The FL-SMIA Network: A Novel Architecture for Time Series Prediction

Asmaa Mahdi and Tillman Weyde
School of Mathematics, Computer Science and Engineering
City, University of London
London, UK
Email: {asmaa.mahdi, t.e.weyde}@city.ac.uk

***Abstract*—In this paper we propose the FL-SMIA model. This is a novel neural network model that combines the principles of the Functional Link Neural Network (FLNN) with the Self-organizing Multilayer Neural Network using the Immune Algorithm (SMIA). We describe the FL-SMIA architecture and operation and evaluate its predictive performance on different financial time series in comparison to other neural network models. The FL-SMIA model combines the higher-order inputs of the tensor-product FLNN, i.e. the products of raw input features, with the self-organizing hidden layer of SMIA that dynamically grows and adapts to the input vectors. The FL-SMIA has two advantages over other models. First that it can dynamically adapt to growing data with model that grows increasingly complex. Second, it keeps an explicit representation of the patterns it recognises in the data.**

Experimental results show that FL-SMIA improves performance as measured by annualised return in five-days-ahead and one-day-ahead prediction tasks for share prices and exchange rates over the SMIA networks alone and over standard multilayer perceptrons. It performs on the same level as the FLNN, sometimes better but not significantly so. The result that FLNN and FL-SMIA outperform multilayer models indicates that particularly the higher-order features contribute to the improved performance and motivate further research into mixed neural network architectures for financial time series prediction.

I. Introduction

Over the last 20 years, the problem of predicting financial time-series data has attracted much interest from both commercial and academic communities, which resulted in a wide range of investigations. Predictive models are contributing to decisions on economic policies by governments and investments by multinational companies relying on computer modeling and forecasts [1], [2] and [3]. Financial time series are highly non-linear and complex [4], as many risk factors, such as political events, weather conditions, and dynamics of financial market themselves affect prices and exchange rates [5].

Artificial Neural Networks (ANNs) as non-linear models have long been seen as promising and used extensively in financial time series prediction [6], [7] but they suffer from some problems, particularly overfitting on smaller datasets [8], [9].

In 1987 Giles [10] introduced the *Higher Order Neural Network*, which was analyzed and improved by [11], leading to the *Functional Link Neural Network* (FLNN). The FLNN was presented to reduce the overfitting problem by removing the hidden layer from the ANN architecture to help reduce the model complexity. The FLNN provides an enhancement of input units to enable the network to perform to non-linearly separable classification tasks. Another approach to improve over MLPs is based on alternative learning methods using to prototypes or clustering, such as Adaptive Resonance Theory [12], or algorithms inspired by artificial immune systems [13] such as the *Self-organized Multilayer neural network using the Immune Algorithm* (SMIA) [14], where the internal representations expand depending on the training data.

In this paper, the integration of the FLNN tensor product model and SMIA in the *Functional Link Self-organized Multilayer neural network using the Immune Algorithm* (FL-SMIA) is proposed as a novel method for financial time series prediction. The rest of this paper is organized as follows: Section 2 discusses related work in the literature. The FL-SMIA network architecture and learning method is detailed in Section 3. The experimental design, datasets, pre-processing, training, and testing of different models are presented in Section 4. The simulation results are presented and discussed in Section 5. Finally, conclusions and perspectives for future work are provided in Section 6.

II. Related Work

Although the main focus of machine learning research has recently has been on deep, recurrent or convolutional neural networks operating on raw data, methods for constructing features and alternative learning algorithms have still potential for improving predictive performance. We focus here on Functional Link Neural Networks and the Immune Algorithm.

