Financial Volatility Trading Using Recurrent Neural Networks

Peter Tiňo, Christian Schittenkopf, and Georg Dorffner

***Abstract*—We simulate daily trading of straddles on the financial indexes DAX and FTSE 100. The straddles are traded based on predictions of daily volatility differences in the underlying indexes. The main predictive models studied in this paper are recurrent neural networks (RNNs). In the past, applications of RNNs in the financial domain were often studied in isolation. We argue against such a practice by showing that, due to the special character of daily financial time-series, it is difficult to make full use of RNN representational power. Recurrent networks either tend to overestimate the noisy data, or behave like finite-memory sources with a relatively shallow memory. In fact, they can hardly beat (rather simple) classical fixed-order Markov models. To overcome the inherent nonstationarity in the data, we use a special technique that combines "sophisticated" models fitted on a larger data set, with a fixed set of simple-minded symbolic predictors using only recent inputs, thereby avoiding older (and potentially misleading) data. Finally, we compare our predictors with the GARCH family of econometric models designed to capture time-dependent volatility structure in financial returns. GARCH models have been used in the past to trade volatility. Experimental results show that while GARCH models are not able to generate any significantly positive profit, by careful use of recurrent networks or Markov models, the market makers can generate a statistically significant excess profit. However, on this type of problems, there is no reason to prefer RNNs over much more simple and straightforward Markov models. We argue that any report containing RNN results on financial tasks should be accompanied by results achieved by simple finite-memory sources combined with simple techniques to fight nonstationarity in the data.**

***Index Terms*—Financial indexes, Markov models, options, prediction suffix trees, recurrent neural networks, straddle, volatility.**

I. Introduction

It has been shown in the past that quantizing real-valued financial time-series into symbolic streams and subsequent use of predictive models on such sequences can be of great benefit in many financial tasks [1]–[9]. This is predominantly due to the inherently noisy and nonstationary nature of financial data. Careful quantization can reduce the noise component in the data while preserving the underlying predictable patterns in the stochastic process.

However, the question of the number and position of quantization intervals has been largely dealt with in an *ad hoc* manner. For example, Papageorgiou [3] quantized daily returns of exchange rates of five major currencies into nine intervals (symbols). No clue was provided as to why nine quantization intervals were used and how the particular cut-values defining the intervals were chosen. Giles *et al.* [1], [2] considered the same set of exchange rates. The returns were quantized using one-dimensional self-organizing feature map [10] without a direct control over the cut-values. Up to seven quantization intervals were considered.

We have introduced a data-driven parametric scheme for quantizing real-valued time-series in [7] and [11]. The scheme is motivated by Bühlmann's treatment of extreme events in financial data [6] and allows for a principled handling of quantization intervals.

Surprisingly, *realistic trading applications* of the quantization heuristic are still missing. As a first step in this direction, we recently performed a large comparative study of various finite-memory model classes used to predict volatility differences in order to trade straddles on the DAX and FTSE 100 [11]. The straddles were traded on a daily basis based on predictions of daily (implied) volatility differences in the underlying indexes. Continuous models operating on the original real-valued sequences of volatility differences, as well as symbolic models trained and tested on their quantized counterparts were considered. Two key observations were made.

1) Predictive models operating on real-valued time-series performed worse than those operating on the quantized sequences. In other words, the quantization technique significantly improves the overall profit.
2) Quantization into just two symbols representing the *sign* of elements in the time-series gave the best results. Nonstationarity in the data allowed for only a relatively small rolling-window size (two and half years) and so quantization schemes using more than two intervals lead to poorer generalization.

Therefore, in this study, we only work with *binary* symbolic streams $s_1 s_2 \ldots, s_t \in \{0, 1\}$, $t = 1, 2, \ldots$, obtained by quantizing the original, real-valued time-series of daily volatility differences, $\delta_1 \delta_2 \ldots, \delta_t \in \Re$, $t = 1, 2, \ldots$, using the "signum" function

$$s_t = \text{sign}(\delta_t) = \begin{cases} 0, & \text{if } \delta_t < 0 \text{ (volatility decreased)} \\ 1, & \text{otherwise (volatility increased).} \end{cases} \quad (1)$$

While our previous study [11] employs only finite-memory models (autoregressive neural networks, Markov models

Manuscript received August 2, 2000; revised February 7, 2001. This work was supported by the Austrian Science Fund (FWF) under Research Project "Adaptive Information Systems and Modeling in Economics and Management Science" (SFB 010). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Transport.

P. Tiňo is with the Neural Computing Research Group, Aston University, Birmingham B4 7ET, U. K. He is also with the Slovak Technical University, Bratislava, Slovakia.

C. Schittenkopf and G. Dorffner are with the Austrian Research Institute for Artificial Intelligence, A-1010 Vienna, Austria.

Publisher Item Identifier S 1045-9227(01)05011-1.

(MMs), prediction suffix trees, ...), the main predictive models studied in this paper are recurrent neural networks (RNNs), in particular Elman networks [12]. Recurrent networks have (theoretically) an *unbounded memory* and been shown to be able to develop sophisticated state-space representations of dynamics underlying complex symbolic time series [13], [14]. We study whether such a representational power can be utilized in daily financial trading. Our primary motivation is that RNN applications in the financial domain have often been studied in isolation, without a proper comparison with more simple and cheaper model classes. Intuitively, RNNs may not be the most attractive candidates for this application area, since nonstationarity of daily financial time-series prevents us from using large samples to train the predictive models. This, combined with a high degree of stochasticity typical of the financial data, can result in a poor generalization when using models endowed with an excessive power.

Furthermore, to overcome the nonstationarity of the data, we design a special technique that combines "sophisticated" models fitted on a larger data set, with a fixed set of simple-minded symbolic predictors using only recent inputs, thereby avoiding older (and potentially misleading) data. Finally, we compare our symbolic models with the GARCH family of econometric models traditionally used to trade volatility.

The paper is organized as follows. In the next section we briefly mention trading of options. We do not assume the reader is familiar with specifics of financial forecasting applications. Therefore, we provide in the Appendix a brief introduction to options and option pricing. Section III-B describes the architecture of recurrent neural networks employed in this study. In Section IV we specify the data used in our experiments and outline the strategy for trading straddles. Section V gives a detailed description of the experiments. The experimental results are presented and discussed in Section VI. The conclusion summarizes the main findings of this study.

II. Trading Options

Option price forecasts are either based on implied volatilities derived from an observed series of option prices, or volatilities computed/modeled from a series of returns of the underlying asset. For example, Noh *et al.* [15] used a GARCH[1] model [16] to predict volatility in returns of an asset and then calculated predictions of option prices based on such GARCH-modeled volatilities. The volatility change forecasts (volatility is going to increase, or decrease) can be interpreted as a buying or selling signal for a straddle.[2] This enables one to implement simple trading strategies to test the efficiency of option markets (e.g., S&P 500 index [15], or German Bund Future Options [17]). If the volatility decreases, we go short (straddle is sold), if it increases, we take a long position (straddle is bought). More details on options and option pricing can be found in the Appendix.

