Computing and Informatics, Vol. 32, 2013, 1001–1015, V 2013-Nov-6

EXPERIMENTAL ANALYSIS OF THE PREDICTION MODEL BASED ON STRING INVARIANTS

Marek Bundzel

Department of Cybernetics and Artificial Intelligence
Technical University of Košice
Letná 9
040 01 Košice, Slovakia
e-mail: marek.bundzel@tuke.sk

Tomáš Kasanický

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: tomas.kasanicky@savba.sk

Richard Pincak

Institute of Experimental Physics
Slovak Academy of Sciences
Košice, Slovakia
e-mail: pincak@saske.sk

Abstract. A new approach of the string theory called the Prediction Model Based on String Invariants (PMBSI) was applied here to time-series forecast. We used 2-end-point open string that satisfies the Dirichlet and Neumann boundary conditions. The initial motivation was to transfer modern physical ideas into the neighboring field called econophysics. The physical statistical viewpoint has proved to be fruitful, namely in the description of systems where many-body effects dominate. However, PMBSI is not limited to financial forecast. The main advantage of PMBSI includes absence of the learning phase when large number of parameters must be set. Comparative experimental analysis of PMBSI vs. SVM was performed and the results on artificial and real-world data are presented. PMBSI performance was in a close match with SVM.

Keywords: String theory, time-series forecast, econophysics, PMBSI, SVM

1 INTRODUCTION

A new approach of the string theory called the Prediction Model Based on String Invariants (PMBSI) was applied here to time-series forecast. This paper focuses on comparative experimental analysis aimed to identify strengths and weaknesses of PMBSI and to compare its performance to a benchmark, Support Vector Machine (SVM) in this case. PMBSI is based on the approaches described in [9] and extends the previous work. PMBSI also represents one of the first attempts to apply the string theory in the field of time-series forecast and not only in high energy physics. The string theory was developed over the past 25 years and it has achieved a high degree of popularity and respect among the physicists [12]. The initial idea was to transfer modern physical ideas into the neighboring field called econophysics. The physical statistical viewpoint proved the ability to describe systems where many-body effects dominate. However, bottom-up approaches are cumbersome to follow the behavior of the complex economic systems, where autonomous models encounter intrinsic variability. Modern digital economy is founded on data. The primary motivation comes from the actual physical concepts [10, 11]; however, the implementation presented here differs from the original attempts in various significant details.

The time-series forecasting is a scientific field under continuous active development covering an extensive range of methods. Traditionally, linear methods and models are used. Despite their simplicity, linear methods often work well and may well provide an adequate approximation for the task at hand and are mathematically and practically convenient. However, the real life generating processes are often non-linear. Therefore plenty of non-linear forecast models based on different approaches has been created (e.g. GARCH [8], ARCH [7], ARMA [6], ARIMA [5] etc). Presently, the perhaps most used methods are based on Artificial Neural Networks (ANN, covering a wide range of methods) and Support Vector Machines (SVM). A number of research articles compares ANN and SVM to each other and to other more traditional non-linear statistical methods. Tay and Cao ([4]) examined the feasibility of SVM in financial time series forecasting and compared it to a multilayer Back Propagation Neural Network (BPNN). They showed that SVM outperforms the BP neural network. Kamruzzaman and Sarker [3] modeled and predicted currency exchange rates using three ANN based models and a comparison was made with ARIMA model. The results showed that all the ANN based models outperform ARIMA model. Chen et al. [2] compared SVM and BPNN taking auto-regressive model as a benchmark in forecasting the six major Asian stock markets. Again, both the SVM and BPNN outperformed the traditional models.

While the traditional ANN implements the empirical risk minimization principle, SVM implements the structural risk minimization ([1]). Structural risk minimization is an inductive principle for model selection used for learning from finite training data sets. It describes a general model of capacity control and provides a trade-off between hypothesis space complexity and the quality of fitting the training data (empirical error). For this reason SVM is often chosen as a benchmark to compare other non-linear models to and it was also the benchmark of choice here.

2 PREDICTION MODEL BASED ON STRING INVARIANTS

2.1 String Maps

The original time-series $p(\tau)$ is converted as follows

$$\frac{p(\tau+h)-p(\tau)}{p(\tau+h)}, \tag{1}$$

where h denotes the lag between $p(\tau)$ and $p(\tau+h)$, τ is the index of the time series element. On financial data, e.g. on the series of the quotations of the mean currency exchange rate, this operation would convert the original time-series onto a series of returns.