A. The Functional Link Neural Network

The Functional Link Neural Network (FLNN) is a type of Higher Order Neural Network (HONNs) that utilizes combination of its inputs [11]. The tensor product model is one type of FLNN where the network input is extended with products of input features. For example, with three inputs features X_1, X_2, X_3 the second order terms X_1X_2, X_1X_3, X_2X_3 can be added to the input layer and also the third order $X_1X_2X_3$ as shown in Figure 1. This model utilizes the joint activation between the input units to extend the input space without adding any external information. The principle of the tensor model has been used in this research to add extra inputs to

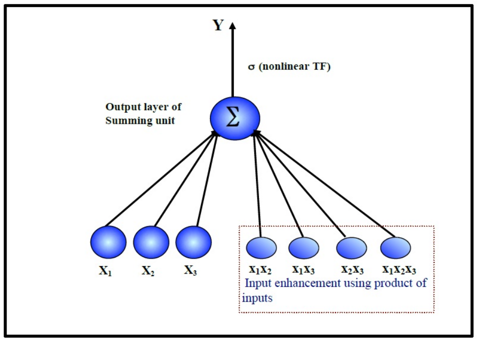

Fig. 1. The FLNN-tensor product model following [11].

the proposed network. Although the architecture of FLNN is simple, it leads a network with greater learning capacity compared to a model using only input features directly as shown in [10] and [15]. The FLNN architecture can suffer from combinatorial explosion, due to an exponential increase in the number of inputs units. Therefore, only second or third order networks are typically used in practice [16], [17].

B. The Immune Algorithm

Artificial Immune Systems (AIS) have been inspired by natural immune systems and the idea of the Immune Algorithm is based on the behaviours of the antigens and B cells in biological immune systems as initially discussed in [18]. The basic concept of the Immune Algorithm is a set of B cells, which each respond to a set of antigens, thus clustering patterns in the training data [19]. The self-organization inspired by the immune system appeared in [20], where the researchers used one layer networks combined with contiguity constrained method by [21] for clustering analysis. Later, the network has been extended with a back-propagation output layer [22]. This approach has been adapted for the use with financial time series in the SMIA model in [14] and is extended with product term inputs in this work.

III. The Functional Link Self-organizing Multilayer Network using the Immune Algorithm (FL-SMIA)

The proposed Functional Link-Self-organized Multilayer network using the Immune Algorithm combines aspects of Functional Link Neural Network with the Self-organizing Multilayer network using the Immune Algorithm in the structure and the learning algorithm.

A. The Structure of the FL-SMIA network

The architecture of the proposed FL-SMIA network consists of the input layer, which comprises a number of input units $X_1, X_2, \ldots, X_Z$, the self-organizing hidden layer with units $H_1, H_2, \ldots, H_N$, and the output layer consisting of one output unit as shown in Figure 2. Here Z, and N refers to the number of units in each layer. This research focuses on adding second order terms to the input units. In our example below the network has five input features $X_1, \ldots, X_5$. Adding the second order term to the inputs results in 10 additional inputs $(X_1X_2, X_1X_3, X_1X_4, \ldots, X_4X_5)$, leading to fifteen input units in total, five with input features and ten with products of inputs features as represented in Figure 2. The FL-SMIA network uses a hidden layer which operates like in [14] and [22]. The design of the hidden units is inspired by B cells recognising pathogens in immune systems.

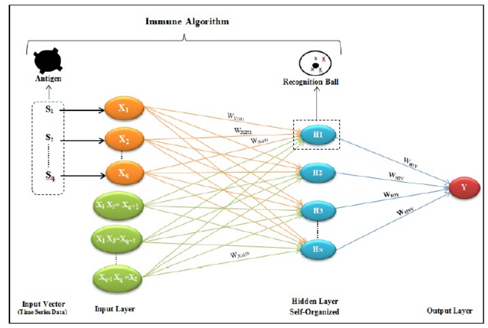

Fig. 2. The proposed FL-SMIA architecture (Functional Link Self-organized Multilayer inspired by Immune Algorithm).

The output of the hidden units is determined using the Euclidean distance between the input units (X_i) and the connection weights between the input units and the hidden units (W_{Hij}). The advantage of using the Euclidean distance is to make the network capable of exploiting local information of the input data. The output of a hidden unit H_j is calculated as:

$$H_j = f_{hts}\left(\sqrt{\sum_{i=1}^{Z}(W_{Hij} - X_i)^2} + B_j\right) \tag{1}$$

where W_{Hij} represents the weight of the connection from the i^{th} input unit to the j^{th} hidden unit, and f_{hts} is the hyperbolic tangent sigmoid function. The number of hidden units is determined from the data by learning with the Immune Algorithm as described in the next section. The outputs of the hidden units are aggregated in a standard layer with the network output given by:

$$Y = f_{ls}\left(\sum_{j=1}^{N} W_{Hjy}.H_j + B_y\right) \tag{2}$$

where W_{Hjy} represent the strength of the connection weights between the j^{th} hidden unit and the output unit, B_y is the bias of the output unit Y, and f_{ls} is the logistic sigmoid function.