[1]GARCH stands for generalized autoregressive conditional heteroskedasticity.

[2]A straddle is a couple of put (the right to sell) and call (the right to buy) options with the same time to maturity and the same strike price. See the Appendix

Against the optimistic hope that such automatic trading strategies can lead to significant excess profits stands the pessimistic received wisdom (at least among academics) of the *efficient markets hypothesis* [18]. In its simplest form, this hypothesis states that asset prices follow a random walk and so apart from a possible constant expected appreciation (e.g., a risk-free return), the movement of an asset price is completely unpredictable from publicly available information. The ability of technical trading rules to generate an excess profit has been a controversial subject for many years. The evidence against, or for the efficient markets hypothesis still seems inconclusive [19].

In our technical trading of options, we quantize the time series of daily volatility changes into a symbolic sequence characterizing the original real-valued sequence only through binary directional events—*increase* (1), *decrease* (0) [see (1)]. The output of predictive models gives an indication of possible volatility change and serves as a buying/selling signal in simple strategies that trade straddles.

III. Predictive Models

As mentioned in the previous section, our aim is to trade straddles on a daily basis, based on our prediction of the character of the next volatility change (today $\rightarrow$ tomorrow). When we think the volatility will increase (decrease), we buy (sell) the straddle. Predictions about the sign $\{s_t\}$ of the future volatility change $\{\delta_t\}$ (day $(t-1) \rightarrow$ day (t)) are provided by predictive models trained on binary sequences $\{s_i\}$ representing (in the quantized form) the previous history of daily volatility changes $\{\delta_i\}$ (see (1)).

In the following we introduce two classes of predictive models used in our study, namely finite memory MMs and RNNs with (theoretically) an unbounded memory.

A. Markov Model

Consider a training sequence $S = s_1 s_2 \ldots$, over the binary alphabet $\{0,1\}$, i.e., every symbol s_i is from $\{0,1\}$.

The basic assumption behind MMs of a (fixed) order L is that for prediction purposes, we do not need to consider the whole history of previously seen symbols. Instead, only the L last seen symbols are taken into an account.

$$P(s_{t+1}|\ldots s_{t-1}s_t) = P(s_{t+1}|s_{t-L+1}\ldots s_{t-1}s_t). \qquad (2)$$

The conditional prediction probabilities (2) are computed as

$$P(s|w) = \frac{P(ws)}{P(w)}, \qquad s \in \{0,1\}, \quad w \in \{0,1\}^L \qquad (3)$$

where the relative frequencies of the strings ws and w of length $L+1$ and L, respectively, are computed on the training sequence S.

B. Elman Recurrent Neural Network

Elman recurrent network is characterized by delayed feedback connections from each of the hidden nodes to all hidden nodes (see Fig. 1). The architecture has two input, H hidden, and two output nodes.

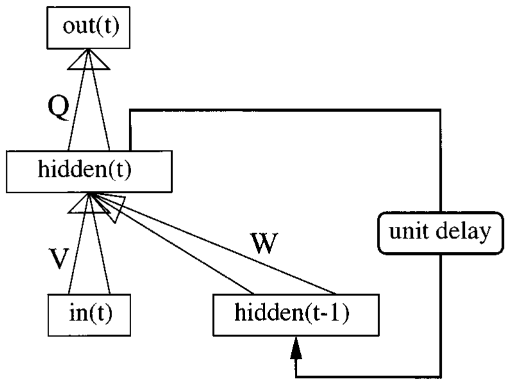

Fig. 1. Elman recurrent neural network.

The binary input symbols 0 and 1 are encoded via a one-hot encoding as two-dimensional vectors $\mathbf{c}(0) = (1, 0)$ and $\mathbf{c}(1) = (0, 1)$, respectively. The network is presented at its input with a binary symbolic stream $s_1 s_2 \ldots, s_t \in \{0, 1\}$, one symbol per time step. At time t, the input symbol s_t is encoded into the input vector $\mathbf{x}(t) = (x_1(t), x_2(t)) = c(s_t)$.

Given the previous activations in hidden nodes $\mathbf{h}(t-1) = (h_1(t-1), h_2(t-1), \ldots, h_H(t-1))$, the hidden nodes are updated as

$$h_i(t) = \sigma\left(\sum_{j=1}^{2} V_{ij} x_j(t) + \sum_{k=1}^{H} W_{ik} h_k(t-1) + T_i^{hid}\right)$$

where V_{ij} and W_{ik} are real valued weights and $T_i^{hid} \in \Re$ are threshold terms. The transfer function σ is the standard logistic sigmoid $\sigma(u) = 1/(1 + e^{-u})$.

The output nodes are then updated as

$$o_i(t) = \sigma\left(\sum_{j=1}^{H} Q_{ij} h_j(t) + T_i^{out}\right)$$

where Q_{ij} and T_i^{out} are weights and thresholds, respectively.

We train the network to predict the next symbol using full real-time recurrent learning [20] with a momentum term. We use sum-of-squared-errors measure[3]

$$SSE = \sum_{t=1}^{N-1} \sum_{i=1}^{2} (o_i(t) - x_i(t+1))^2 \quad (4)$$

as well as the relative entropy error measure [21]

$$RE = -\frac{1}{N-1} \sum_{t=1}^{N-1} [s_{t+1} \log(o_2(t)(1 - o_1(t))) + (1 - s_{t+1}) \log((1 - o_2(t)) o_1(t))]. \quad (5)$$

Slightly abusing the notation, if $s_{t+1} = 0$, the first term in the sum vanishes and the network output which minimizes the contribution to the RE is (1, 0). If $s_{t+1} = 1$, the second term vanishes and the desired network output is given by (0, 1).

Following [2], we control overfiting by linearly reducing learning rate over the training period. Momentum rate is reduced in the same manner. The initial value for learning and momentum rates is 0.25. Training period consists of 1000 epochs (passes through the training set). Weights and thresholds are randomly initiated from the interval [−0.5, 0.5]. At the beginning of the training, a start state $\mathbf{h}(0)$ is randomly generated from the interval $[0.25, 0.75]^H$, where H is the number of hidden units. Then, after each training epoch the network is reset to that state.

Typically, training sets are not balanced. Since in the series of volatility differences there is usually a large positive shock followed by a longer relaxation period of smaller (predominantly) negative volatility changes, the quantized training sequences contain more *decrease* symbols (0) than *increase* ones (1). This phenomenon is actually directly modeled in the GARCH family of econometric models [16].

Lawrence, Tsoi and Giles [2] deal with the unbalanced training set problem by killing-off some of the error signals, so that the proportions of weight updates for the next-symbols 0 and 1 are the same. However, due to nonstationarity in the data we are forced to use relatively small training sets. All training signals are useful and should not be dismissed. Also, compared to predicting volatility differences, the unbalanced training set problem is much less pronounced when predicting returns as in [2]. We account for the lack of balance in the training sets by using different learning rates for different next-symbols desired at the network output. More specifically, let U and D represent the numbers of symbols 1 and 0, respectively, in the training sequence. Then, prior to updating the network parameters, the base learning and momentum rates (that are linearly reduced as the training proceeds) are multiplied by $D/(U+D)$, if the next symbol is *increase* (1), or $U/(U+D)$, otherwise. For related techniques, see [22].

