



## SYNTHESIS OF SELF-ORGANIZING MAP AND FEEDFORWARD NEURAL NETWORK FOR BETTER FORECASTING

Oles Hodych<sup>1</sup>, Yuriy Shcherbyna<sup>1</sup>, Michael Zylan<sup>2</sup>

1) Ivan Franko Lviv National University, Ukraine

2) Myklan Systems, Melbourne, Australia

**Abstract:** *In this article the authors propose an approach to forecasting the direction of the share price fluctuation, which is based on utilization of the Feedforward Neural Network in conjunction with Self-Organizing Map.*

*It is proposed to use the Self-Organizing Map for filtration of the share price data set, whereas the Feedforward Neural Network is used to forecast the direction of the share price fluctuation based on the filtered data set. The comparison results are presented for filtered and non-filtered share price data sets.*

**Keywords:** *artificial neural network; financial analysis; stock market; prediction.*

### 1. INTRODUCTION

The study of the share market is generally broken into two major approaches:

- ‘Fundamental Analysis’ the study, examination and analysis of the economy, industrial sector’s and a company’s business acumen, it’s strategy, management, product as well as it’s current and future financial status and a range of financial statistics available about the company;

- ‘Technical Analysis’ the study of past price movements and trading volume to forecast future price movements.

Both methods aim at providing answers to the successful timing of entry into or exiting from the market whilst trading stocks, indices, commodities or any other tradable instrument.

The study of Technical analysis provides the trader with a host of trading indicators. These indicators are normally based upon the past performance of a stock or commodity, which in turn is based on past investor attitudes and their psychology. Thus investors past trading habits are then revealed in charts of price movements and trading volume over specific time frames.

Price refers to any combination of the open, high, low or close for a given security over a specific timeframe. The time frame can be based on intraday, daily, weekly or monthly price data and last a few hours or many years.

The study of historical trading generates patterns and which may tend to repeat themselves. Technical analysis tools analyze this historical movement and try to determine the direction of the market and to pinpoint proper entry and exit points to maximize the potential return on a trade.

Not all technical signals and patterns work and each array of patterns on a chart together with a selection of indicators have their own set of rules to match. The rules are not steadfast and can be subject to other factors such as volume and momentum. What works for one particular stock may not work for another. Even though many principles of technical analysis are universal, each security will have its own idiosyncrasies.

In recent years an application of artificial neural networks gained a significant popularity in a wide variety of domains. The analysis of share market is one of them. In [1] the author provides a good overview of artificial neural networks studies for capital market applications. This overview covers the period from the very first applications of neural networks for the purpose of share market forecasting in 1988 up to year 2001.

Many studies address different aspects of the neural network application, which are specific to share markets. Some researches tackle several shares simultaneously to predict future returns, other trying to analyze shares individually. A combining approach is proposed in [2], where the prediction of future returns for different stocks or shares are viewed as different tasks, which share some parameters across stocks. This approach is known as multi-task learning. In [3] the authors provide a study of artificial neural networks for application in stock index forecasting. This paper discusses the utilization of the rescaled range analysis in conjunction with feed-forward neural networks in order to determine the relationship between the technical indicators and the levels of the index in the market under study over a period of time. The main

rational for this is that by purchasing the indexed stocks in the respective proportions a significant paper profit can be achieved applying different trading strategies. Some other studies [4][5] approach the analysis of share markets utilizing neural networks to model stock traders. During the training, neural networks are tuned for making traders more successful in selection of trading strategies.

In this paper the authors concentrate their research on the prediction of a single share. Having ability to successfully predict one share can be view as a component for the prediction of a whole share market sector which consists of multiple shares. The subject of the forecast is the movement of a share price rather than a price itself.

The main idea which is proposed here is the determination of principal components, or other words patterns, among the set of share price values captured over a period of time. The actual forecast of the share price movement is based on a set of all found principal components rather than a set of raw share prices.

## 2. SHARE PRICES DATA SET AND ITS INITIAL PROCESSING

Discussion and demonstration of the proposed technique utilizes a data set of the SRA share price. This data set consists of 2870 records of the daily share prices, which is more than 7 years of data. Each record of the data set has the structure represented in Table 1.

Let's denote the whole data set as  $X$  with the capacity  $|X| = N = 2870$ . Its elements  $x_i = (x_i^1, x_i^2, x_i^3, x_i^4, x_i^5) \in X$ ,  $i = \overline{1, N}$  are vectors, where  $x_i^1$  is Open Price,  $x_i^2$  is High Price,  $x_i^3$  is Low Price,  $x_i^4$  is Volume and  $x_i^5$  is Close Price. It is important to note that vectors  $x_i \in X$  are ordered chronologically.