Using the string theory let us first define the 1-end-point open string map

$$P^{(1)}(\tau,h)=\frac{p(\tau+h)-p(\tau)}{p(\tau+h)}, \qquad h\in\langle 0,l_{\mathrm{s}}\rangle \tag{2}$$

where the superscript (1) refers to the number of endpoints and l_{s} to the length of the string (string size). l_{s} is a positive integer.

Here the variable h may be interpreted as a variable which extends along the extra dimension limited by the string size l_{s}. A natural consequence of the transform, Equation (2), is the fulfillment of the boundary condition

$$P^{(1)}(\tau,0)=0, \tag{3}$$

which holds for any τ. To highlight effects of the rare events a power-law Q-deformed model is introduced

$$P^{(1)}(\tau,h)=\left(1-\left[\frac{p(\tau)}{p(\tau+h)}\right]^{Q}\right), \qquad Q>0. \tag{4}$$

The 1-end-point string has defined the origin and it reflects the linear trend in $p(.)$ at the scale l_{s}.

The presence of a long-term trend is partially corrected by fixing $P^{(2)}(\tau,h)$ at $h = l_{\mathrm{s}}$. The open string with two end points is introduced via the nonlinear map which combines information about trends of p at two sequential segments

$$P^{(2)}(\tau,h) = \left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right)\left(1 - \left[\frac{p(\tau+h)}{p(\tau+l_{\mathrm{s}})}\right]^Q\right), \qquad h \in \langle 0, l_{\mathrm{s}}\rangle. \tag{5}$$

The map is suggested to include boundary conditions of *Dirichlet type*

$$P^{(2)}(\tau,0) = P_q(\tau,l_{\mathrm{s}}) = 0, \qquad \text{at all ticks } \tau. \tag{6}$$

In particular, the sign of $P^{(2)}(\tau,h)$ comprises information about the behavior differences of $p(.)$ at three quotes $(\tau, \tau+h, \tau+l_{\mathrm{s}})$. The $P^{(2)}(\tau,h) < 0$ occurs for trends of the different sign, whereas $P^{(2)}(\tau,h) > 0$ indicates the match of the signs.

2.2 String Invariants

The meaning of invariant is that something does not change under transformation, e.g. such as some equations from one reference frame to another. This idea is to be extended on the time-series forecast by finding invariants in the data and utilize them to predict the future values. Similar research aimed to find invariant states of a financial market is described in [16]. Let us introduce a positive integer l_{pr} denoting the prediction scale of how many steps ahead of τ_0 lies the predicted value. Let us introduce an auxiliary positive integer Λ and a condition

$$\Lambda = l_{\mathrm{s}} - l_{\mathrm{pr}}, l_{\mathrm{s}} > l_{\mathrm{pr}}. \tag{7}$$

The power of the nonlinear string maps of time-series data is to be utilized to establish a prediction model similarly as in [13, 14, 15]. We suggest a 2-end-point mixed string model where one string is continuously deformed into the other. The approach to define the string invariants was published before and here it is described in the Appendix (Section 6). The family of invariants is written using the parametrization

$$\begin{aligned} C(\tau,\Lambda) \;=\;& (1-\eta_1)(1-\eta_2)\sum_{h=0}^{\Lambda} W(h) \\ & \times \left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right)\left(1 - \left[\frac{p(\tau+h)}{p(\tau+l_s)}\right]^Q\right) \\ & + \eta_1(1-\eta_2)\sum_{h=0}^{\Lambda} W(h)\left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right) \\ & + \eta_2\sum_{h=0}^{\Lambda} W(h)\left(1 - \left[\frac{p(\tau+h)}{p(\tau+l_s)}\right]^Q\right), \end{aligned} \tag{8}$$

where $\eta_1 \in (-1,1)$, $\eta_2 \in (-1,1)$ are variables (variables which may be called homotopy parameters), Q is a real valued parameter, and the weight $W(h)$ is chosen in the bimodal single parameter form

$$W(h) = \begin{cases} 1 - W_0, & h \le l_s/2, \\ W_0, & h > l_s/2, \end{cases} \tag{9}$$

and

$$W_0 = \frac{1}{\sum_{h'=0}^{l_s} e^{-h'/\Lambda}}. \tag{10}$$

The above is not the only nor the ideal setting of the weight parameters and modifications are planned.