IV. Data, Volatility Measures, and Trading Strategy

We encourage the readers not familiar with option contracts and option pricing to consult the Appendix prior to reading this section.

Two large data sets are analyzed in this study. First, we describe the set of daily options data on the DAX and give some details on the set of intra-day option contracts on the FTSE 100. Then, three measures of volatility are defined which are applied in the trading strategy introduced in the final part of this section.

A. DAX Data Set

The data set is a series of daily closing values of the German stock index DAX together with a series of daily closing prices of call and put options on the DAX with different maturities and exercise prices. In particular, the first in-the-money and the first out-of-the money call and put option maturing next month are available. The at-the-money point is assumed to be the value of the DAX at that time.[4]

Straddle prices are obtained by adding call and put prices. The series starts on August 22, 1991, and end on June 8, 1998, which corresponds to a period of 1700 trading days.

[3]Assuming the length of the training sequence is N. Also note that the code $\mathbf{x}(t+1)$ of the next symbol s_{t+1} forms the desired output at time t.

[4]It would be probably more appropriate to take the value of the futures contract with the same maturity as the at-the-money point. This is left for future studies.

The time series of daily returns r_t, obtained from daily index values ν_t via

$$r_t = \log \nu_t - \log \nu_{t-1} \tag{6}$$

is depicted in the upper graph in Fig. 2.

B. FTSE 100 Option Contracts

This sample comprises transactions data of FTSE 100 option contracts traded at the London International Financial Futures Exchange (LIFFE). Intra-day bid-ask prices of American-style options on the FTSE 100 between May 29, 1991, and December 29, 1995, are available. This time period corresponds to 1161 trading days. The option prices are recorded synchronously with the FTSE 100 and time-stamped to the nearest second. Since our trading strategy is set up on a daily basis, we must fix a reference point in time on each trading day. This reference point is 3 pm on normal trading days and 12 pm on days where the stock exchange closes earlier. The first quote of the FTSE 100 after the reference point is extracted for each day and used to calculate the historical volatility measure (see Section IV-C). In addition, the first quotes of call and put options maturing the next month with the same strike price as close as possible to the value of the FTSE 100 at that time are extracted for the actual trading day (and also for the next trading day). For these options, which are roughly at-the-money, the average of bid-ask quotes is calculated as an approximation of a reasonable option price. Then the prices of call and put options are added to obtain the straddle prices. The time series of daily returns (between the reference points) is depicted in the lower graph in Fig. 2.

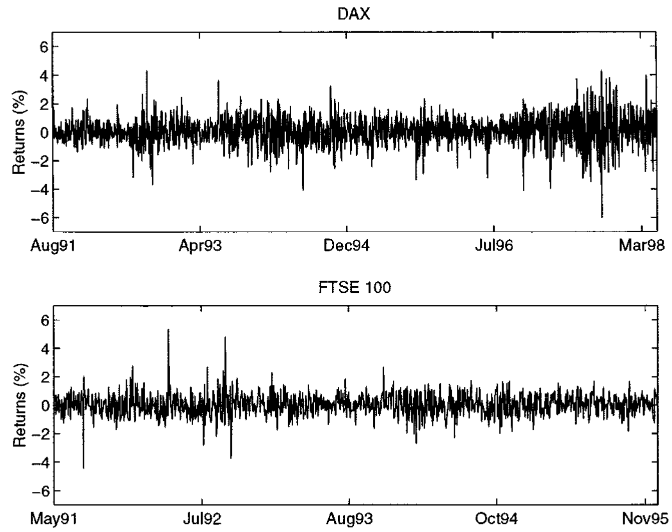

Fig. 2. The time series of daily returns of the DAX (upper graph) and of the FTSE 100 (lower graph).

C. Volatility Measures

There are basically three notions of volatility in literature: 1) *historical volatility*; 2) *implied volatility*; and 3) *model-based volatility* determined, e.g., by models from the GARCH family.

1) The historical measure is a popular approach where exponentially declining weights are given to past volatilities, approximated by squared returns. That is, the historical volatility is an exponentially weighted average of the past squared returns (**EWA**-measure). For the DAX, the returns are calculated from the closing values of the index whereas the returns of the FTSE 100 are calculated from the value of the index at the reference point in time. Denoting the index values ν_t and the continuously compounded returns $r_t = \log(\nu_t/\nu_{t-1})$, the historical measure V_t is defined as

$$V_t = (1-\alpha)\sum_{\tau \le t} \alpha^{t-\tau} r_\tau^2. \tag{7}$$

The weighting factor $\alpha \in (0,1)$ determines the impact of past returns on the actual volatility: The larger α, the larger the impact and the longer the "memory." Our choice was $\alpha = 0.9$. The measure defined in (7) can also be specified in a recursive fashion

$$V_t = \alpha V_{t-1} + (1-\alpha) r_t^2 \tag{8}$$

with $V_0 = 0$. The historical measure V_t is similar to the basic volatility measure applied in RiskMetrics and it is well suited for the purpose of trading options on financial indexes [23].

2) The implied volatility measure (**IV**-measure) is estimated from the extracted options' volatilities implied by the Black–Scholes model [24]. For the DAX, the average value of the implied volatilities of the first in-the-money and the first out-of-the money call and put option is calculated. For the FTSE 100, the average value of the implied volatilities of the call and put option closest to the at-the-money point is used. In other words, the average value of four (two) implied volatilities is calculated as the IV-measure V_t for the DAX (FTSE 100) every day.

3) The GARCH models [16] try to reconstruct volatilities V_t from the series of returns $\{r_t\}$ (6). Basic to these models is the notion that the returns $\{r_t\}$ can be decomposed into a predictable component μ_t and an unpredictable component e_t, which is assumed to be a zero mean Gaussian (or t-distributed) noise of finite variance σ_t^2: $r_t = \mu_t + e_t$. The models are thus characterized by time-varying conditional variances σ_t^2 and are therefore well suited to explain volatility clusters typically present in the series of returns. The conditional mean is modeled as a linear function of the previous value: $\mu_t = a r_{t-1} + b$. For the GARCH(1, 1) model, the conditional variance σ_t^2 is given by

$$\sigma_t^2 = a_0 + a_1 e_{t-1}^2 + a_2 \sigma_{t-1}^2. \tag{9}$$

The GARCH model estimates the volatility V_t at time t as the conditional variance σ_t^2.

Given a training set of historical returns $\{r_t\}$, the free parameters of the GARCH model are estimated by the maximum likelihood [16]. Note that, unlike MMs and recurrent neural networks that are trained on quantized volatility differences, the GARCH models that *define a volatility measure* are trained on a series of returns of the underlying index.

D. Trading Strategy

The basic trading strategy is to buy (sell) at-the-money straddles whenever volatility is predicted to increase (decrease). Since at-the-money straddles are approximately delta-neutral, there is no need to delta-hedge, and the strategy is thus a pure volatility trading strategy. In detail, every day, a constant amount of money is invested to buy (sell) the straddles, and on the next day, the straddles are sold (bought). If a model is uncertain about the sign of the volatility change (equal evidence for increase and decrease), the invested money is put into the bank at an annualized interest rate of 4%.