The direction of the Close Price value fluctuation is the subject for prediction, which is based on the idea of determining dependencies between Cost Price and the rest of the components. Thus, by knowing today's Open Price, High Price, Low Price and Volume it should be possible to predict whether tomorrow's Close price would rise or fall.

In order to predict future fluctuation's direction, the value of  $x_i^5$  (Close Price) was shifted back for all

4. For each value  $x'_i$  in a data set calculate

$$x'_i = \frac{x'_i}{D}, i = \overline{1, N'} \text{ (covariance equalization).}$$

Formulae (2) and (3) for calculation of the standard deviation provide different results, thus constituting two different approaches to the data normalization. Let's denote these two approaches as **A1** and **A2** respectively.

of the  $x_i \in X$ . Thus producing a new data set  $X'$ , where  $|X'| = N - 1 = N' = 2869$  and its elements are constructed as  $x'_i = (x_i^1, x_i^2, x_i^3, x_i^4, x_{i+1}^5)$ ,  $i = \overline{1, N'}$ . This data manipulation ensures that the dependencies of the tomorrow's Close Price component are analyzed in respect to today's Open, High, Low Price and Volume components.

As can be observed from Table 1, Open Price, High Price, Low Price and Close Price components have similar statistical characteristics. On the contrary, the Volume component has much higher values. This leads to the need of initial data normalization in order to even components for better neural network learning. Later it will be shown how much of a difference makes a good initial normalization on the learning process and therefore on the forecasting capabilities of the trained neural network.

The general idea behind normalization is to normalize the data in such a way that mean is close to zero, or is small in comparison with standard deviation [6]. Three different approaches have been evaluated to perform normalization of share price data and are discussed below.

### Mean removal and covariance equalization.

This is one of the most common approaches to the data normalization [7]. Mean removal ensures that the mean of the normalized data is zero, and covariance equalization scales all values to a narrow interval. The following algorithm formalizes the normalization process:

1. Calculate the mean value  $\bar{x}'$  using formula (1):

$$\bar{x}' = \frac{1}{N'} \sum_i x'_i, \quad (1)$$

where  $N'$  is the cardinality of a data set  $X'$ ,  $x'_i \in X'$ ;

2. For each value  $x'_i \in X'$  calculate  $x'_i = x'_i - \bar{x}'$ ,  $i = \overline{1, N'}$  (mean removal);

3. Calculate standard deviation  $D$  using formula (2) or (3):

$$D = \max(x) - \min(x), \quad (2)$$

where  $\max(x)$  and  $\min(x)$  are respectively maximum and minimum values in data set  $X'$ .

$$D = \sqrt{\frac{1}{N'-1} \sum_i (x'_i - \bar{x}')^2}, \quad (3)$$

**Order of the maximum.** Another approach, which was evaluated, is based on the order of the maximum value in a data set  $X'$ . In this case, each value is divided by  $\Theta$ , which is calculated using formula (4):

$$\Theta = 10^{(order+1)}, \quad (4)$$

where  $order$  is the order of the  $\max(x)$ .

Formula (5) provides calculation of normalized values:

$$x'_i = \frac{x_i}{\Theta}, i = \overline{1, N'} \quad (5)$$

**Table 1. The structure of share price data set record**

Component	Description	Mean	Standard Deviation	Minimum	Maximum
<i>Open Price</i>	The price of the share at the moment of share market opening	1.02609927	0.83095612	0.09369999	3.40759993
<i>High Price</i>	The highest price of the share during the day	1.03939589	0.85133389	0.09799999	7.05369997
<i>Low Price</i>	The lowest price of the share during the day	1.01336197	0.82255913	0.08940000	3.39059997
<i>Volume</i>	Quantity of shares sold during the day	457,827.98	1,753,117.5	258	37,516,456
<i>Close Price</i>	The price of the share at the moment of share market closing	1.02434511	0.82966699	0.09369999	3.39059997

Let's denote this approach as **A3**. Statistical characteristics of the normalized data set  $X'$  for approaches A1, A2 and A3 are presented in Table 2.

As can be observed from Table 2, the presented statistical measurements for all five data components belong to the interval  $[-1, 1]$  for approaches A1, A2 and A3 with one exception – component Volume has a large maximum value when A2 is used. The following section discusses the learning process of the Self-Organizing Maps (SOM), Feedforward Neural Network (FNN) and data filtration.