2.3 Making Prediction

$p(\tau_0 + l_{\mathrm{pr}})$ is expressed in terms of the auxiliary variables

$$A_1(\Lambda,\tau) = (1-\eta_1)(1-\eta_2)\sum_{h=0}^{\Lambda} W(h)\left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right), \tag{11}$$

$$A_2(\Lambda,\tau) = -(1-\eta_1)(1-\eta_2)\sum_{h=0}^{\Lambda} W(h)\left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right) p^Q(\tau+h), \tag{12}$$

$$A_3(\Lambda,\tau) = \eta_1(1-\eta_2)\sum_{h=0}^{\Lambda} W(h)\left(1 - \left[\frac{p(\tau)}{p(\tau+h)}\right]^Q\right), \tag{13}$$

$$A_4(\Lambda,\tau) = \eta_2\sum_{h=0}^{\Lambda} W(h), \tag{14}$$

$$A_5(\Lambda,\tau) = -\eta_2\sum_{h=0}^{\Lambda} W(h)p^Q(\tau+h). \tag{15}$$

Thus the expected prediction form reads

$$p(\tau_0 + l_{\mathrm{pr}}) = \left[\frac{A_2(\Lambda,\tau') + A_5(\Lambda,\tau')}{C(\tau_0 - l_{\mathrm{s}},\Lambda) - A_1(\Lambda,\tau') - A_3(\Lambda,\tau') - A_4(\Lambda,\tau')}\right]^{1/Q}, \tag{16}$$

where $\tau' = \tau_0 + l_{\mathrm{pr}} - l_s$, ($\tau' = \tau_0 - \Lambda$). The derivation is based on the invariance

$$C(\tau, l_s - l_{\mathrm{pr}}) = C(\tau - l_{\mathrm{pr}}, l_s - l_{\mathrm{pr}}), \tag{17}$$

and the model will be efficient if

$$C(\tau_0,\Lambda) \simeq C(\tau_0 + l_{\mathrm{pr}},\Lambda). \tag{18}$$

The model's free parameters are l_s, l_{pr}, η_1, η_2 and Q. These must be set during the optimization phase. PMBSI does not require learning phase in the traditional sense.

PMBSI requires the time-series being processed to be non-negative. Otherwise the forecasts will not be defined (NaN). Even so PMBSI sometimes returns NaN values. This problem was fixed here by substitution of the NaN forecast by the most recent input for $l_{\mathrm{pr}} = 1$ (naive prediction) and by the last valid forecast recorded for $l_{\mathrm{pr}} > 1$.

3 EXPERIMENTAL ANALYSIS

3.1 Experimental Setup

The experiments were performed on two time-series. The first series represented artificial data, namely a single period of a sinusoid sampled by 51 regularly spaced samples. The second time series represented proprietary financial data sampled daily over the period of 1295 days. The performance of PMBSI was compared to SVM and to naive forecast. There were two error measures used, mean absolute error (MAE) and symmetric mean absolute percentage error (SMAPE) defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t|, \tag{19}$$

$$\mathrm{SMAPE} = \frac{100}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{0.5(|A_t| + |F_t|)}, \tag{20}$$

where n is the number of samples, A_t is the actual value and F_t is the forecast value.

Each time-series was divided into three subsets: training, evaluation and validation data. The time ordering of the data was maintained; the least recent data were used for training, the more recent data were used to evaluate the performance of the particular model with the given parameters' setting. The best performing model on the evaluation set (in terms of MAE) was chosen and made forecast for the validation data (the most recent) that were never used in the model optimization process. Experimental results on the evaluation and validation data are presented below.

The parameters of the models were optimized by trying all combinations of parameters sampled from given ranges with a sufficient sampling rate. Naturally, this process is slow but it enabled to get an image of the shape of the error surface corresponding to the given settings of parameters and ensured that local minima are explored. The above approach was used for both, PMBSI and SVM.

The SVM models were constructed so that the present value and a certain number of the consecutive past values comprised the input to the model. The input vector corresponds to what will be referred to here as the *time window* with the length l_{tw} (representing the equivalent of the length of the string map l_s by PMBSI).