B. Learning in the FL-SMIA network

The FL-SMIA as described above has two weight matrices, the first between the input layer and the hidden layer, the second between the hidden layer and the output layer. The second weight matrix is trained using the standard back-propagation algorithm [23] with regularisation to penalise large weights [24] in batch mode. In our case with a single hidden output neuron the weight change is calculated as:

$$\Delta W_{Hjk} = -\eta_b \frac{\partial J}{\partial W_{Hj}} - \lambda W_{Hjk} \quad (3)$$

where W_{Hjk} is the weight of the connection from hidden units Hj to the output unit, $\eta_b \in [0, 1]$ is the learning rate, and J the mean squared error on the training set. The second term on the right-hand side effects the regularisation, which is controlled by the parameter λ. The bias is adapted in the same way but without regularisation.

Before the second weight matrix is trained, the first set of weights and the structure of the hidden layer is trained using the Immune Algorithm [22] as indicated in Figure 2. In the Immune Algorithm a hidden unit corresponds to a recognition ball (BR) in the immune system. Each hidden unit represents one or more input vectors with the weights of the connections from the input layer to the hidden unit. The hidden unit Hj is represented by (P_j, W_{Hj}), where P_j is the number of input vectors associated with Hj and W_{Hj} is the vector of weights from the input layer to Hj.

We start with one hidden unit ($N = 1$) and the first hidden unit is created with $P_1 = 1$ and $W_{H1} = X_1$. The Immune Algorithm then performs the following steps to create and update the hidden units until all inputs of the network have found their corresponding hidden unit.

1. For $m = 1, \ldots, M$ perform the following:
   i. For $j = 1, \ldots, N$, calculate the Euclidean distance between the m-th input and the weight vector of the j^{th} hidden unit:

   $$dist_{mj} = \sqrt{\sum_{i=1}^{Z} (x_{mi} - w_{Hji})^2}$$

   where x_{mi} is the i^{th} element of input vector x_m, and w_{Hji} is the i^{th} component of vector w_{Hj}.
   ii. Determine the closest unit c, i.e. the unit with the shortest distance to x_m:

   $$dist_{mc} = min_j(dist_{mj})$$

   iii. If the shortest distance $dist_{mc}$ is below a stimulation level Sq (where Sq is selected between 0 and 1), then the input has found its corresponding hidden unit. In this case the weight vector w_{Hc} of the hidden unit closest to x_m will be updated to the centroid of all assigned vectors and P_c will be incremented by 1:

   P_c= P_c+1

   Then update the weight vector of the hidden units:

   $$w_{Hc_{new}} = w_{Hc} + \eta * dist_{mc}$$

   Where $\eta_i \in (0, 1)$ is the learning rate for the Immune Algorithm, w_{Hc} is the weight vector of the hidden unit closest to x_m.

   Otherwise the shortest distance $dist_{mc}$ is greater than the stimulation level Sq (no hidden unit is found) and we create a new hidden unit (P_N, W_N) with $P_N = 1$ and $W_{HN} = X_m$. Then we update: $N = N + 1$
   And the values of P_k for $(k = 1...v)$ are set to 1.
2. Repeat 1 as long as new hidden units have been created.

As an alternative to the SMIA method, we introduced a second method to update the connection weights from input units to hidden units after we found the shortest distance $dist_{mc}$. This method calculates the new vector for the selected hidden unit as the average of all input vectors assigned to the unit:

$$w_{Hc_{new}} = \frac{Hc(P_j - 1) + x_m}{P_j}.$$

We refer to the second method above FL-SMIA* in the section of Simulation Results and Discussion.

By applying the Immune Algorithm in either variant, a hidden layer representation is created that reflects the variety of vectors in the input data, which helps avoid over- and under-fitting problems because the hidden layer expands with the size of the training data. The nodes in the hidden layer contain explicit patterns that the network uses for predictions, which can be examined and interpreted by financial domain experts. This form of evaluation was however outside the scope of this paper.