### 3. SOM AND FNN TRAINING, DATA FILTRATION

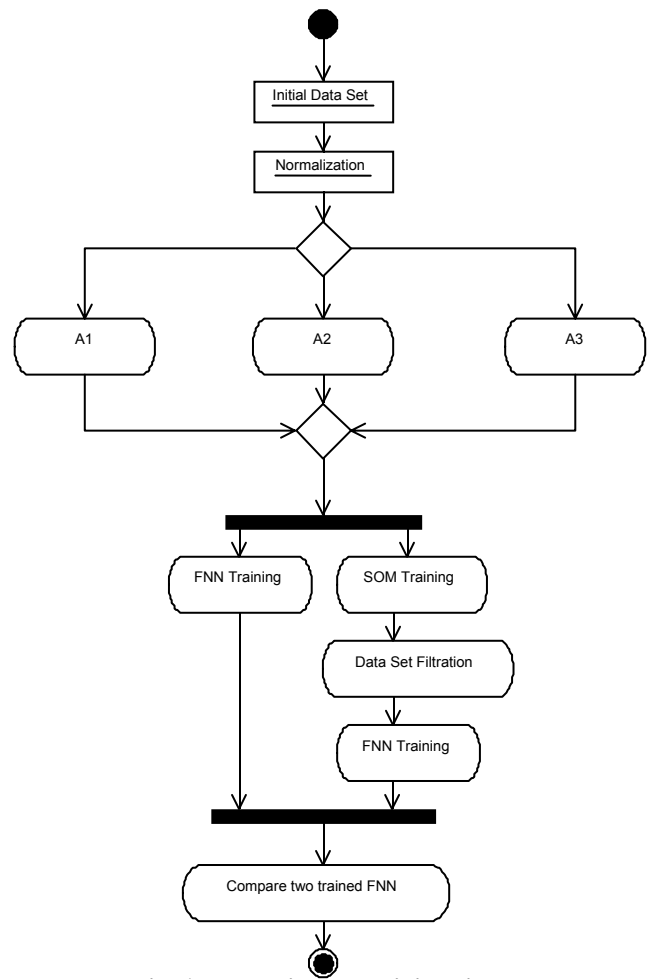
Discussion of the SOM, FNN training and data filtration is based on three experiments, which are schematically illustrated in Fig 1. The most significant difference between these experiments is the normalization approach chosen for each of them. Let's denote experiments for data set normalized with approaches A1, A2 and A3 as Experiment 1, Experiment 2 and Experiment 3 respectively.

The activity diagram of the experiments can be summarized as follows. First, the initial data set is normalized utilizing one of the normalization approaches (A1, A2 or A3). Then, two feedforward neural networks are trained. One FNN is trained on a normalized data set, whereas another FNN is trained on a normalized and filtered data set. Finally, the two training processes for the two FNNs are compared.

**Training Data.** For neural network training the original data set  $X'$  was split into two subsets:  $X'_{tr}$  for training and  $X'_{ts}$  for testing. Subset  $X'_{tr}$  contained the first 1000 data entries corresponding to approximately the first 3 years, where is  $X'_{ts}$  contained the rest of the original data set.

**SOM training.** Filtration plays a significant role in the data analysis presented here. Its core purpose is to identify principal components in the data set of share prices, thus limiting the level of noise and improving FNN abilities for prediction. For the purpose of filtration, a SOM neural network was used. Training of the SOM was performed using the Kohonen algorithm [7]. The SOM, which was

subsequently trained, consisted of a 2D lattice with neighbouring neurons located in accordance with a



**Fig. 1 - Experiment activity diagram**

cross pattern. A fragment of such a network is depicted in Fig 2, with the neighbourhood pattern highlighted.

The lattice was composed using a method where new neurons are added outward and grouped in levels. In the case of the cross neighbourhood pattern, level 0 consists of one neuron with coordinates (0, 0). Level 1 is built by ensuring (and where necessary introducing) that the neuron at Level 0 has four neighbours. All subsequent levels are built in the same manner forming the structure illustrated in Fig 2. The SOM that was utilized had 7 levels, resulting in 113 neurons in total.

A more detailed description of the algorithm for SOM composition is outside the scope of this article.

**Table 2. Statistical measurements for the normalized data set**

	Mean			Standard Deviation			Minimum			Maximum		
	A1	A2	A3	A1	A2	A3	A1	A2	A3	A1	A2	A3
Open	3.936E-10	3.633E-11	1.0261	0.2507	0.6293	0.8310	-0.2815	-0.7061	0.0937	0.7186	1.8035	3.4076
High	-5.946E-11	-1.945E-10	1.0394	0.1224	0.6336	0.8513	-0.1353	-0.7006	0.0980	0.8647	4.4760	7.0537
Low	9.623E-11	2.107E-10	1.0134	0.2492	0.6302	0.8226	-0.2799	-0.7078	0.0894	0.7201	1.8212	3.3906
Volume	1.070E-11	2.150E-10	0.0458	4.673E-2	0.9675	0.1753	-0.0122	-0.2525	0.258E-4	0.9878	20.4525	3.7516
Close	-6.978E-11	-4.462E-10	1.0243	0.2517	0.6293	0.8297	-0.2823	-0.7059	0.0937	0.7177	1.7949	3.3906

Parameters of the Kohonen algorithm for SOM training are given in Table 3. The names of the parameters are consistent with those used in [7]:  $\delta_0$  and  $\tau_1$  are used in the calculation of the neighbourhood function;  $\eta_0$  and  $\tau_2$  are used in the calculation of the learning rate;  $n$  is the number of iterations.