The choice of a fixed but otherwise arbitrary investment is intended to facilitate the interpretation of results with respect to transactions costs. Finally, only straddles maturing the following month are bought or sold, which avoids the influence of strong price movements toward the end of the contracts.

V. Experimental Setup

Given the series $\{V_t\}$ of estimated daily volatilities, we create a new series $\{\delta_t\}$ of daily volatility differences $\delta_t = V_t - V_{t-1}$. On the basis of the series $\{\delta_t\}$ we construct, select, and test predictive models used in our trading strategy.

To deal with nonstationarity in $\{\delta_t\}$, we use the sliding window technique. At each position, the sliding window of length 630 contains the training set (the first 500 points—roughly two years), followed by a validation set (125 points—roughly six months), and a test set (5 points—one week). Predictive models are estimated on the training set. Within each model class, the best performing candidate with respect to profit is selected on the validation set, and finally, the profit of the selected model is determined on the test set. Then the time window is shifted by five days, predictive models are reestimated, etc.

Predictive models in this study operate on binary sequences $\{s_t\}$ obtained by quantizing daily volatility differences $\{\delta_t\}$

$$s_t = \begin{cases} 0\ (\text{down}), & \text{if } \delta_t < 0 \\ 1\ (\text{up}), & \text{otherwise.} \end{cases} \tag{10}$$

A. Predictive Model Classes

To make predictions about the nature of the volatility move for the next day that will be used in our strategy for trading the straddles, we use the following predictive model classes:

- **MM-5**—MMs of order up to five (one week). An MM of order n predicts the next symbol based on the previous n symbols. The conditional next-symbol probabilities are determined from the training set. This model class includes MMs of order 0, 1, 2, ..., 5. The model order to be used on the test set is selected on the validation set, i.e., for predicting on the test set we use the MM order that yields the highest validation set profit.
- **MM-10**—MMs up to order ten (two weeks). The class includes MMs of order 0, 1, 2, ..., 10. As in the previous class, the model order is determined on the validation set.
- **RNN-SSE**—Elman recurrent neural networks (RNN) (see Section III-B trained via real-time recurrent learning to minimize the sum-of-squared-errors functional SSE (4) (linearly decreased learning and momentum rates). We train RNN's with two, four, and six hidden nodes. The optimal size of the network is determined on the validation set, i.e., for predicting on the test set we select the number of hidden units that leads to the highest profit on the validation set.
- **RNN-RE**—The same as RNN-SSE, but trained to minimize the relative entropy error measure RE (5).
- **RNN-SSE-LR**—RNN-SSE with the next-symbol conditional learning and momentum rates to deal with the problem of unbalanced training sets. See Section III-B.
- **RNN-RE-LR**—RNN-RE with the next-symbol conditional learning and momentum rates.
- **GARCH**—GARCH(1, 1) models with either a Gaussian, or a t-distribution (see Section IV-C). This model class was included as a benchmark, since it has been used in similar studies (e.g., [15] and [17]). In contrast to the previous model classes, the GARCH models are not trained on the quantized, precomputed series of differences of historical volatilities. Instead, the GARCH models try to reconstruct the volatilities from the series of returns $\{r_t\}$ (6). At each sliding window position, the GARCH(1, 1) models with a Gaussian and a t-distribution are fit, in the maximum likelihood framework, to the series of returns corresponding to the training set. The optimal form of the noise distribution (Gaussian versus t-distribution) is then determined on the validation set.

Given the historical data up to the current day,[5] the models predict the sign of the next volatility change as follows:

- **MMs** predict *decrease* (*increase*) when the model predicts symbol 0 (1) with higher probability than symbol 1 (0); if the two probabilities coincide, MMs output *don't know* (equal evidence for *decrease* and *increase*).
- **Recurrent networks** predict *decrease* (*increase*), when the activation of the output unit corresponding to symbol 0 (1) is greater than that of the other unit. If the activations are the same, RNNs declare *don't know*.

When a RNN is used to predict on the validation set (that follows the training set), the network is first initiated with the start state $\mathbf{h}(\mathbf{0})$ (see Section III-B) and subsequently driven by a concatenation of the training and validation sequences. The outputs are registered only after the validation sequence is reached. Similarly, when predicting on the test set, the RNN is initiated with $\mathbf{h}(\mathbf{0})$, driven by a concatenation of the training, validation and test sequences. We record the outputs only on the test set.

Since prior to training, the network weights are randomly initiated, we stabilize the RNN predictions by training a committee of ten networks (for each hidden layer size—2, 4, 6). Given an input, each member of the committee makes its prediction (*decrease*, *increase*, or *don't know*). The overall output of the committee is then based on the majority vote. Hence, each member of a

[5]Sequence of the past quantized (historical, or implied) volatility differences (in case of MMs and recurrent networks), or sequence of the past returns of the underlying index (in case of the GARCH model).

RNN model class is a committee of ten networks sharing the same architecture, the same size and the same training details.

- The **GARCH** model makes a prediction about the direction of the volatility change based on the difference between the consecutive GARCH-volatilities, i.e., based on the sign of $\sigma_t^2 - \sigma_{t-1}^2$ (9).

B. Taking the Nonstationarity Seriously

At each position of the time-window, the models are fitted to data that start 2 1/2 years before their actual use for volatility predictions on the test set. This can be dangerous, especially when sudden large "stationarity breaks" are present in the data (see [25]). On the other hand, using a shorter sliding window can lead to undesirable over-fitting effects.

Our idea, tested in [11], is to use yet another model class, **SIMPLE**, that is a small collection of simple, *fixed* predictors *requiring no training on the training set*. A suitable candidate model to be applied on the test set is selected on the validation set. The class **SIMPLE** avoids using the old data in the sliding window. Of course, the price to pay is the fixed nature of the predictors in **SIMPLE**. However, in financial prediction tasks, simple, short memory models often outperform more sophisticated ones. The class **SIMPLE** is a collection of four simple-minded predictors operating on the series of quantized volatility differences: *always predict 0 (decrease)*, *always predict 1 (increase)*, *copy the last symbol* and *revert the last symbol* (i.e., predict the other symbol). Without having to deal with the training set, we directly apply each of the four simple predictors on the validation set and select the one that achieved the highest profit. The selected predictor is then used on the test set as the representative of the model class **SIMPLE**.

We hope to build a more powerful prediction strategy by *combing* the model classes $\mathcal{M}$ summarized in Section V-A with the class **SIMPLE**. At each position of the time-window and for each model class $\mathcal{M}$, we switch between the winner candidate of the class $\mathcal{M}$ and the winner from **SIMPLE** based on the profit achieved on the validation set. In other words, when combining the model classes $\mathcal{M}$ and **SIMPLE**, at each position of the time-window, we compare the validation set profits gained by the winner candidates of $\mathcal{M}$ and **SIMPLE**, and select the one that achieved the higher profit. We denote such compound model classes by C-$\mathcal{M}$.