IV. Experimental Design

In this work, seven financial time series have been used to evaluate various neural network architectures. We use time series data available from the Federal Reserve, Board of Governors. There are three types of financial time series used in this work: exchange rate prices, stock opening prices, and stock closing prices. The exchange rate time series and the stock prices are daily time series covering the period from 1/7/2002 to 11/11/2008, giving 1605 trading days as shown in Table I. We use the values of the last five days as the input features for all models.

Financial time series data are known to be highly noisy, non-stationary signals. The relative difference of the price (RDP) has been used in this work in order to reduce the non-stationarity. This transformation makes the distribution of the data more symmetrical and closer to a normal distribution.

A. Training and testing of the proposed network

The performance of the FL-SMIA network has been compared against the performance of several other NN architectures, using the method of [14], specifically with Multilayer Perceptron (MLP), Functional Link Neural Network

TABLE I
FINANCIAL TIME SERIES DATASETS.

No	Time series dataset	Acronym
1	US dollar to UK pound exchange rate	US/UK
2	US dollar to EURO exchange rate	US/EU
3	Japanese yen to US dollar exchange rate	JP/US
4	NASDAQ composite stock opening price	NQO
5	NASDAQ composite stock closing price	NQC
6	Dow Jones Industrial average opening stock price	DJO
7	Dow Jones Industrial average closing stock price	DJC

(FLNN), Regularised Multilayer Perceptron (R-MLP), Self-organised Multilayer neural network using the Immune Algorithm (SMIA), and Regularised SMIA (R-SMIA). We apply 4-fold cross-validation with training validation and test sets comprising 50%, 25%, and 25% of the data. Early stopping has been used for all networks.

B. Evaluation metrics

To measure the network's financial applicability we have devised a simple trading strategy and run simulations. The trading strategy is simply to *buy* if the network forecasts a positive change and to *sell* if the network predicts a negative change for the next period. We focus on Annualised Return (AR) to evaluate the overall performance, measuring the total profitability of the strategy over a year [14].

The Annualised Return (AR) measure estimates the effectiveness of a model for automatic trading. It measures the total profitability in a year of a strategy using *buy* and *sell* signals generated by the models [25]. The Annualised Return (AR) is calculated as follows following [25]:

$$AR = (Profit/AllProfit) * 100$$

$$Profit = (252/n) * CR$$

$$CR = \sum_{i=1}^{n}(R_i)$$

$$R_i = +|y_i| \text{ if } (y_i)(y_i^*) >= 0 \text{ , otherwise } R_i = -|y_i|$$

$$AllProfit = (252/n) * \sum_{i=1}^{n} abs(R_i)$$

Where n is the total number of the data sample, y_i is the target output value and y_i^* represent the predicted output value, and (R_i) refers to the returns. 252 is taken as the number of trading days per year.

We have also used a variety of statistical metrics to further evaluate the performance of the models: the Signal to Noise Ratio (SNR), the Normalized Mean Squared Error (NMSE), Mean Squared Error (MSE), and the Mean absolute Error (MAE) [26], [25], which we are not reporting here.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The networks have been tested on all datasets from Table I using the metrics described above.

A. Five days ahead prediction

For five steps ahead prediction, the simulation results show the best average annualised return on all seven datasets for the FL-SMIA network followed by the FLNN (88.58% vs. 87.58%) as shown in Table II. This gives evidence that the Immune Algorithm helps improve the performance compared to multilayer perceptron networks. The FL-SMIA, FLNN and FL-SMIA* models generally outperform the MLP models. The best two models are the ones using product terms (FLNN and FL-SMIA).

B. One day ahead prediction

The AR results for one day ahead prediction as shown in Table III . The FL-SMIA here produced the second highest result on all average for the exchange rates and stock prices, only outperformed by the FLNN (72.00% vs. 72.85%). The FL-SMIA network has higher annualised return in all seven datasets than the MLP and R-MLP networks. The FLNN is, however, better on average indicating the effectiveness of product terms in predicting financial data.
Interestingly, the MSE for training and test is higher for the FL-SMIA than for FLNN or MLP, but the FL-SMIA delivers higher AR in the five day case nevertheless.