**Table 3. Kohonen algorithm parameters**

$\delta_0$	$\tau_1$	$\eta_0$	$\tau_2$	$n$
7	3692	0.1	10000	10000

The SOM learning process is depicted in Fig 3, where three graphs are built for accumulated mean square error, which was calculated during the learning process on each iteration.

The mean square error is calculated using the following formula [9]:

$$MSE = \frac{1}{N_r'} \sum_{i=1}^{N_r'} \|x_i' - w_c\|^2,$$

where  $x_i' \in X_r'$ ,  $N_r' = |X_r'|$  and  $w_c$  is the best matching neuron for  $x_i'$ .

The value of MSE measures the ability of a SOM neural network to approximate a data set. As can be observed from Fig 3 the worst result was obtained for Experiment 2, and the best result was achieved in Experiment 3. The final mean square errors are given in Table 4.

**Table 4. SOM learning performance (MSE)**

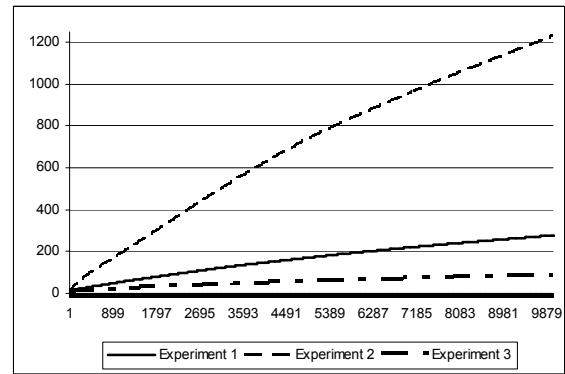
Experiment 1 (SOM <sub>A1</sub> )	Experiment 2 (SOM <sub>A2</sub> )	Experiment 3 (SOM <sub>A3</sub> )
0.018155052	0.088601807	0.005241772

Data filtration is performed using trained SOM. The basic idea for data filtration is to substitute each vector in the data set  $X'$  with a weight vector of the corresponding best matching neuron from the trained SOM, thus producing a filtered data set  $X^F$ . The following algorithm is proposed for building a filtered data set:

1. Initialize index  $i \leftarrow 1$ ; filtered data set  $X^F$  is empty at the beginning;
2. For  $x_i' \in X'$  find the best matching neuron in the trained SOM. Let it be  $w_c$ .

3. Add a weight vector of the neuron  $w_c$  to the data set  $X^F$ ;
4. Increment  $i$  by one:  $i \leftarrow i + 1$ . If  $i < N'$  where is  $N' = |X'|$  then return to step 2, else go to step 5;
5. Finish: filtered data set  $X^F$  is composed,  $|X^F| = |X'|$ .

Please note that the proposed algorithm can be applied to any data set, which requires filtration.



**Fig. 3 - SOM learning process for all three experiments**

**FNN training.** The feedforward neural network was trained using the back-propagation algorithm [7]. It is a supervised learning, therefore the training data set  $X_r'$  was used to compose a data set of training pairs {input; expected output}:  $\{(x_i^1, x_i^2, x_i^3, x_i^4); x_i^5\}, i = \overline{1, N_r'}$ . The input layer of the FNN had four neurons and the output layer had one neuron. Activation functions for neurons in calculating layers of the FNN were chosen as  $\sin(x)$  and  $\cos(x)$ , which were alternating between the neurons [8].

One of the main difficulties with the application of this algorithm is the selection of a learning rate. It was proposed to use a learning rate calculated by formula (6):

$$\alpha = \alpha_0 \times \exp\left(-\frac{k \times i}{l}\right), \quad (6)$$

where  $\alpha_0$  is the initial leaning rate,  $i$  – current iteration, parameters  $k$  and  $l$  allow controlling the rate of decay. In the experiments that were performed, these parameters were varied in order to achieve a better result in each experiment (see Table 5).