C. Significance of Profits

The most natural performance measure for any model with respect to a trading strategy is the average profit per day calculated from the sequence of out-of-sample (test set) profits. A standard approach to deal with variation (uncertainty) in the profits is to apply statistical tests. In related works [15], [26], [27], a t-test is used to test whether profits are significantly positive or not. However, the t-test relies on two crucial assumptions:

1) *Profits are assumed to be independent*. Although the test sets are nonoverlapping, some dependencies might be introduced because the training sets and the validation sets are overlapping. Since we did not find significant correlations in the series of daily profits, the assumption of independent profits is likely to be correct.

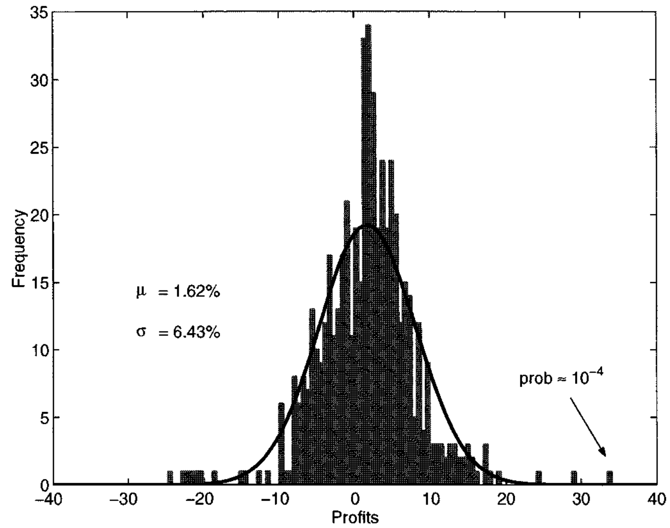

Fig. 3. Histogram of the profits of the simple model (in percent) for the implied volatility measure of the FTSE 100 together with the (appropriately scaled) normal density with the same mean and the same standard deviation.

2) *Profits are assumed to be normally distributed*. Fig. 3 shows a histogram of profits of the **SIMPLE** model class for the implied volatility measure of the FTSE 100, together with the (appropriately scaled) normal density with the same mean and the same standard deviation as the sample of profits. Obviously, the distribution of profits is far from a normal distribution.[6] In particular, the distribution has fat tails which means that large positive and negative profits are observed much more frequently than predicted by the normal distribution. This assumption is thus hardly valid and a t-test should not be applied. Similar histograms are also obtained for the other models.

Since t-tests are not directly applicable, one could think of using a nonparametric test, such as the Wilcoxon test [28]. However, the Wilcoxon test calls for a transformation of the series of profits into a series of ranks meaning that the information about the absolute size of the profits is lost: For instance, it makes no difference if the largest profit is 20% or 200%. This property is rather undesirable for the analysis of profits of a trading strategy and therefore we decided not to use the Wilcoxon test either.

We tested for significance of the profits obtained from the trading strategy by applying the following procedure: The profits on the individual test sets are concatenated to form a large series of out-of-sample profits. This series is divided into distinct blocks of length 40, which corresponds to a period of roughly two months. Due to the central limit theorem, which holds under rather general conditions, the sum of profits, and therefore also the average block profit (i.e., the average profit over a block of 40 trading days) can be assumed to be normally distributed.[7] Hence we can subject the series of average block

[6]A Jarque-Bera test rejects the null hypothesis of normally distributed profits at any reasonable significance level.

[7]Although the central limit theorem provides an asymptotic result, the normal distribution is usually approached very fast in practice as the number of independent random variables (profits) increases. In general, 30 to 40 random variables are assumed to be sufficient.

profits to t-tests.[8] The number of out-of-sample profits for the DAX and the FTSE 100 data set is 1060 and 525, respectively. Consequently, the number of average block profits for the DAX and the FTSE 100 data set is 26 and 13, respectively.[9]

D. Summary of the Experimental Setup

The overall picture of our experimental setup is shown in Fig. 4. A sliding window rolling through the series of (quantized) daily volatility differences contains a training, a validation and a test sequence. At each sliding window position, and for each model class $\mathcal{M}$, the models from $\mathcal{M}$ are trained on the training sequence. Then the models are used to predict the signs of volatility differences on the validation set. These predictions are used as trading signals in our trading strategy thereby producing a series of validation set profits. Based on the average validation set profit, we select the candidate of the model class $\mathcal{M}$ to be used on the test set. The selected candidate is then used to predict the signs of volatility differences on the test set, which are in turn plugged into the trading strategy to produce the test set profits. Also, at each position of the sliding window and for each model class $\mathcal{M}$, we save the average profit achieved on the validation set by the selected candidate from $\mathcal{M}$. The validation set profits are used to combine simple strategies from **SIMPLE** with the more complex model classes, as discussed in Section V-B.

After the sliding window reaches its final position, for each model class, we concatenate all the daily test set profits into a single profit series. This series is then partitioned into several nonoverlapping blocks of length 40. For each block we compute the average profit per day. Finally, we report statistics computed on thus obtained series of average block-profits.

VI. Results and Discussion

Tables I and II contain results obtained for the base model classes $\mathcal{M}$ studied in this paper (see Section V-A), as well as for the compound models (denoted by C-$\mathcal{M}$) that combine the base models with the class **SIMPLE**, as discussed in Section V-B.

To test usefulness of a particular notion of volatility for automatic trading strategies, we report profits for the hypothetic *always correct predictor*, (**ACP**), i.e., the predictor that always *knows* in advance the sign of the next volatility difference.

We report the following statistics of the EWA and the IV volatility notions (see Section IV-C): The first and the second columns give the mean and the standard deviation of daily profits.[10] The third column gives the highest transactions costs (HTC) in percent that may be subtracted from the profits on a daily basis so that the average block profit is still significantly positive (under a t-test with a 5% significance level). The reported transactions costs are assumed to be total daily transactions costs for buying and selling a straddle portfolio. For transactions costs below 0.01$% the table entry is "–."

[8]After calculating the average block profits for the profits depicted in Fig. 3, a Jarque-Bera test does not reject the null hypothesis of a normal distribution at any reasonable significance level.

[9]The last block contains more than 40 profits for both data sets.

[10]The actually invested amount of money is not specified in the experiments since profits are calculated and reported in percentages of the investment.

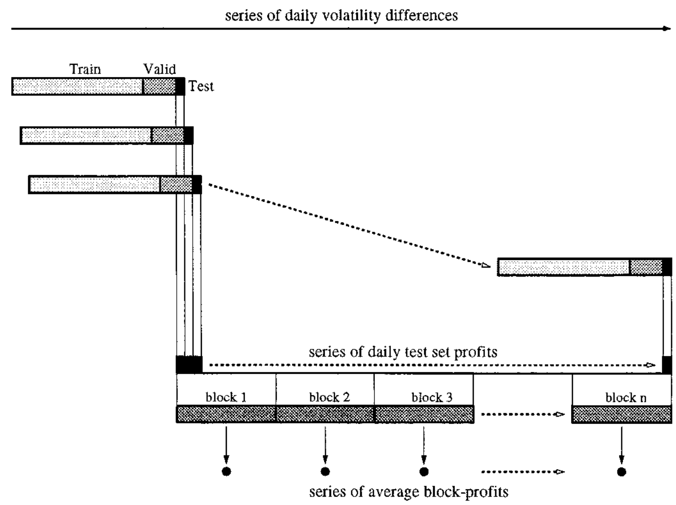

Fig. 4. An illustration of the experimental setup used in the DAX and FTSE 100 experiments.