C. Statistical Evaluation

Two statistical test have been used in this research to determine the difference between FL-SMIA and other networks:

1) Significance of differences in AR: We used the Wilcoxon signed rank test for paired samples to determine differences in the overall AR performance of the models over all seven datasets for five day ahead prediction. We found that the FL-SMIA produces significantly better AR values than all other networks ($p < 0.01$), except the FLNN where the difference is not significant at the 0.05 threshold for p.

2) Similarity between residuals: To measure the similarity between model behaviour over the datasets, we used the correlation coefficient between applied to the residuals for each model. The results showed that FL-SMIA performs more similarly to FLNN, SMIA and R-SMIA than to MLP and R-MLP, indicating that some characteristics of FLNN and SMIA components are retained.

VI. CONCLUSION

We proposed the FL-SMIA network, a novel architecture combining product term inputs from Functional Link Neural Networks with a self-organising hidden layer, using the two variants Immune Algorithm as in SMIA networks. We evaluated the FL-SMIA on seven financial datasets for one and five day predictions of prices and exchange rates. The FL-SMIA performed similar to the FLNN, worse on the one day prediction, better on five day prediction but not significantly different in either case. The difference to SIMA and multilayer perceptron models is significant. The results show that overall both the self-organisation with the Immune Algorithm and particularly the higher-order terms as in the FLNN contribute

TABLE II
THE ANNUALISED RETURN FOR FIVE DAYS AHEAD PREDICTION IN PERCENT.

Network	US/UK	US/EU	JP/US	NQO	NQC	DJO	DJC	Average
MLP	86.40	87.59	84.78	76.88	71.66	79.20	86.12	81.80
FLNN	87.24	91.90	85.70	**87.32**	86.37	86.71	**87.79**	87.58
SMIA	89.97	91.99	87.16	86.54	85.58	83.66	86.87	87.39
R-MLP	91.16	91.50	83.34	83.21	80.77	80.40	84.38	84.97
R-SMIA	88.40	91.93	86.24	86.57	87.26	84.47	86.47	87.33
FL-SMIA	**92.02**	**92.50**	**87.49**	86.63	**87.34**	86.63	87.45	**88.58**
FL-SMIA*	88.47	91.18	86.03	83.96	86.06	**87.10**	86.53	87.05

TABLE III
THE ANNUALISED RETURN FOR ONE DAY AHEAD PREDICTION IN PERCENT.

Network	US/UK	US/EU	JP/US	NQO	NQC	DJO	DJC	Average
MLP	71.49	71.54	71.41	67.61	65.16	59.15	60.24	66.66
FLNN	75.23	**78.07**	77.05	70.13	66.13	**71.50**	**71.85**	**72.85**
SMIA	74.24	74.04	**78.91**	69.26	65.48	71.23	67.29	71.49
R-MLP	72.11	73.24	74.50	68.31	66.15	67.77	64.76	69.55
R-SMIA	73.71	78.13	75.65	**71.89**	**67.11**	62.45	62.59	70.22
FL-SMIA	**76.22**	76.65	75.04	70.42	66.70	68.69	70.28	72.00
FL-SMIA*	73.11	72.81	73.57	69.37	66.40	67.96	65.62	69.83

to improved performance on financial time series prediction compared to multilayer perceptrons.

These results are encouraging for future work on new network architectures that include both these elements. Interesting directions include combinations with are deeper and recurrent networks. The observation that the lowest mean squared errors did not coincide with the highest annualised returns indicates another area of further investigation.

ACKNOWLEDGMENT

Asmaa Mahdi would like to thank the Republic of Iraq, Ministry of Higher Education and Scientific Research for supporting her Ph.D study.

REFERENCES

[1] D. K. Bebarta, A. K. Rout, B. Biswal, and P. Dash, “Forecasting and classification of indian stocks using different polynomial functional link artificial neural networks,” in *2012 Annual IEEE India Conference (INDICON)*. IEEE, 2012, pp. 178–182.