**Table 5. Values of parameters for learning rate in the back-propagation algorithm**

Param.	Experiment 1	Experiment 2	Experiment 3
$\alpha_0$	0.1	1.0	0.1
$k$	2	2	2
$l$	1000	1000	10000

**Table 6. FNN learning performance (mean square error)**

Experiment 1		Experiment 2		Experiment 3	
Filtered Data ( $FNN_{A1}^F$ )	Non-filtered Data ( $FNN_{A1}$ )	Filtered Data ( $FNN_{A2}^F$ )	Non-filtered Data ( $FNN_{A2}$ )	Filtered Data ( $FNN_{A3}^F$ )	Non-filtered Data ( $FNN_{A3}$ )
4.1369494E-4	0.0026272	0.1064891	0.1127779	9.8653348E-5	9.9157672E-5

The learning process of FNN can be illustrated in a similar manner as was done for SOM learning (see fig. 3) by graphing a chart of the accumulated mean square errors, which were calculated during the learning process for the whole training data set. Fig 4 illustrates three charts of the accumulated mean square errors for three experiments: 4(a) represents Experiment 1, 4(b) – Experiment 2 and 4(c) – Experiment 3. Each of those figures contains two charts reflecting the learning process of FNN on filtered and non-filtered data sets.

As can be observed, for all three experiments the learning process performed better on the filtered data set, thus producing a more reliable neural network. The worst result was obtained for Experiment 2, where approach A2 was used for data normalization.

The last figure 4(d) represents the combined comparison between the best results for Experiment 1 and Experiment 3, clearly indicating more smooth learning process for Experiment 3. Please note that due to the very high values of the mean square error produced by Experiment 2, its chart was excluded from figure 4(d) in order not to suppress two other charts. The final values of the mean square error

produced by trained the FNNs are presented in Table 6.

The next section discusses the evaluation of the forecasting capabilities of the trained feedforward neural networks.

#### 4. EVALUATION OF THE FORECASTING CAPABILITIES AND FORECASTING

**Evaluation.** All six FNNs, who’s learning performance is presented in Table 6, were evaluated for forecasting capabilities. Both filtered and non-filtered data sets were used resulting in twelve measurements of the forecasting capabilities (two for each FNN). Testing data set  $X'_s$  was normalized using approaches A1, A2 and A3 producing data sets  $X_{A1}$ ,  $X_{A2}$  and  $X_{A3}$  respectively. Each of those data sets was filtered using the proposed filtration algorithm with an appropriately trained SOM ( $SOM_{A1}$ ,  $SOM_{A2}$  and  $SOM_{A3}$ ). The filtered data sets produced are denoted as  $X_{A1}^F$ ,  $X_{A2}^F$  and  $X_{A3}^F$ . The final evaluation schema is presented in Table 7.

**Table 7. Evaluation schema.**

Derived from	Experiment 1				Experiment 2				Experiment 3			
Evaluated FNN	$FNN_{A1}^F$		$FNN_{A1}$		$FNN_{A2}^F$		$FNN_{A2}$		$FNN_{A3}^F$		$FNN_{A3}$	
Evaluation Data Set	$X_{A1}^F$	$X_{A1}$	$X_{A1}^F$	$X_{A1}$	$X_{A2}^F$	$X_{A2}$	$X_{A2}^F$	$X_{A2}$	$X_{A3}^F$	$X_{A3}$	$X_{A3}^F$	$X_{A3}$

The basic idea behind the evaluation algorithm is the comparison between actual fluctuation and forecasted fluctuation. For the purpose of generalization, let’s denote a data set which is used for evaluation as  $Z$ . Thus, by substituting  $Z$  with one of the data sets from Table 7 we can perform twelve evaluations. Similarly as was done before, data set  $Z$  can be split into pairs {input; expected output}:  $\{(z_i^1, z_i^2, z_i^3, z_i^4); z_i^5\}, i = \overline{1, |Z|}$ .

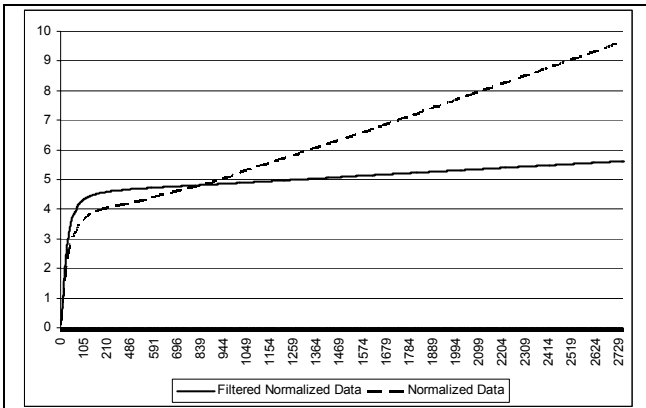
Let’s denote the output value of the trained FNN as  $FNN(\tilde{z}_i)$  for the input vector  $\tilde{z}_i = (z_i^1, z_i^2, z_i^3, z_i^4)$ . Actual fluctuation is calculated using formula (7) and forecasted fluctuation using formula (8).