TABLE I
TRADING PERFORMANCE OF THE BASE AND COMPOUND MODELS ON THE DAX

Model	EWA-Measure			IV-Measure		
	Mean	Std.	HTC	Mean	Std.	HTC
ACP	1.82	4.25	1.61	1.32	4.40	1.03
SIMPLE	0.41	4.42	0.16	0.38	4.40	0.17
MM-5	0.25	4.32	–	0.26	4.30	0.05
MM-10	0.33	3.93	0.13	0.23	4.20	–
RNN-SSE	0.20	4.27	–	$-0.11^{(-)}$	4.33	–
RNN-RE	0.15	4.28	–	-0.02	4.32	–
RNN-SEE-LR	0.13	4.50	–	$-0.11^{(-)}$	4.33	–
RNN-RE-LR	$-0.01^{(-)}$	4.45	–	$-0.09^{(-)}$	4.33	–
C-MM-5	0.44	4.42	0.21	0.38	4.40	0.19
C-MM-10	0.49	4.39	0.29	0.39	4.41	0.21
C-RNN-SSE	0.41	4.42	0.16	0.38	4.40	0.17
C-RNN-RE	0.41	4.42	0.16	0.32	4.41	0.10
C-RNN-SEE-LR	0.41	4.42	0.16	0.38	4.40	0.17
C-RNN-RE-LR	0.42	4.42	0.18	0.40	4.41	0.19

TABLE II
TRADING PERFORMANCE OF THE BASE AND COMPOUND MODELS ON THE FTSE 100

Model	EWA-Measure			IV-Measure		
	Mean	Std.	HTC	Mean	Std.	HTC
ACP	1.79	6.38	1.23	2.70	6.06	2.13
SIMPLE	0.30	6.62	–	1.62	6.43	1.12
MM-5	0.34	6.62	–	1.48	6.46	1.10
MM-10	0.45	6.28	0.01	1.43	6.47	1.02
RNN-SSE	0.39	6.62	–	1.52	6.45	1.01
RNN-RE	0.31	6.62	–	0.72	6.59	0.20
RNN-SEE-LR	-0.42	6.61	–	1.61	6.43	1.10
RNN-RE-LR	-0.12	6.63	–	1.44	6.47	0.95
C-MM-5	0.37	6.61	–	1.48	6.46	1.10
C-MM-10	0.55	6.44	–	1.43	6.47	1.02
C-RNN-SSE	0.30	6.62	–	1.64	6.42	1.15
C-RNN-RE	0.26	6.62	–	1.32	6.50	0.83
C-RNN-SEE-LR	0.31	6.62	–	1.62	6.43	1.12
C-RNN-RE-LR	0.32	6.62	–	1.59	6.44	1.06

We critically assess usefulness of RNNs for trading straddles by computing significance results obtained by running paired t-tests on the average block-profits achieved by RNNs and **SIMPLE**: (−) and (+) mean that the RNN model class performance is significantly worse and better, respectively, than that of **SIMPLE**.

Profits achieved using the GARCH notion of volatility (as in [15], [17]) are given in Table III.

Our approach to reporting transactions costs takes into account the fact that market participants can be subject to different levels of transactions costs. For instance, according to information provided by the German Futures and Options Exchange (Deutsche Terminbörse), normal transactions costs (for traders) are about 0.5% per straddle. However, market makers only pay about 0.1% and investors who are not members of the exchange face transactions costs of roughly 1% [27]. In addition, investors who are trading on a daily basis (as in our setup) and who are willing to invest potentially large amounts of money usually negotiate special discount rates for transactions costs at the stock exchange.

We stress that our assessment of profits is rather pessimistic, since the straddle portfolio is bought and sold every day regardless, of future volatility changes. For instance, if the portfolio was purchased yesterday due to a predicted increase in volatility and the volatility is again predicted to increase today, it is more profitable to simply keep the portfolio than to sell it and buy it again afterwards. This refinement of the trading strategy would thus reduce transactions costs and thereby raise profits.

Based on the experimental results shown in Tables I–III, the following main observations can be made.

The profits gained by the "always correct predictor" (**ACP**) show that, theoretically, predicting daily volatility changes provides a good basis for automatic trading strategies buying/selling straddles: For both EWA and IV volatility measures and both data sets, average daily profits are significantly positive even after substantial transactions costs of more than 1%. In theory, considerable profits can be realized by the implemented trading strategy.

The performance of the GARCH model is disappointing. The average profits are close to zero which can be, on average, obtained by applying the random strategy where a coin is flipped every day to decide whether straddles are bought or sold.

For the IV-measure of volatility, the performance of the models on the FTSE 100 data set is substantially different from the performance on the DAX data set. For the DAX data set, the results are similar to the ones for the EWA-measure. This is in total contrast to the FTSE 100 data set where the trading strategy is much more profitable for all the models. One possible explanation for this observation is that the DAX data set consists of closing prices and closing values. As repeatedly mentioned in literature, the closing prices of options and the corresponding closing values of the underlying index are potentially recorded asynchronously, which may substantially blur the estimates of the implied volatility. Therefore, the FTSE 100 data set which consists of intra-day option prices recorded at the same time as the underlying index provides a more realistic setup.

The best performance is achieved by the compound models (Section V-B) where transactions costs of up to 1.15% yield significantly positive profits.

TABLE III
TRADING PERFORMANCE OF THE GARCH MODELS

	DAX			FTSE 100		
	Mean	Std.	HTC	Mean	Std.	HTC
GARCH	0.10	4.38	–	0.00	6.63	–

MMs with (potentially) deeper memory tend do slightly outperform MMs with a more shallow memory when EWA-measure of volatility is used. The reverse holds for the IV-measure of volatility. Intuitively, the EWA volatilities contain a deeper memory component than IV volatilities that are driven by the current option and underlying asset prices.

Recurrent neural networks never perform much better than more simple and easier-to-fit MMs. The success of the class **SIMPLE** warns us that, in the financial domain, we should always include simplistic models and trading rules, when reporting results of potentially powerful models, such as recurrent neural networks. On the other hand, as the DAX experiment shows, probably the most reasonable approach to financial forecasting is to construct compound models combining robustness of simple and straightforward methods to nonstationarity with benefits of more powerful models, such as deeper memory MMs.

