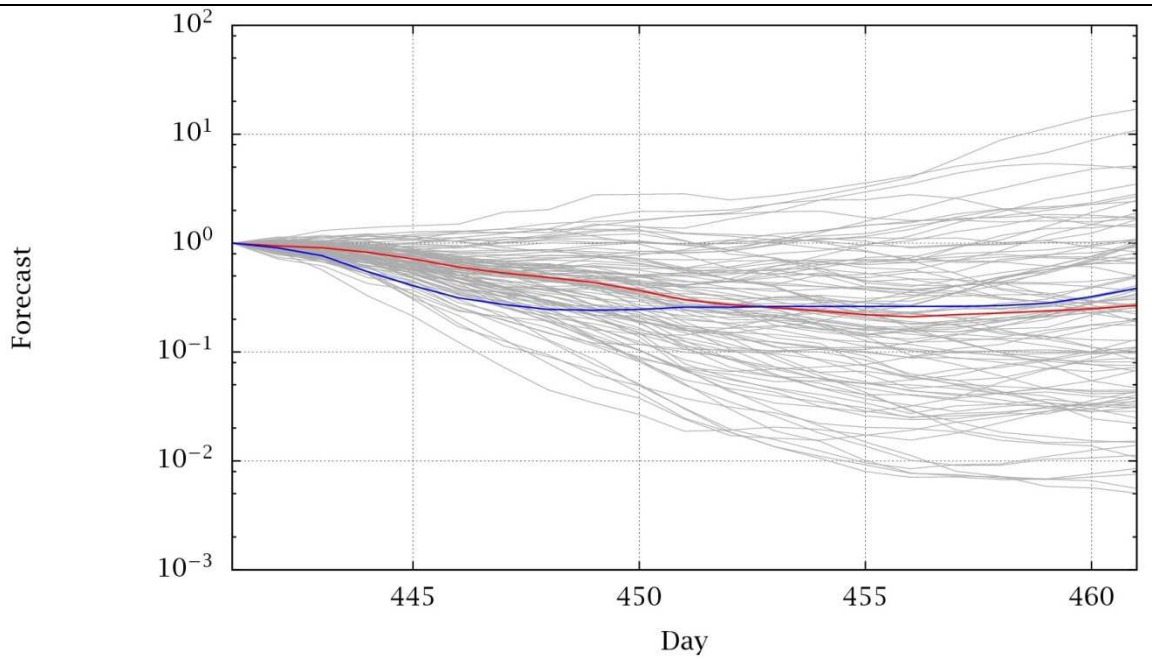

Figure 5. 20-day forecast for the Baltic Dry Index. Actual values in blue, forecast in red as average over all ensemble members. The gray lines show single members. With increasing distance into the future uncertainty increases. 440 past daily observations are used to produce this forecast.

Then, the 20-day realized potential is defined at time *t* of a transaction as

$$RP_t(20) = 1 - (p_t^{realized} - p_t^{min}(20)) / (p_t^{max}(20) - p_t^{min}(20)).$$

$p_t^{max}(20)$ and $p_t^{min}(20)$ represent the maximum and minimum prices in the 20-day window starting at time *t*. $p_t^{realized}$ is the price realized when following the forecast at time *t*. Please note that $p_t^{realized}$ is not the forecast price, but the actual price at which it would have been possible to trade. The shared layer perceptron is used to make a 20-day forecast and then *RP* is evaluated. Then, the perceptron is updated with new data from the day and *t* moves one time step forward.

Figure 5 presents results for a 20-day forecast involving the Baltic Dry Index. Although the lowest price is realized around day 447 and the shared layer perceptron forecasts a low at around day 456, it is still a very good time to buy shipping capacity. The forecast avoids the high prices at the beginning of the time period being considered. The suggested entry point is not very time sensitive. The price rise at the end of the investigated period is also avoided. The forecast is obtained by averaging over all members of the ensemble. The rationale of averaging is that a priori every ensemble model accurately describes history and there is no reason to prefer one over another. Ensemble averaging is the method employed by Bayes (Zimmermann 2010). There is more information contained in the forecast than visible in a single number: The gray lines in Figure 5 show forecasts of individual ensemble members. We see that during the first five days of the forecast the ensemble moves very closely together. With increasing distance into the future divergences appear, that is, uncertainty increases. That is natural. It is a warning flag for the decision maker not to focus on the single point forecast but rather on the distribution.