$$\Delta_i = z_i^5 - z_{i-1}^5 \tag{7}$$

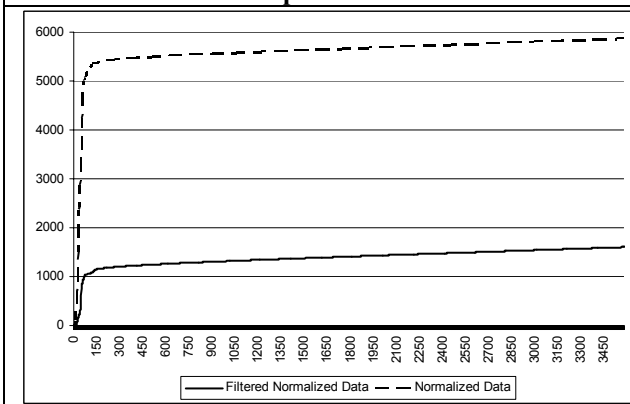
$$\tilde{\Delta}_i = FNN(\tilde{z}_i) - FNN(\tilde{z}_{i-1}) \tag{8}$$

It is important to note that  $z_i^5$  corresponds to the Close Price and that this component was shifted backward during the initial data transformation (refer to the first section). Therefore, if elements  $(z_i^1, z_i^2, z_i^3, z_i^4)$  represent today’s data then  $z_i^5$  represents tomorrow’s Close Price. Thus to predict future fluctuation it is necessary to subtract  $z_{i-1}^5$ ,

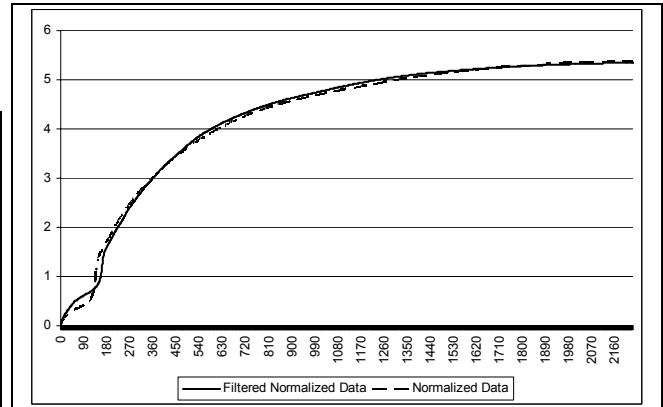
which is today's Close Price, from  $z_i^5$ . Hence formulae (7) and (8). The sign of the fluctuation indicates its direction.



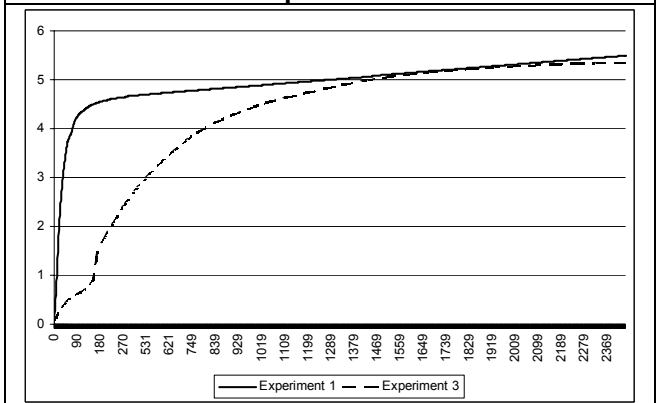
(a) Mean square error for FNN learning in Experiment 1



(b) Mean square error for FNN learning in Experiment 2



(c) Mean square error for FNN learning in Experiment 3



(d) The best results of Experiment 1 and Experiment 3

Fig. 4 - Accumulated mean square errors

Using the introduced nomination the following evaluation algorithm was proposed:

1. Initialize index  $i \leftarrow 2$ ; Initialize counter for the number of successful predictions  $k \leftarrow 0$ ;
2. Calculate  $\Delta_i$  and  $\tilde{\Delta}_i$  using formulae (7) and (8) respectively;
3. If  $\text{sign}(\Delta_i) = \text{sign}(\tilde{\Delta}_i)$  then it is a successful prediction and thus  $k \leftarrow k + 1$ ;
4. Increment index  $i \leftarrow i + 1$ ;
5. If  $i < |Z|$  then return to step 2, otherwise go to step 6;

Calculate percentage of successful predictions:

$$P \leftarrow \frac{k}{|Z|-1} \times 100\%$$

Please note, due to the fact that

calculation of fluctuations  $\Delta_i$  and  $\tilde{\Delta}_i$  requires index  $i$  to start with the value 2, the total number of predictions for all elements in data set  $Z$  is  $|Z|-1$ . Thus, in the calculation of the percentage of successful predictions, the value  $|Z|-1$  is used, but not  $|Z|$ .