3.2 Experimental Results on the Artificial Time-Series

The sigmoid data were divided into subsets so that the positive half of the period was used for training and evaluation and the negative half for validation. This was done to assess the ability of PMBSI to extrapolate and generalize. For PMBSI the time series was shifted above zero by adding a positive constant. The constant was then subtracted from the forecast. SVM with linear kernel was used as a benchmark. The positive half of the period was divided 7/3 for training/validation. Predictions of 1, 2 and 3 steps ahead were made. It became obvious that PMBSI performs well in one step ahead prediction but for multiple steps ahead predictions its performance drops rapidly. Therefore, iterated prediction using the one step prediction model was made, improving the PMBSI results significantly. For illustration, Figure 1 shows the comparison of iterated versus the direct prediction using PMBSI. Table 1 shows the experimental results. The results of the best performing models are highlighted.

Method	l_{pr}	MAE eval	MAE valid	SMAPE valid
PMBSI	1	0.000973	**0.002968**	**8.838798**
	2	0.006947	0.034032	14.745538
	3	0.015995	0.161837	54.303315
Iterated PMBSI	1	–	–	–
	2	0.003436	0.011583	10.879313
	3	0.008015	0.028096	14.047025
SVM	1	0.011831	0.007723	10.060302
	2	0.012350	**0.007703**	**10.711573**
	3	0.012412	**0.007322**	**11.551324**
Naive forecast	1	–	0.077947	25.345352
	2	–	0.147725	34.918149
	3	–	0.207250	41.972591

Table 1. Experimental results on artificial time-series

The optimal l_{tw} for SVM was 3 for all predictions. Table 2 shows the optimal settings found for PMBSI. For $l_{pr} = 1$ when PMBSI outperformed linear SVM the optimal length of the string map was shorter than the optimal time window for SVM; in the remaining cases it was significantly longer.

l_{pr}	l_s	Q	η_1	η_2
1	2	0.30	0.80	−0.20
2	5	0.10	0.80	−0.60
3	8	0.10	0.80	−0.60

Table 2. Optimal PMBSI parameters

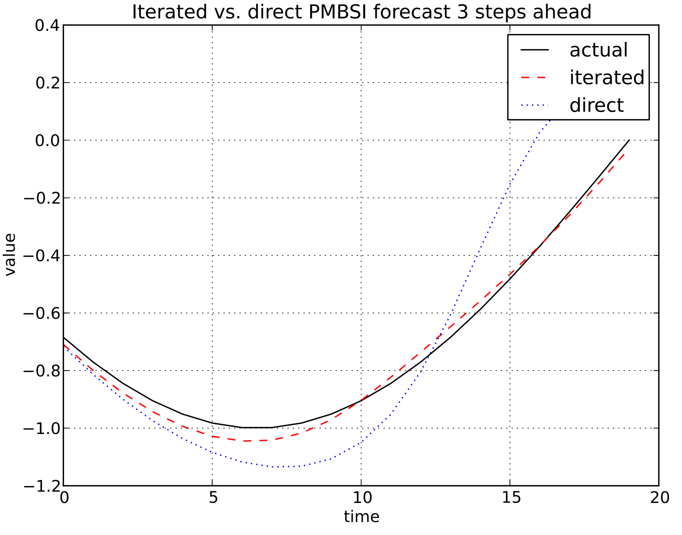

Figure 1. Iterated and direct prediction using PMBSI on artificial data

3.3 Experimental Results on the Financial Time-Series

The financial time-series was divided into subsets so that the most recent 40 % of the data was used for validation and the remaining data were used for training/validation divided in the ratio of 6/4. While extrapolation of sigmoid was a relatively simple task, the financial time-series was highly non-linear and chaotic. SVM with Gaussian RBF kernel was used as the benchmark. Predictions 1–10 steps ahead were made. Table 3 shows a selection of the experimental results. The results of the best performing models are highlighted. Table 4 summarizes the optimal parameters found and states the percent count of NaNs forecast by PMBSI. Interestingly, SVM preferred long time windows reaching the upper limit on the length while PMBSI utilized much less of the past data to make a forecast.

Again, the prediction accuracy of PMBSI was deteriorating significantly for longer forecasts and the results have improved significantly with iterated prediction. The longest prediction of 10 steps ahead was chosen to depict the experimental results on the financial data graphically (Figures 3, 4, 5).