We compare RP as achieved by a shared layer perceptron forecast over a time span of 20 days, with RP achieved by always buying on the same (fixed) day of a month, for example, by always buying on the 15th of the month. The excess (or otherwise) RP is then cumulated over the out-of-sample time span and compared for all assets and every fixed day in a month (assuming 20 business days in a month). All cumulated excess RP are positive. This means that the transaction decision made using the forecast always adds value compared to the commonly used fixed day strategy. These results are obtained by training the shared layer perceptron using 440 past daily values and evaluating forecasts for the next 110 days.

Note that excess RP is positive regardless of the asset under consideration. The shared layer perceptron provides robust decisions in the sense that high decision quality is achievable without tuning (or over fitting) the model to a specific asset. Robustness translates into confidence in using the model in new situations.

However, even when the out-of-sample forecast is extended to a period of eight years, the results are still very satisfactory, with most assets outperforming for most fixed day strategies. DAX 30, CAC 40, FTSE MIB, S&P500, NASDAQ 100, 12-month LIBOR and three-month EURIBOR, UK yield curve, Gold Bullion, CRB Index and Baltic Exchange Dry Index are very good performers without any negative cumulated RP, that is, the forecast is always better than a fixed day strategy. We also note that currencies are difficult to predict over the long term. Foreign exchange is notorious for having very little exploitable inefficiencies (Yu et al. 2007a). Especially EUR|USD underperforms in a majority of cases for the time span of eight years. Interestingly, USD|JPY performs surprisingly well, see also (Gagnon and Chabound 2007). There are only four fixed days of negative performance. The satisfactory performance is attributed to the internal dynamics modeled with hidden states. From a theoretical viewpoint, it is better to regularly retrain the model. This updates the weight matrices and allows adapting to shifts in the underlying dynamics. When not retraining, learning only occurs through short term memory in hidden states.

In conclusion, we emphasize that the results are very satisfying. For the short term forecast of 110 days, the shared layer perceptron forecast always beats the benchmark. It shows very consistent performance. What is even more satisfying is that a single ensemble of perceptrons shows robust performance over a time span of eight years. Training data only spans two years. Although performance is not always positive for all assets, for a considerable number of assets, especially commodities and Western equity indices, performance is much better than the benchmark.

Electricity Load Forecast and Other Applications

Up to now we have only focused on two detailed case studies for financial time series. In this short paragraph we briefly present other applications (complex non-linear dynamic systems) that can profit from the robustness of shared layer perceptrons.

One dynamic system for which robust decision support systems are very relevant is the domain of electricity production and load (Kamper and Eßer 2010). Electrical utility companies have to announce the load they want to purchase 24 hours in advance, at a granularity of 15 minutes. Producers also have to announce the quantity they want to sell at the same granularity. While this problem has already been adequately solved for conventional energy sources, the concept of smart grids introduces significant challenges when alternative energies such as solar and wind power are involved. The difficulty lies in the fact that energy produced by alternative energies is not easy to forecast: it varies depending on the weather, and exact local weather forecasts still lack precision.

Because of this, the concept of a virtual power plant (VPP) is introduced. A VPP includes different solar and wind power sources, but also CHPs. The CHP can in principle be used to account for deficits in solar and wind energy production. However, CHPs are mostly used for heat production: electricity is a by-product. An important goal when using smart grids is therefore the accurate forecast of CHP load that induces electricity production.

Here, we introduce the possibility of slightly altering heat production so that the combined output of solar, wind and CHP follows some previously announced path. This means that when too much power is produced from wind and solar sources, we reduce CHP load. If there is a lack of power due to some unexpected absence of wind or sun, we increase CHP load. Figure 6 shows forecasts of CHP load for 240 hours, or ten days. This forecast is made 24 hours in advance. It is important to note that the shared layer perceptron accurately forecasts the turning points, that is, the local maxima and minima at which load starts to decrease or increase. The time series used in this forecast also includes weather data such as temperature, wind speed, wind chill, and global radiation, among other factors. The application to VPP shows that shared layer perceptrons can also contribute to better (and significant) utilization of alternative energy sources.