[2] K. Li, “Forecasting of government’s financial educational fund by using neural networks model,” in *Genetic and Evolutionary Computing, 2008. WGEC’08. Second International Conference on*. IEEE, 2008, pp. 120–123.

[3] K. Zou and R. Dong, “A novel approach for time series analysis based rbf neural network,” in *Information Technology and Applications (IFITA), 2010 International Forum on*, vol. 3. IEEE, 2010, pp. 139–142.

[4] Y. Li and F. Ying, “Multivariate time series analysis in corporate decision-making application,” in *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, vol. 2. IEEE, 2011, pp. 374–376.

[5] C. Chatfield, “The analysis of time series: theory and practice,” *Chapman Hall*, 1975.

[6] E. Guresen, G. Kayakutlu, and T. U. Daim, “Using artificial neural network models in stock market index prediction,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 10 389–10 397, 2011.

[7] G. Zhang and M. Y. Hu, “Neural network forecasting of the british pound/us dollar exchange rate,” *Omega*, vol. 26, no. 4, pp. 495–506, 1998.

[8] S. Lawrence and C. L. Giles, “Overfitting and neural networks: conjugate gradient and backpropagation,” in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 1. IEEE, 2000, pp. 114–119.

[9] F. E. Tay and L. Cao, “Application of support vector machines in financial time series forecasting,” *Omega*, vol. 29, no. 4, pp. 309–317, 2001.

[10] C. L. Giles and T. Maxwell, “Learning, invariance, and generalization in high-order neural networks,” *Applied optics*, vol. 26, no. 23, pp. 4972–4978, 1987.

[11] Y. Pao, “Adaptive pattern recognition and neural networks,” 1989.

[12] G. A. Carpenter and S. Grossberg, *Adaptive Resonance Theory*, 2nd ed. Cambridge, MA: MIT Press, 2010, pp. 87–90.

[13] L. N. De Castro and J. Timmis, *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media, 2002.

[14] A. Mahdi, A. J. Hussain, P. Lisbo, and D. Al-Jumeily, “The application of the neural network model inspired by the immune system in financial time series prediction,” in *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*. IEEE, 2009, pp. 370–376.

[15] L. Mirea and T. Marcu, “System identification using functional-link neural networks with dynamic structure,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 205–210, 2002.

[16] T. Kaita, S. Tomita, and J. Yamanaka, “On a higher-order neural network for distortion invariant pattern recognition,” *Pattern Recognition Letters*, vol. 23, no. 8, pp. 977–984, 2002.

[17] G. Thimm, “Optimization of high order perceptrons,” 1997.

[18] J. Timmis, “Artificial immune systems: a novel data analysis technique inspired by the immune network theory,” Ph.D. dissertation, Department of Computer Science, 2000.

[19] X. Shen, X. Z. Gao, and R. Bie, “Artificial immune networks: Models and applications,” *International Journal of Computational Intelligence Systems*, vol. 1, no. 2, pp. 168–176, 2008.

[20] R. Widyanto, Y. T. Megawati, and K. Hirota, “Clustering analysis using a self-organized network inspired by immune algorithm,” in *Proceeding of the IASTED International Conference on Artificial and Computational Intelligence, ACTA Press, Tokyo*, 2002, pp. 197–202.

[21] M. Y. Kiang, “Extending the kohonen self-organizing map networks for clustering analysis,” *Computational Statistics & Data Analysis*, vol. 38, no. 2, pp. 161–180, 2001.

[22] M. R. Widyanto, H. Nobuhara, K. Kawamoto, K. Hirota, and B. Kusumoputro, “Improving recognition and generalization capability of back-propagation nn using a self-organized network inspired by immune algorithm (sonia),” *Applied Soft Computing*, vol. 6, no. 1, pp. 72–84, 2005.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*, 1986, vol. 1.

[24] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[25] C. Dunis and M. Williams, "Modelling and trading the eur/usd exchange rate: Do neural network models perform better?" *Derivatives use, trading and regulation*, vol. 8, no. 3, pp. 211–239, 2002.

[26] R. Ghazali, A. Hussain, and W. El-Deredy, "Application of ridge polynomial neural networks to financial time series prediction," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 913–920.