3.4 Analysis

There was a preliminary experimental analysis of the PMBSI method performed. The goal was to evaluate the prediction accuracy, the generalization performance, the convenience of the method in terms of the operators effort needed to prepare

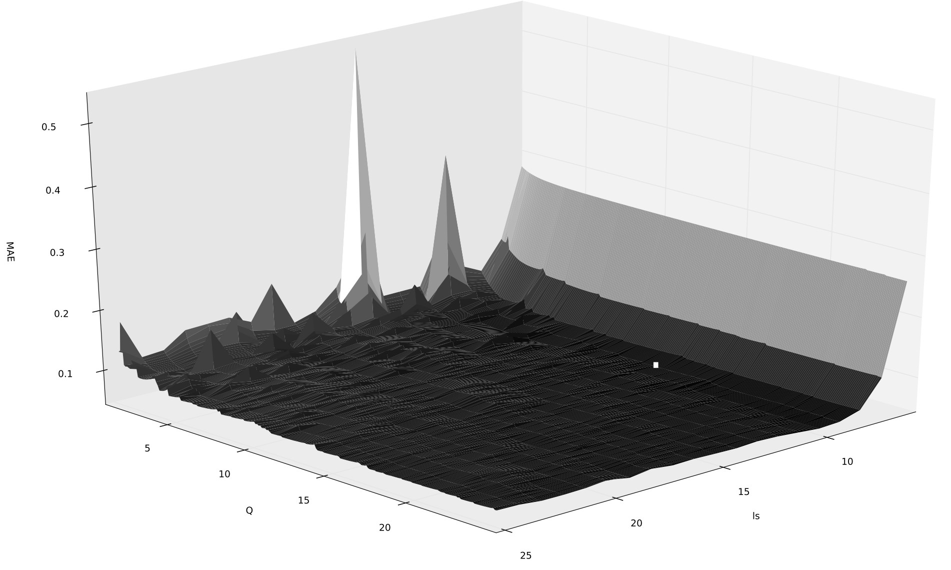

Figure 2. MAE corresponding to various settings of ls and Q on the financial data. The white square is the global minimum of MAE