Further applications include dynamic systems in which fast multi-step and multivariate decisions are required. Optimal control problems can be modeled through simulation. For example the Game of Two Cars, a classic dynamic game, generally not solvable in real-time, can be tackled. This leads to a collision avoidance system. Space shuttle guidance is another example.

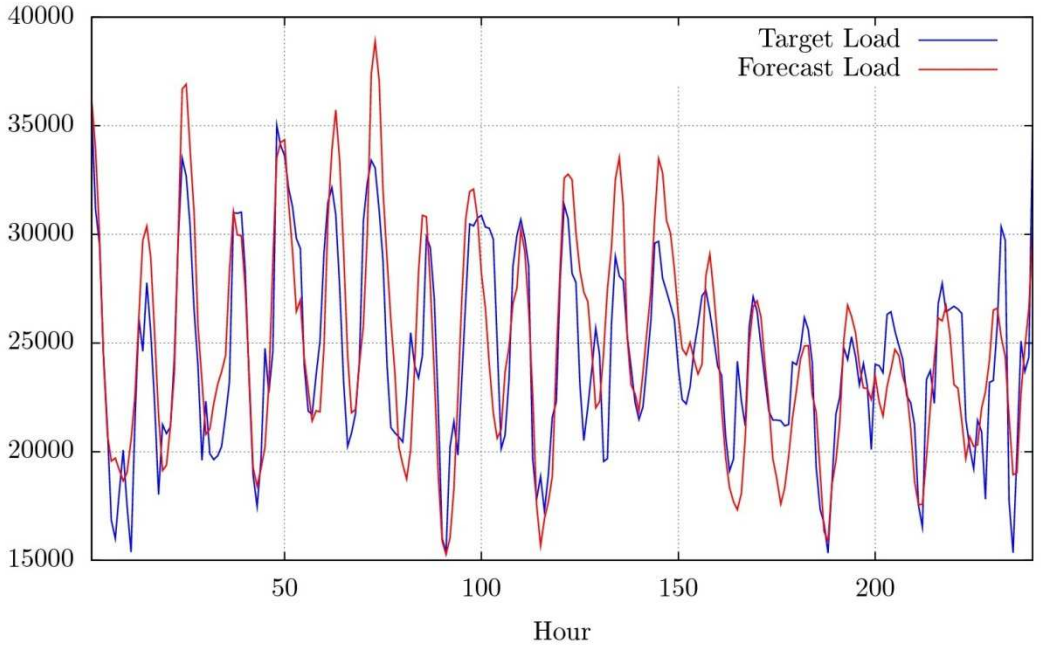

Figure 6. Case study: electricity load forecast of combined heat and power plants.

Finally, there are other financial applications that have not yet been mentioned. The shared layer perceptron can also be employed to create profitable (after transaction costs) trading systems. Because it produces a distribution of forecasts, it is also used for option pricing and beats classic models based on Black Scholes in accuracy, see also (Pires and Marwala 2005). Long-term forecasts of macro-economic variables also prove promising. Readers interested in more details are invited to contact us.

Conclusions and Outlook

In this paper we provide a decision support and modeling approach for dynamic systems. The shared layer perceptron presented here emphasizes the robustness of modeling. It avoids the weaknesses of other modeling approaches identified in the introduction. Robustness is achieved by not fitting (or over fitting) the model to a specific time series. Instead, all time series in the model are equally trained and have to be forecast equally well. Trust in the model grows. In our first two case studies involving financial applications, we show that the shared layer perceptron is able to accurately model several time series simultaneously. We are also able to create multi-step forecasts. Each forecast serves as input for the next time step and a transaction decision support system shows that our approach is capable of correctly forecasting entry and exit points.

Another mechanism also ensures robustness. We do not rely on a single model, rather on an ensemble of models. Each member model is a priori equally well-suited to forecasting. By using an ensemble, we get a distribution of possible outcomes that is data dependent, as it has not been chosen a priori. This helps to support risk management decisions, as exemplified by the determination of value at risk via the ensemble in the first case study. It is still possible to extract a single number by averaging the ensemble. Several other applications of the shared layer perceptron are also described, such as a case study on electricity load forecasts for combined heat and power plants. These forecasts are important for incorporating renewable energy sources into smart grids. The shared layer perceptron is not limited to forecasting financial time series.