As can be observed, for all three experiments the learning process performed better on the filtered data set, thus producing a more reliable neural network.

The worst result was obtained for Experiment 2, where approach A2 was used for data normalization. Evaluation results for all trained FNNs were obtained using the above algorithm and presented in Table 8. As can be observed from Table 8 the best forecasting FNNs' capabilities are revealed when filtered data sets are used. This is relevant for all trained feedforward neural networks.

An interesting result is produced for FNNs, which were trained on non-filtered data sets. Their forecasting capabilities are insignificantly different to those of FNNs trained on filtered data sets. Specifically,  $FNN_{A1}^F(X_{A1}^F)$  had only 0.38% better result than  $FNN_{A1}(X_{A1}^F)$ ;  $FNN_{A2}^F(X_{A2}^F)$  had only 0.28% worse result than  $FNN_{A2}(X_{A2}^F)$ ;  $FNN_{A3}^F(X_{A3}^F)$  had the same result as  $FNN_{A3}(X_{A3}^F)$ . This indicates that the filtering of training data sets did not have a significant impact on the training of FNNs in respect to their forecasting capabilities.

Comparison of the evaluation results in Table 8 pertaining to non-filtered and filtered data sets shows a significant improvement of the forecast

when filtration is applied to a data set. In all cases filtration led to more than double improvement. The best forecasting capabilities were revealed by the  $FNN_{A3}^F$  and  $FNN_{A3}$  neural networks, which were trained on a data set normalized with approach A3.

**Forecasting.** The presented evaluation procedure is a batch procedure, which utilizes all elements in a testing data set. The main practical use of this

procedure it to ensure that the trained neural network is reliable and can be used for prediction. Forecasting of the direction of the future price fluctuation can be performed similarly to evaluation, but for the last element in a data set. There are two steps involved in performing the forecast: data set preparation and the calculation of the future fluctuation.

**Table 8. Evaluation results**

<i>FNN</i>	$FNN_{A1}^F$		$FNN_{A1}$		$FNN_{A2}^F$		$FNN_{A2}$		$FNN_{A3}^F$		$FNN_{A3}$	
<b>Data Set</b>	$X_{A1}^F$	$X_{A1}$	$X_{A1}^F$	$X_{A1}$	$X_{A2}^F$	$X_{A2}$	$X_{A2}^F$	$X_{A2}$	$X_{A3}^F$	$X_{A3}$	$X_{A3}^F$	$X_{A3}$
<b>%</b>	88.82	37.45	88.44	37.03	80.95	35.31	81.23	36.92	92.88	37.40	92.88	37.35

Data set preparation consists of updating the data set with the latest information. Based on the previously established model, the latest information means a vector  $x'_{N'+1} = (x'_{N'+1,1}, x'_{N'+1,2}, x'_{N'+1,3}, x'_{N'+1,5}, ?) \in X'$  with today's Open Price, High Price, Low Price, Volume and unknown Close Price. Today's Close Price becomes a value for  $x'_{N'}$  in vector  $x'_{N'} \in X'$ . Because normalization is applied to the data set  $X'$  it is necessary to assign the mean value of the Close Price component in  $X'$  to the unknown value  $x'_{N'+1,5}$ .

Thus,  $x'_{N'+1} = (x'_{N'+1,1}, x'_{N'+1,2}, x'_{N'+1,3}, x'_{N'+1,5}, \overline{x'_{N'+1,5}})$ .

Calculation of the future fluctuation requires application of formula (8) to vectors  $x'_{N'}$  and  $x'_{N'+1}$ :  $\tilde{\Delta}_{N'+1} = FNN(x'_{N'+1}) - FNN(x'_{N'})$ . The sign of the calculated fluctuation  $\tilde{\Delta}_{N'+1}$  indicates the direction of the fluctuation: if  $\text{sign}(\tilde{\Delta}_{N'+1}) = 1$  then tomorrow's Close Price should be higher than today's Close Price; if  $\text{sign}(\tilde{\Delta}_{N'+1}) = -1$  then tomorrow's Close Price should be lower than today's Close Price; if  $\text{sign}(\tilde{\Delta}_{N'+1}) = 0$  then tomorrow's Close Price should be the same as today's Close Price.

The probability of a successful forecast can be calculated using the proposed evaluation algorithm. In the case where the probability is too low it is necessary to retrain the feedforward neural network on a data set with the latest updated information.