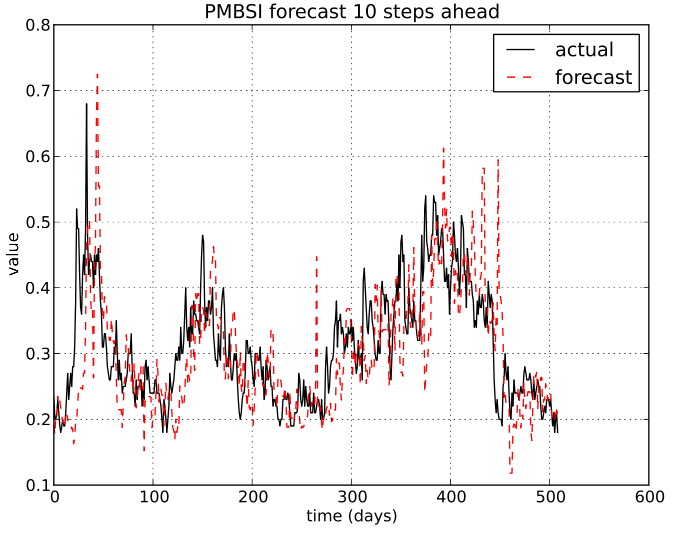

Figure 3. Forecast 10 steps ahead, PMBSI vs. the financial time-series

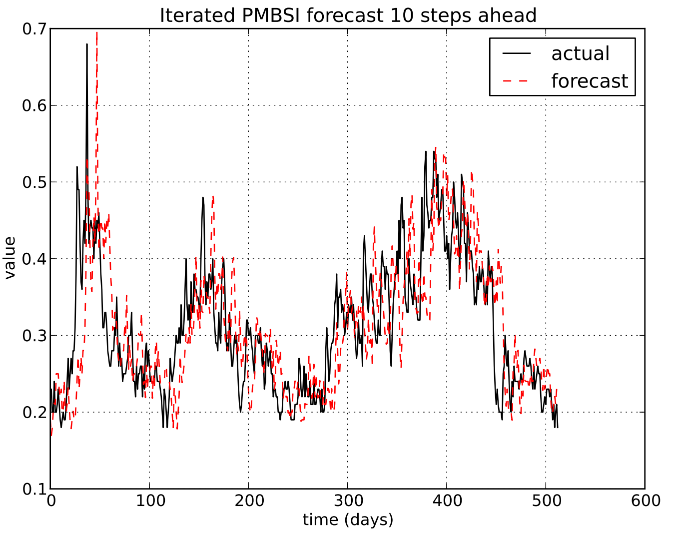

Figure 4. Forecast 10 steps ahead, iterated PMBSI vs. the financial time-series

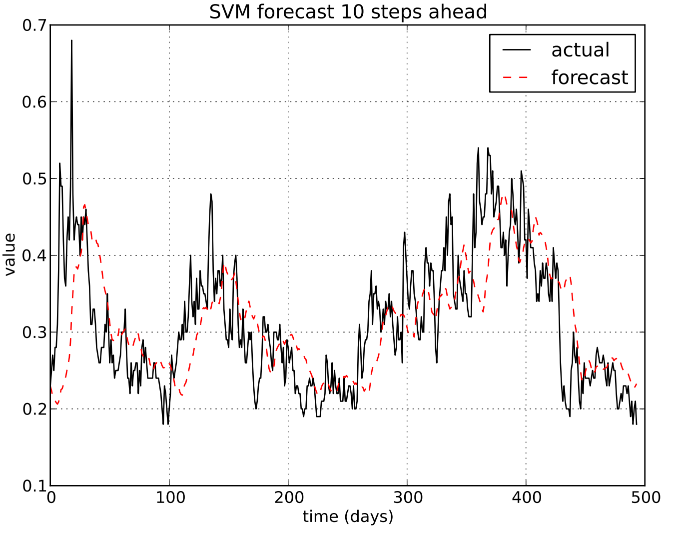

Figure 5. Forecast 10 steps ahead, SVM vs. the financial time-series

Method	l_{pr}	MAE eval	MAE valid	SMAPE valid
PMBSI	1	0.023227	0.023595	7.380742
	2	0.037483	0.036335	11.378275
	4	0.048140	0.046381	14.876330
	6	0.054556	0.049755	16.094349
	8	0.057658	0.056097	18.546008
	10	0.060192	0.058216	18.752986
Iterated PMBSI	1	–	–	–
	2	0.032706	0.031940	9.953547
	4	0.043134	0.042414	13.250729
	6	0.049916	0.047784	15.102693
	8	0.055326	0.051355	16.306971
	10	0.057802	0.052353	16.552731
SVM	1	0.021383	0.025546	8.046289
	2	0.027721	0.031878	10.046793
	4	0.036721	**0.039702**	**12.578553**
	6	0.041984	**0.044450**	**14.157343**
	8	0.044525	**0.047175**	**15.036534**
	10	0.046166	**0.050236**	**15.898355**
Naive forecast	1		**0.023273**	**7.287591**
	2		**0.031486**	**9.822408**
	4		0.041811	13.078883
	6		0.047238	14.958371
	8		0.050788	16.148619
	10		0.051923	16.428804

Table 3. Experimental results on the financial time-series

a working model, computational time and other aspects of the PMBSI method that may have became obvious during the practical deployment. The prediction capability of PMBSI was proven and it was shown that it can match and even outperform SVM in some cases (see the results on the artificial data). On the financial data both methods, SVM and PMBSI, struggled to match the naive forecast. The reason for this is probably the complexity, intrinsic variability and chaotic nature of the system the time-series is describing. Although the tests of PMBSI method on a larger set of time-series are on the way, the presented results have proven that PMBSI can be successfully used for single step forecast. The problem is that the more chaotic is the time series the shorter is the period when the invariant Equation (18) is fulfilled. Therefore PMBSI is effective for the single step prediction because the probability of a significant change in the time series is lower. The situation is different for multiple steps prediction leading to small efficiency in such cases.

The way to improve the performance in multiple steps predictions is to chose more appropriate weighting coefficients (Equations (9), (10)). The optimized weights

	SVM	PMBSI				
l_{pr}	l_{tw}	l_s	Q	η_1	η_2	NaN (%)
1	51	2	20.3	0.0	0.0	20.43
2	51	5	15.5	0.0	−0.05	14.6
4	51	8	11.9	0.0	0.0	19.25
6	51	10	14.3	0.1	−0.05	21.42
8	51	18	16.4	0.4	0.10	25.69
10	51	14	11.9	0.3	0.05	21.32

Table 4. Optimal parameters on the financial time-series and percent of NaNs forecast by PMBSI

should considerably extend the interval where Equation (18) is fulfilled. Therefore this is planned in the future development.