Our approach provides a useful addition to the theorist's toolbox. When the underlying dynamics of a system are not known, modeling with shared layer perceptrons is advisable due to the flexibility and robustness of the model. Using a single shared weight matrix, it is even possible to interpret the weights and extract causalities. This leads to a gray box model, rather than a black box model. The shared weight matrix delivers a multi-causal model. That is, we are able to get a sense of the influence of every time series in the model on every other time series.

The structure of the shared layer perceptron is also easy to interpret. It uses non-linear mapping from one system state to another. We have demonstrated a system that is able to map seemingly complicated dynamic systems, is robust, is still comprehensible and provides useful decision support in terms of added value. To put it in the words of Hevner et al. (2004), the implemented shared layer perceptron is a "new and innovative artifact" that "extends human and organizational capabilities".

The potential of the shared layer perceptron is far from being exploited. It is necessary to better ground the shared layer perceptron in the theory of decision support for dynamical systems. Further research will focus on broadening the field of application by performing relevant case studies. It is especially important to analyze how our model can improve decisions in other real-life scenarios. There are plans to integrate our VaR model at a local bank. We will implement the electricity load forecasting application at a leading energy services provider.

References

Ang, K. K., Quek, C. "Stock trading using rsprop: A novel rough set-based neuro-fuzzy approach", *IEEE Transactions on Neural Networks*(17:5), September 2006, pp. 1301-1315.

Baesens, B., Setiono, R., Mues, C., Viaene, S., Vanthienen, J. "Building Credit-Risk Evaluation Expert Systems Using Neural Network Rule Extraction and Decision Tables", in *Proceedings of the Twenty-Second International Conference on Information Systems*, New Orleans, Louisiana, December 2001, pp. 159-168.

Balestrassi, P. P., Popova, E., Paiva, A. P., Lima, J. W. M. "Design of experiments on neural network's training for nonlinear time series forecasting", *Neurocomputing*(72:4-6), 2009, pp. 1160-1178.

Bekiros, S. D., Georgoutsos, D. A. "Evaluating direction-of-change forecasting: Neurofuzzy vs. Neural Networks", *Mathematical and Computer Modelling*(46), 2007, pp. 38-46.

Bookstaber, R. *A Demon of our own Design: Markets, Hedge Funds, and the Perils of Financial Innovation*, Wiley, Hoboken, New Jersey, 2007.

Breitner, M. H., Köller, F., König, S., Mettenheim, H.-J. von "Intelligent decision support systems and neurosimulators: A promising alliance for financial services providers", in *Proceedings of the Fifteenth European Conference on Information Systems*, Österle, H., Schelp, J., Winter, R. (Eds.), St. Gallen, June 2007, pp. 478-489.

Breitner, M. H., Luedtke, C., Mettenheim, H.-J. von, Rösch, D., Sibbertsen, P., Tymchenko, G. "Modeling Portfolio Value at Risk with Statistical and Neural Network Approaches", in *Proceedings of the 17th Forecasting Financial Markets Conference: Advances for Exchange Rates, Interest Rates and Asset Management*, Dunis, C., Dempster, M., Breitner, M. H., Rösch, D., Mettenheim, H.-J. von (Eds), Leibniz Universität Hannover, May 2010.

Chan, E. P. *Quantitative Trading*, Wiley, Hoboken, New Jersey, 2009.

Chang, P. C., Liu, C. H., Lin, J. L., Fan, C. Y., Ng, C. S. P. "A neural network with a case based dynamic window for stock trading prediction", *Expert Systems with Applications*(36), 2009, pp. 6889-6898.

Crone, S. F., Lessmann, S., Pietsch, S. "Forecasting with computational intelligence – an evaluation of support vector regression and artificial neural networks for time series prediction", in *Proceedings of the International Joint Conference on Neural Networks*, 2006, pp. 3159-3166.

Dunis, C. L., Chen, Y. X. "Alternative volatility models for risk management and trading: An application to the EUR/USD and USD/JPY rates", *Derivatives Use, Trading and Regulation* (11:2), 2005, pp. 126-156.

Dunis, C. L., Laws, J., Evans, B. "Modeling and trading the gasoline crack spread: A non-linear story", *Derivatives Use, Trading and Regulation* (12:1-2), 2006, pp. 126-147.