When analyzing state-space representations developed in RNNs we generally found two types of dynamics:

1) Dynamics inside a RNN was driven by attractive fixed points, each input symbol coded by a different fixed point. When such a network is driven with a symbolic input stream, complicated sets of points in the recurrent layer can arise. Such sets can be be compactly described using (generalized)[11] Moran-like constructions [29], [30]. Briefly, the Moran-like geometric constructions iteratively construct limit sets using a collection of basic sets that may have a very complicated geometry. The basic sets have symbolic addresses and are increasingly refined (with increasing address length) according to a given symbolic dynamical system. As the construction proceeds, the diameter of the basic sets diminishes to zero. We rigorously analyzed an example of such fixed-point recurrent networks in [31], and in [8] and [9] we showed that such networks behave like finite-memory sources, closely resembling variable memory length MMs [32]. In these cases, RNNs and MMs achieved similar profits.
2) Dynamics of RNNs were rather complicated and evolved along aperiodic, chaotic, or long-period attractors. In such cases, RNNs typically over-fitted the training set and achieved poor results on the test set.

VII. CONCLUSION

As typical in financial prediction tasks, a simple model, which in our case selects one out of four simple rules to predict an increase or decrease of volatility, produces profits which are not easily improved by more sophisticated approaches. A

[11]Moran constructions assume nonoverlapping basic sets. Here we allow the basic sets to overlap.

careful combination of these simple rules with high-order MMs produces the best results. This may indicate that two memory regimes characterized by shallow and deep memory dominate the studied series of volatility changes.

We analyzed two markets: the German market represented by the DAX and the British market represented by the FTSE 100 index. For both EWA and IV volatility measures on the DAX, economically meaningful predictions resulting in abnormal profits are only obtained for very low transactions costs. The market of DAX options thus tends to be informationally efficient. One source of uncertainty about the rigidity of this statement is the fact that only closing prices were available which could blur the implied volatility estimates. This point is supported by the results for the intra-day FTSE 100 options data. For the IV-measure, all models generate abnormal profits after substantial transactions costs. In other words, the market of FTSE 100 index options does not seem to be fully efficient.

Recurrent neural networks never perform much better than more simple and easier-to-fit MMs. In the financial domain, we should always include simplistic models like **SIMPLE** and low-order MMs, when reporting results of potentially powerful models, such as recurrent neural networks. Recurrent networks tend to either overfit the training sequence by developing unnecessarily complicated dynamical regimes, or behave like finite-memory sources.

In a related study [11], we performed similar experiments (using implied volatility) with a large set of *finite-memory* models, including auto-regressive feedforward networks. Such networks did never outperform MMs.

The most promising approach to financial forecasting is constructing compound models that combine robustness of simple and straightforward methods for dealing with nonstationarity in the data with benefits of more powerful (but potentially hazardous) predictors.

APPENDIX
OPTION PRICING

A holder of a European *call option* on an underlying asset S (e.g., a stock) has the right, but not the obligation, to *buy* the *underlying* (asset) at a specified price K (the *strike price*) at a specified time T in the future (at maturity). The strike price and the maturity of the option are fixed when the option contract is signed (at time $t \leq T$). If the price S_T of the underlying at time T is larger than the strike price, the option holder will exercise his right to buy the underlying, thereby realizing a profit of $S_T - K$. If $S_T < K$, his option is worthless. Therefore, the option payoff at time T is given by $\max\{S_T - K, 0\}$. The fair price of an option depends on the assumptions about the stochastic process driving the price S_t, $t \leq T$. Black and Scholes [24] pioneered the field of option pricing by deriving a closed-form expression for the fair price $\mathcal{C}$ of a European call option (assuming that the price process is a geometric Brownian motion with drift μ and volatility σ). Although real market dynamics are much more complicated, their famous formula

$$\mathcal{C} = S_t N(d_1) - K e^{-r(T-t)} N(d_1 - \sigma\sqrt{T-t}) \qquad (11)$$

$$d_1 = \frac{\ln \frac{S_t}{K} + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \qquad (12)$$

$$N(d) = \int_{-\infty}^{d} \frac{1}{\sqrt{2\pi}} e^{-(x^2/2)}\, dx \qquad (13)$$

where r denotes the risk-less interest rate, is still applied by many traders at financial markets around the world. All variables in the Black-Scholes formula are explicitly known except for the volatility σ which must be estimated from market data.

A European *put option* gives the option holder the right to *sell* the underlying asset at maturity, thereby yielding a profit $\max\{K - S_T, 0\}$. The option is thus worthless, if $S_T > K$. In the Black-Scholes world, the fair price $\mathcal{P}$ is given by

$$\mathcal{P} = \mathcal{C} + K e^{-r(T-t)} - S_t \qquad (14)$$

where $\mathcal{C}$ is the price of the call option with the same strike price K and the same maturity T.

Depending on the value S_t of the underlying and the strike price K, a *call* option is called *in-the-money* and *out-of-the-money*, if $S_t > K$ and $S_t < K$, respectively. A *put* option is called *in-the-money* and *out-of-the-money*, if $S_t < K$ and $S_t > K$, respectively. An option is called *at-the-money*, if $S_t = K$.

The combination of one call option and one put option with the same K and T is called a *straddle* on the underlying, and its price is given by $\mathcal{S} = \mathcal{C} + \mathcal{P}$. If a straddle is *at-the-money*, which roughly means that $S_t = K$, it is *delta-neutral*, i.e., small price changes ΔS_t of the underlying have a negligible effect on the straddle price $\mathcal{S}$. In other words, the owner of the straddle is *not* exposed to market risk (with respect to price fluctuations). However, the straddle price does depend on changes in the volatility σ of the underlying (the larger σ, the larger $\mathcal{S}$). Indeed, the sensitivity of straddle prices with respect to volatility is maximal for at-the-money straddles. Therefore, buying (selling) at-the-money straddles if volatility is expected to increase (decrease), is a *volatility trading strategy* which is mainly independent of the direction of price changes (see [35] for more details).

One of the popular approaches to estimating the (inherently unobservable) volatility from historical data is to keep track of the prices of options on the underlying that appeared in the past and then invert an option pricing model [such as the Black-Scholes model given by (11)–(13)] to calculate the volatility parameter σ. Volatility estimated in this manner is known as *implied volatility*. The volatility measures used in this study are the implied volatilities estimated by inversion of the Black-Scholes model. In the Black-Scholes world, volatility is assumed to be constant and therefore one should obtain the same value for the implied volatility for options with different strike prices. In reality, however, volatility is not constant but often has the shape of a smile if plotted against the strike prices. One way to deal with this inconsistency is to average the implied volatilities over a range of strike prices. For the DAX, the average value of the implied volatilities of the first in-the-money and the first out-of-the money call and put option is calculated. For the FTSE 100, the average value of the implied volatilities of the call and put option closest to the at-the-money point is used. So the average value of 4 (2) implied volatilities is calculated for the DAX (FTSE 100) every day.

The definition of an *American* option is similar to the definition of a European option. The only difference is that the owner has the right to exercise his option whenever he or she wants, i.e., he or she can exercise the option *before* maturity. The pricing of American options is more difficult but one can use the prices of the corresponding European options as a rough approximation.

ACKNOWLEDGMENT

Many thanks go to R. Tompkins for valuable discussions on trading options and for suggesting a regime-shifting model for predicting the daily volatility differences.