PMBSI predictor does not undergo a training process that is typical for ANN and SVM where a number of free parameters must be set (synaptic weights by ANN, α coefficients by SVM). PMBSI features a similar set of weights (W) but often very small and calculated analytically. The parameters to be optimized are only four: ls, Q, η_1, η_2. This, clearly, is an advantage. On the other hand the optimal setting of the parameters is not easy to be found as there are many local minima on the error surface. In this analysis the optimal setting was found by testing all combinations of parameters sampled from given ranges. Figure 2 shows the Mean Absolute Error (MAE) of the 5-steps ahead forecast of the financial time series corresponding to various settings of ls and Q ($\eta_1, \eta_2 = 0$); but the figure makes also obvious that PMBSI's performance is approximately the same for a wide range of settings, making it unnecessary to explore the whole error surface. Because of this PMBSI can be fast to construct and to deploy.

4 FUTURE WORK

Beside the further test of PMBSI we consider that fast methods for optimization of parameters must be developed. Because of the character of the error surface we have chosen to use evolutionary optimization as the method of choice. After a fast and successful parameters' optimization method is developed, optimization of the weighting parameters (Equations (9), (10)) will be included into the evolutionary process. Further extensive experimental testing is to be performed.

5 CONCLUSION

A new prediction method PMBSI based on the string theory was described and tested on artificial and real-world data. The experimental results are shown. It has been proven that PMBSI is viable and further development was outlined. Finally we can conclude that PMBSI is applicable with a good efficiency for iterative prediction using a single step forecast when it matches and sometimes outperforms SVM. The main advantage of PMBSI is the low number of free parameters compared to learning methods such as SVM and ANN.

Acknowledgment

This work is partially the result of implementation of the project "Development of the Center of Information and Communication Technologies for Knowledge Systems" (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

The calculations were partly done on a computational cluster designed and managed by the company FURT Solutions, s.r.o., Košice, Slovak Republic.

The work was supported by the Slovak Academy of Sciences in the framework VEGA Grant No. 2/0037/13. R. Pincak would like to thank the TH division in CERN for hospitality.

6 APPENDIX: CORRELATION FUNCTION AS INVARIANT

A correlation function is a statistical correlation between random variables at two different points in our case the strings in time series. For simplicity in the Appendix we used only one point strings (Equation (4)) with parameter $Q = 1$. Usually the correlation function is defined as $C(\tau, l_0) = \langle P^1(\tau, l_0) P^1(\tau + 1, l_0)\rangle$. We suppose the invariant in the form of the correlation function

$$C(\tau, l_0) = \sum_{h=l_0}^{h=l} W(h) \left(1 - \frac{p(\tau - h)}{p(\tau - 1 - h)}\right) \left(1 - \frac{p(\tau - 1 - h)}{p(\tau - 2 - h)}\right), \tag{21}$$

with weight $W(h)$ defined as in Equation (9). We assume the condition of the invariance between close strings in τ and at the next step $\tau + 1$ in time series (exact meaning of the one step prediction) in the form

$$C(\tau, l_0) = C(\tau + 1, l_0). \tag{22}$$

Now we want to find exact expression for the one step prediction $p(\tau + 1)$. Therefore we evaluate one step correlation invariant Equation (22) with initial condition $l_0 = 0$

$$\begin{aligned} W(0) \quad & \left(1 - \frac{p(\tau)}{p(\tau - 1)}\right) \left(1 - \frac{p(\tau - 1)}{p(\tau - 2)}\right) = \\ W(0) \quad & \left(1 - \frac{p(\tau + 1)}{p(\tau)}\right) \left(1 - \frac{p(\tau)}{p(\tau - 1)}\right) + \\ W(1) \quad & \left(1 - \frac{p(\tau)}{p(\tau - 1)}\right) \left(1 - \frac{p(\tau - 1)}{p(\tau - 2)}\right), \end{aligned} \tag{23}$$

which can be rewritten in the more compact form

$$C(\tau,0) = W(0)\left(1-\frac{p(\tau+1)}{p(\tau)}\right)\left(1-\frac{p(\tau)}{p(\tau-1)}\right)+C(\tau+1,1) \tag{24}$$

and

$$\left(1-\frac{p(\tau+1)}{p(\tau)}\right) = \frac{C(\tau,0)-C(\tau+1,1)}{W(0)\left(1-\frac{p(\tau)}{p(\tau-1)}\right)}. \tag{25}$$

We finally obtain the prediction

$$p(\tau+1) = p(\tau)\left(1+\frac{C(t+1,1)-C(t,0)}{W(0)\left(1-\frac{p(\tau)}{p(\tau-1)}\right)}\right), \tag{26}$$

valid for $p(\tau) \neq p(\tau-1)$. These are general definitions for the one step prediction correlation invariants. A similar way (Equation (16)) can be found for 2-end-point and 1-end-point mixed strings model with $Q > 0$.