Dunis, C. L., Laws, J., Evans, B. "Trading futures spread portfolios: applications of higher order and recurrent networks", *The European Journal of Finance* (14:6), 2008, pp. 503-521.

Fulcher, J., Zhang, M., Xu, S. "Application of higher-order neural networks to financial time-series prediction", in *Artificial Neural Networks in Finance and Manufacturing,* Kamruzzaman, J., Begg, R. K., Sarker, R. A. (Eds.), Idea Group Publishing, Hershey, 2006, pp. 80-108.

Fusai, G., Roncoroni, A. *Implementing Models in Quantitative Finance: Methods and Cases*, Springer, Berlin, 2008.

Gagnon, J. E., Chaboud, A. P. "What can the data tell us about carry trades in Japanese Yen?", *International Finance Discussion Papers* (889), The Board of Governors of the Federal Reserve System, July 2007.

Gavrishchaka, V. V., Banerjee, S. "Support vector machines as an efficient framework for stock market volatility forecasting", *Computational Management Sciences* (3:2), 2006, pp. 147-160.

Gencay, R., Gibson, R. "Model risk for European-style stock index options", *IEEE Transaction on Neural Networks* (18:1), January 2007, pp. 193-202.

Haykin, S. *Neural Networks and Learning Machines*, Pearson Education, Upper Saddle River, New Jersey, 2009.

Hevner, A. R., March, S. T., Park, J., Ram, S. "Design Science in Information Systems Research", *MIS Quarterly* (28:1), March 2004, pp. 75-105.

Hevner, A. R. "A Three Cycle View of Design Science Research", *Scandinavian Journal of Information Systems* (19:2), 2007, pp. 87-92.

Huang, W., Lai, K. K., Wang, S. "Application of neural networks for foreign exchange rates forecasting with noise reduction", *Lecture Notes in Computer Science* (4488), 2007, pp. 455-461.

Huang, W., Wang, S., Zhang, H., Xiao, R. "Selection of the appropriate lag structure of foreign exchange rates forecasting based on autocorrelation coefficient", *Lecture Notes in Computer Science* (3979), 2006, pp. 512-517.

Hull, J. C. *Options, Futures and Other Derivatives*, Prentice Hall, Upper Saddle River, 2008.

Kamper, A., Eßer, A. "Strategies for Decentralised Balancing Power", *Biologically-Inspired Optimisation Methods*, *Studies in Computational Intelligence* (210), 2010, pp. 261-289, Springer, Berlin, Heidelberg.

Katsanos, M. *Intermarket Trading Strategies*, Wiley, Chichester, 2008.

Khandani, A. E., Lo, A. W. "What happened to the quants in August 2007?", *Journal of Investment Management* (5:4), 2007, pp. 5-54.

Kwon, Y.-K., Moon, B.-R. "A hybrid neurogenetic approach for stock forecasting", *IEEE Transactions on Neural Networks* (18:3), May 2007, pp. 851-864.

Laidi, A. *Currency Trading and Intermarket Analysis: How to Profit from the Shifting Currents in Global Markets*, Wiley, Hoboken, New Jersey, 2009.

Lee, V. C. S., Wong, H. T. "A multivariate neuro-fuzzy system for foreign currency risk management decision making", *Neurocomputing* (70), 2007, pp. 942-951.

Lindemann, A., Dunis, C. L., Lisboa, P. "Level estimation, classification and probability distribution architectures for trading the EUR/USD exchange rate", *Neural Computing and Applications* (14:3), 2005a, pp. 437-470.

Lindemann, A., Dunis, C. L., Lisboa, P. "Probability distributions and leveraged trading strategies: An application of Gaussian mixture models to the Morgan Stanley Technology Index Tracking Fund", *Quantitative Finance* (5:5), 2005b, pp. 459-474.

Marzi, H., Turnbull, M. "Use of neural networks in forecasting financial markets", in *Proceedings of the IEEE International Conference on Granular Computing*, Fremont, November 2007, pp. 516-521.

McNelis, P. D. *Neural Networks in Finance: Gaining Predictive Edge in the Market*, Academic Press, Burlington, 2005.

Mettenheim, H.-J. von *Advanced Neural Networks: Finance, Forecast, and other Applications*, PhD thesis, Leibniz Universität Hannover, Germany, 2009.