**5. CONCLUSION AND FUTURE RESEARCH**

In this paper we propose an approach to forecasting the direction of the data fluctuations, which encompasses the synthetic utilization of Feedforward Neural Networks and Self-Organized Maps. The essence of this approach is to identify principle components of the underlying data set in order to build a more reliable data model of the process under analysis. The set of principle components is subsequently used for actual forecasting of the data fluctuations' directions.

It was proposed to use a Self-Organized Map for the purpose of finding the principle components. A

filtration algorithm was developed to filter a data set by substituting its elements with the corresponding principle components. The filtered data set was used for two purposes: Feedforward Neural Network training and forecasting. It wasn't clearly determined whether the training process of Feedforward Neural Network improves when a filtered training data set is used. However, the use of a filtered data set for the purpose of forecasting indicated more that double improvement of performance when compared with a non-filtered data set. Additionally, special attention was paid to the normalization of the original data set, and the influence of the three different normalization approaches was analyzed.

The proposed forecasting approach was evaluated based on the data set of SRA share prices, which represents data gathered on a daily basis during a period of more than seven years.

The future research will be extended in two stages. In recognition of the fact that it is not practical to use only one share in a trade, the first stage will apply the proposed technique to analyze a share market sector which consists of several shares. A special attention will be paid to the cross correlation between the shares in order to determine the influence of the price movement between the shares.

In the second stage an expert system should be developed, which would take an advantage of the share price movement forecast in conjunction with decision trees. Decision threes in particular will be used to build a set of decision making rules based on the predicted share price movement in order to help making a decision for buying and selling of shares to achieve a maximum profit.

**6. REFERENCES**

1. Arnold F. Shapiro, "Capital Market Applications of Neural Networks, Fuzzy Logic and Genetic Algorithms", [www.afir2003.nl/download/AFIR%202003%20wc%20pap%20def%20shapiro.PDF](http://www.afir2003.nl/download/AFIR%202003%20wc%20pap%20def%20shapiro.PDF)
2. Joumana Ghosn, Yoshua Bengio, Multi-"Task Learning for Stock Selection",

[www.iro.umontreal.ca/~lisa/bib/pub\\_subject/multitask/pointeurs/multitask-nips97.ps](http://www.iro.umontreal.ca/~lisa/bib/pub_subject/multitask/pointeurs/multitask-nips97.ps)

3. Jingtao Yao, Chew Lim Tan, Hean-Lee Poh, "Neural Networks for Technical Analysis: A Study on KLCI", International Journal of Theoretical and Applied Finance, Vol. 2, No. 2, pp. 221-241, 1999.

4. Graham Kendall, Yan Su, "The Co-evolution of Trading Strategies in A Multi-agent, Based Simulated Stock Market Through the Integration of Individual Learning and Social Learning",

[www.cs.nott.ac.uk/~yxs/papers/icmla03.pdf](http://www.cs.nott.ac.uk/~yxs/papers/icmla03.pdf)

5. Jing Yang, "The Efficiency of an Artificial Double Auction, Stock Market with Neural Learning Agents", [www.essex.ac.uk/ccfea/Seminarpapers/doubleauctionsimulation.pdf](http://www.essex.ac.uk/ccfea/Seminarpapers/doubleauctionsimulation.pdf)

6. Le Cun, Y, "Efficient learning and Second-order Methods", A Tutorial at NIPS 93, Denver, 1993.

7. Simon Haykin, "Neural Networks: A comprehensive foundation", Prentice-Hall, New Jersey, 1999.

8. I. Nikolski, O. Hodych, Yu. Shcherbyna, "Application of artificial neural networks to the problem of time series forecasting. International Conference On Inductive Modelling ICIM – 2002, pp. 144-149.

9. Si J., Lin S., Vuong M.-A, "Dynamic Topology Representing Networks", Neural Networks, No. 13, pp. 617-627, 2000.



**Michael Zylan** is a director of Myklan Systems, a small private investment company in Australia. Myklan Systems has been active in the Australian Stock Market since 1986 and in the DAX and FTSE commodity markets since 1999. The trading approach is utilizing a Technical Analysis

approach utilizing Meta Stock and its inbuilt System Explorations.



**Oles Hodych** received the M.S. degree in mathematics from Ivan Franko Lviv National University, Ukraine, in 2001.

His current research lies in the domain of data mining and knowledge discovery with application of artificial neural networks, interval analysis and genetic algorithms.

The areas of his interest are parallel computation, grid computing, cluster analysis and pattern recognition.

**Yuri Shcherbyna** received his M.S. and Ph.D. degree in mathematics from Ivan Franko Lviv National University (Ukraine) in 1971 and 1975 respectively. He is the author of more than 80 scientific publications including text books.



In present time he is a senior lecturer in discreet analysis and artificial intelligence at Ivan Franko Lviv National University. His current interest includes non-linear and discreet optimization, decision making and intellectual systems.