REFERENCES

[1] C. L. Giles, S. Lawrence, and A. C. Tsoi, "Rule inference for financial prediction using recurrent neural networks," in *Proc.Conf. Comput. Intell. Financial Eng.*, New York, 1997, pp. 253–259.
[2] ——, "Noisy time series prediction using a recurrent neural network and grammatical inference," *Machine Learning*, vol. 44, pp. 161–183, 2000.
[3] C. P. Papageorgiou, "High frequency time series analysis and prediction using Markov models," in *Proc. Conf. Comput. Intell. Financial Eng.*, New York , 1997, pp. 182–185.
[4] ——, "Mixed memory Markov models for time series analysis," in *Proc. Conf. Comput. Intell. Financial Eng.*, New York, 1998, pp. 165–170.
[5] C. Apte and S. J. Hong, "Predicting equity returns from securities data," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: AAAI/MIT Press, 1994, pp. 541–560.
[6] P. Buhlmann, "Extreme events from return-volume process: A discretization approach for complexity reduction," *Appl. Financial Economics*, no. 8, pp. 267–278, 1998.
[7] P. Tiňo, C. Schittenkopf, G. Dorffner, and E. J. Dockner, "A symbolic dynamics approach to volatility prediction," in *Proc. 6th Int. Conf. Computational Finance)*. Cambridge, MA, pp. 137–151.
[8] P. Tiňo and G. Dorffner, "Building predictive models from fractal representations of symbolic sequences," in Advances in Neural Information Processing Systems 12. Cambridge, MA: MIT Press, 2000, pp. 645–651, to be published.
[9] ——, "Predicting the future of discrete sequences from fractal representations of the past," Machine Learning, 2001, to be published.
[10] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1479, 1990.
[11] P. Tiňo, C. Schittenkopf, and G. Dorffner, "Volatility trading via temporal pattern recognition in quantized financial time series," Pattern Anal. Applicat., 2001, to be published.
[12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.
[13] J. Tani and N. Fukurama, "Embedding a grammatical description in deterministic chaos: An experiment in recurrent neural learning," *Biol. Cybern.*, vol. 25, no. 72, pp. 365–370, 1995.
[14] P. Tiňo and M. Koteles, "Extracting finite state representations from recurrent neural networks trained on chaotic symbolic sequences," *IEEE Trans. Neural Networks*, vol. 10, pp. 284–302, Mar. 1999.
[15] J. Noh, R. F. Engle, and A. Kane, "Forecasting volatility and option prices of the s&p 500 index," *J. Derivatives*, pp. 17–30, 1994.
[16] T. Bollerslev, "A generalized autoregressive conditional heteroscedasticity," *Journal of Econometrics*, vol. 31, pp. 307–327, 1986.
[17] E. J. Dockner, G. Strobl, and A. Lessing, "Volatility forecasts and the profitability of option trading strategies," University of Vienna, Austria, Tech. Rep., 1988.
[18] H. White, "Economic prediction using neural networks: The case of IBM daily stock returns," in *Proc. 2nd Annu. IEEE Conf. Neural Networks II*, 1988, pp. 451–458.
[19] B. LeBaron, "Technical trading rule profitability and foreign exchange intervention," *J. Int. Econ.*, vol. 49, pp. 125–143, 1999.
[20] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.
[21] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, UK: Claredon, 1995.
[22] S. Lawrence, I. Burns, A. D. Back, A. C. Tsoi, and C. L. Giles, "Neural-network classification and unequal prior class probabilities," in *Tricks of the Trade: Lecture Notes in Computer Science State-of-the-Art Surveys*, G. Orr, K.-R. Muller, and R. Caruana, Eds. New York: Springer-Verlag, 1998, pp. 299–314.
[23] S. Figlewski, "Forecasting volatility," *Financial Markets, Institutions, Instruments*, vol. 6, no. 1, pp. 1–88, 1997.
[24] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *J. Political Economy*, vol. 81, pp. 637–654, 1973.
[25] M. H. Pearsan and A. Timmermann, "Model instability and choice of observation window," Cambridge Univ., Cambridge, U.K., Tech. Rep. 99-19, 1999.
[26] C. R. Harvey and R. E. Whaley, "Market volatility prediction and the efficiency of the s&p 100 index option market," *J. Financial Economics*, vol. 31, pp. 43–73, 1992.
[27] C. Schmitt and J. Kaehler, "Delta-neutral volatility trading with intra-day prices: An application to options on the index," Zentrum für Europäische Wirtschaftsforschung, Tech. Rep. 96-25, 1996.
[28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[29] Y. Pesin, *Dimension Theory in Dynamical Systems: Rigorous Results and Applications*. Chicago, IL: Univ. Chicago Press, 1997.
[30] Y. Pesin and H. Weiss, "On the dimension of deterministic and cantor-like sets, symbolic dynamics, and the Eckmann-Ruelle conjecture," *Comm. Math. Phys.*, vol. 182, no. 1, pp. 105–153, 1996.
[31] P. Tiňo, "Spatial representation of symbolic sequences through iterative function system," *IEEE Syst., Man, Cybern. A*, vol. 29, pp. 386–392, 1999.
[32] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia," *Machine Learning*, vol. 25, pp. 117–150, 1996.
[33] Ch.-M. Kuan and T. Liu, "Forecasting exchange rates using feedforward and recurrent neural networks," *J. Appl. Econometrics*, vol. 10, no. 4, pp. 347–364, 1995.
[34] P. Tiňo, C. Schittenkopf, and G. Dorffner, "Temporal pattern recognition in noisy nonstationary time series based on quantization into symbolic streams: Lessons learned from financial volatility trading," Adaptive Information Systems and Modeling in Economics and Management Science, WU, Wien, Austria, Tech. Rep. 46, 2000.
[35] R. Tompkins, *Options Explained*. New York: Macmillan, 1994.

Peter Tiňo received the M.Sc. from the Slovak Technical University (STU) and the Ph.D. degree from the Slovak Academy of Sciences in 1997.

He taught at STU, worked at NEC Research Institute, Princeton NJ, and the Austrian Research Institute for AI in Vienna, Austria. He is currently a Research Fellow with the Neural Computation Research Group at Aston University, U.K. His scientific interests include visualization, probabilistic modeling of static and dynamic data, and dynamical systems.

Christian Schittenkopf received the Ph.D. degree in computer science from Technische Universität München, Germany, in 1997. He is currently pursuing the Ph.D. degree in finance at Vienna University, Austria.

He has been a Postdoctoral Fellow at the Austrian Research Institute for Artificial Intelligence since 1998. His research interests include computational finance, dynamical systems, neural networks, and time series analysis.

Georg Dorffner received the M.Sc. degree from Vienna University of Technology, Austria, and the Ph.D. degree from Indiana University, Bloomington, in 1989.

Currently he is Associate Professor at the Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, and heads the Neural Networks Group at the Austrian Research Institute for Artificial Intelligence, Vienna. His research interests are neural networks, time series analysis and economic modeling, signal processing, and connectionist models of conceptual development and language.