Mettenheim, H.-J. von, Breitner, M. H. "Neural network forecasting with high performance computers", in *Proceedings of the 13th International Workshop on Dynamics and Control: Modeling and Control of Autonomous Decision Support Based Systems*, Wiesensteig, Shaker, Aachen, May 2005.

Mettenheim, H.-J. v., Breitner, M. H. "Distributed neurosimulation", in *Operations Research Proceedings: Selected Papers of the Annual International Conference of the German Operations Research Society*, Bremen, September 2005, Springer, Heidelberg, 2006.

Miazhynskaia, T., Frühwirth-Schnatter, S., Dorffner, G. "Neural network models for conditional distribution under bayesian analysis", *Neural Computation* (20:2), 2008, pp. 504-522.

Nagarajan, V., Wu, Y., Liu, M., Wang, Q.-G. "Forecast studies for financial markets using technical analysis", in *Proceedings of the International Conference on Control and Automation* (1), June 2005, pp. 259-264.

Pires, M., Marwala, T. "American option pricing using Bayesian multi-layer perceptrons and Bayesian support vector machines", in *Proceedings of the 3rd IEEE International Conference on Computational Cybernetics*, Mauritius, April 2005, pp. 219-224.

Powell, W. B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley, Hoboken, New Jersey, 2007.

Samarasinghe, S. *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*, Auerbach Publications, Boca Raton, New York, 2007.

Schaefer, A. M., Udluft, S., Zimmermann, H.-G. "A recurrent control neural network for data efficient reinforcement learning", in *Proceedings of the IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning,* Honolulu, 2007, pp. 151-157.

Setiono, R., Mues, C., Baesens, B. "Risk Management and Regulatory Compliance: A Data Mining Framework based on Neural Network Rule Extraction", in *Proceedings of the Twenty-Seventh International Conference on Information Systems*, Haseman, W. D. (Ed.), Milwaukee, Wisconsin, December 2006, pp. 71-86.

Steil, J. J. "Online stability of backpropagation-decorrelation recurrent learning", *Neurocomputing* (69), 2006, pp. 642-650.

Tsangari, H. "An alternative methodology for combining different forecasting models", *Journal of Applied Statistics* (34:4), 2007, pp. 403-421.

Turban, E., Sharda, R., Delen, D., Aronson, J. E., Liang, T.-P., King, D. *Decision Support and Business Intelligence Systems*, Prentice Hall, New Jersey, 2010

Vanstone, B., Finnie, G. "Enhancing Existing Stockmarket Trading Strategies Using Artificial Neural Networks: A Case Study", *Lecture Notes in Computer Science* (4985), 2008, pp. 478-487.

Vanstone, B., Finnie, G. "An empirical methodology for developing stock-market trading systems using artificial neural networks", *Expert Systems with Applications* (36), 2009, pp. 6668-6680.

Wu, L., Brynjolfsson, E. "The Future of Prediction: How Google Searches Foreshadow Housing Prices and Quantities", in *Proceedings of the Thirtieth International Conference on Information Systems*, Phoenix, Arizona, December 2009, Paper 147.

Yu, L., Wang, S., Huang, W., Lai, K. K. "Are foreign exchange rates predictable? A survey from artificial neural networks perspective.", *Scientific Inquiry* (8:2), 2007a, pp. 207-228.

Yu, L., Wang, S., Lai, K. K, Huang, W. W. "Developing and assessing an intelligent forex rolling forecasting and trading decision-support system for online e-service", *International Journal of Intelligent Systems* (22), 2007b, pp. 475-499.

Zimmermann, H. G. "Advanced Forecasting with Neural Networks", in *Proceedings of the 17th Forecasting Financial Markets Conference: Advances for Exchange Rates, Interest Rates and Asset Management*, Dunis, C., Dempster, M., Breitner, M. H., Rösch, D., Mettenheim, H.-J. von (Eds.), Leibniz Universität Hannover, May 2010.

Zimmermann, H. G., Grothmann, R., Schaefer, A. M., Tietz, C. "Modeling large dynamical systems with dynamical consistent neural networks", in *New Directions in Statistical Signal Processing: From Systems to Brain*, Haykin, S., Principe, J. C., Sejnowski, T. J., McWhirter, J. (Eds.), MIT Press, Cambridge, 2006, pp. 214-255.