REFERENCES

[1] Vapnik, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., ISBN: 0-387-94559-8, 1995.

[2] Chen, W.-H.—Shih, J.-Y.—Wu, S.: Comparison of Support-Vector Machines and Back Propagation Neural Networks in Forecasting the Six Major Asian Stock Markets. Int. J. Electronic Finance, Vol. 1, 2006, No. 1, pp. 49–67.

[3] Kamruzzaman, J.—Sarker, R.: Forecasting of Currency Exchange Rates Using ANN: A Case Study. Proc. IEEE Intl. Conf. on Neur. Net. & Sign. Process. (ICNNSP03), China, 2003.

[4] Tay, F. E. H.—Cao, L.: Application of Support Vector Machines in Financial Time-Series Forecasting. Omega, Vol. 29, 2001, pp. 309–317.

[5] Box, G. E. P.—Jenkins, G. M.: Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day, 1970.

[6] Deistler, M.: The Structure of ARMA Systems in Relation to Estimation, in Geometry and Identification. Proceedings of APSM Workshop on System Geometry, System Identification, and Parameter Estimation, Systems Information and Control, Vol. 1, edited by P. E. Caines and R. Hermann, 1983, pp. 49–61. Brookline, MS: Math Sci Press.

[7] Engle, R.: Autoregressive Conditional Heteroskedasticity with Estimates of United Kingdom Inflation. Econometrica, Vol. 50, 1982, pp. 987–1008.

[8] Bollerslev, T.: Generalized Autoregressive Conditional Heteroskedasticity, Journal of Econometrics, Vol. 31, 1986, pp. 307–327.

[9] Horváth, D.—Pincak, R.: From the Currency Rate Quotations Onto Strings and Brane World Scenarios. Physica A: Statistical Mechanics and Its Applications. Volume 391, 2012, Issue 21, pp. 5172–5188.

[10] McMahon, D.: String Theory Demystified. The McGraw-Hill Companies, Inc., 2009.

[11] Zwiebach, B.: A First Course in String Theory. Cambridge university press, 2009.

[12] Polchinski, J.: String Theory. Cambridge University Press, 1998.

[13] Christiansen, J. D.: The Journal of Prediction Market 1, 17. 2007.

[14] Chang, Ch. Ch.—Hsieh Ch. Y.—Lin, Y. Ch.: Applied Economics Letters iFirst, 1. 2011.

[15] Wolfers, J.—Zitzewitz, E.: Prediction Markets. Journal of Economic Perspectives, Vol. 18, 2004, No. 107.

[16] Münnix, M.—Shimada—Takashi, R.—Schäfer—Seligman, F.—Leyvraz, T. H.—Guhr, T.—Stanley, H. E.: Identifying States of a Financial Market. Quantitative Finance Papers, arXiv.org, 2012.

Marek Bundzel is with the Technical University of Košice, Department of Cybernetics and Artificial Intelligence, Slovakia. He is active in teaching and research. His expertise includes mainly methods of computational intelligence like Artificial Neural Networks, Support Vector Machines and Evolutionary algorithms and applications of the above methods in pattern recognition, forecast and robotics. Marek Bundzel has spent two years at Waseda University, Tokyo where he was developing a model based on the memory-prediction framework, a theory of brain function, for the purposes of object recognition in mobile robot.

Tomáš Kasanický is a researcher at Institute of Informatics of Slovak Academy of Sciences and a Ph. D. student at Faculty of Electrical Engineering and Information Technology, Slovak University of Technology Slovakia. His main research areas are Artificial Neural Networks, Artificial immune systems and bio-inspired algorithms. He has participated in several national and European research projects.

Richard Pincak is with Institute of Experimental Physics, Slovak Academy of Sciences; he was several years senior researcher in Bogoliubov Laboratory of Theoretical Physics, Joint Institute for Nuclear Research, Dubna, Russia. He is an active theoretical physicist with a lot of branches of research. The last time he is also trying to find the fundamental revalualution principles of the behavior prediction of financial forex market and options pricing. He studies the projections of the real exchange rate dynamics onto the string-like topology. His approach in this field is inspired by the contemporary movements in the string theory.