Financial Indices Modelling and Trading utilizing Deep Learning Techniques

The ATHENS SE FTSE/ASE Large Cap Use Case

Marios Mourelatos
Pattern Recognition Laboratory / Dept. Computer Engineering and Informatics
University of Patras
Patras,Greece
mourelatos@ceid.upatras.gr

Thomas Amorgianiotis
Pattern Recognition Laboratory / Dept. Computer Engineering and Informatics
University of Patras
Patras,Greece
amorgianio@ceid.upatras.gr

Christos Alexakos
Pattern Recognition Laboratory / Dept. Computer Engineering and Informatics
University of Patras
Patras,Greece
alexakos@ceid.upatras.gr

Spiridon Likothanassis
Pattern Recognition Laboratory / Dept. Computer Engineering and Informatics
University of Patras
Patras,Greece
likothan@ceid.upatras.gr

***Abstract*— Prediction and modelling of the financial indices is a very challenging and demanding problem because its dynamic, noisy and multivariate nature. Modern approaches have also to challenge the fact that they are dependencies between different global financial indices. All this complexity in combination with the large volume of historic financial data raised the need for advanced machine learning solutions to the problem. This article proposes a Deep Learning approach utilizing Long Short-Term Memory (LSTM) Networks for the modelling and trading of financial indices. The technique is evaluated in the use case of the Athens SE FTSE/ASE Large Cap Index in comparison with a hybrid approach combining Genetic Algorithms and Support Vector Machines with promising results.**

Keywords— financial indices modelling, financial indices prediction, deep learning, big data, Long Short-Term Memory Recurrent Neural Networks

I. INTRODUCTION

Prediction and modelling of the financial indices is a very challenging and demanding problem cause its complex, dynamic, noisy and multivariate nature. As long as the data analysis technologies evolve, new approaches are trying to give answers to this problem, but most of them are just minor improvements of older solutions. The FTSE/ASE Large Cap Index – also known as FTSE/ASE 20 Index - consists of 25 of the largest and most liquid stocks that trade on the Athens Stock Exchange (Greece). It was developed in September 1997 out of a partnership between the Athens Stock Exchange and FTSE International. Last crisis period raised lot of failures on the application of traditional economic prediction methods and machine learning ones in financial time series while trying to capture the complexity and the nonlinearities that exist in such time series data. From these failures, certain disadvantages have been identified such us the difficulties in tuning the parameters of the algorithm, the overfitting problem, the fact that modelling and trading are most of the times considered as different problems and the disability of linear methods to provide good prediction results. The identification of these problems leads to the proposal of several machine learning methods which despite their encouraging results, they are leaving an open window for further performance enchantment [1].

Many researchers developed various intelligent methods for modelling and trading of the financial indices. Most common approaches are based on simple tradition AI methods, such as ARMA [2], or on machine learning techniques. The second group includes applications using either simple neural network techniques, such as Higher Order Neural Networks (HONN) [3] or more complex and hybrid approaches, such as Artificial Bee Colony Algorithm with Recurrent Neural Networks [4] and Genetic Algorithms with Support Vector Machines (SVM) [5].

A common aspect of the aforementioned techniques is that they deal with the each stock index as being independent and cut off from the global stock market. Modern research, however, shows that this approach is not completely true and that they are dependencies between different financial indices [6, 7, 8]. The major problem lies in the fact that these dependencies have not yet been studied thoroughly. Thus, in the current approach, several inputs from other financial indices, such as FTSE100, DJIA, GDAX, utilized alongside with the traditional autoregressive and technical indicator inputs for the optimal prediction and trading results using the FTSE/ASE 20 index.

978-1-5386-5150-6/18/$31.00 ©2018 IEEE

The most of the financial indices in world market exist for over 20 years with available history data. Taking account also the fact that the size of the universe of financial indices which could possibly be dependent with FTSE/ASE Large Cap is so high, the size of data that must be analysed makes this problem relevant to the "big data" [9].

The combination of the large number of inputs, the nonlinearity of the problem and the large size of data that must be examined, require modern machine learning approaches. In the current article, we are focusing on the application of Deep Learning techniques for the prediction of a stock in FTSE/ASE Large Cap Index. Deep Learning techniques are very popular in bid data prediction problems with multiple inputs. More specifically, Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) were used in order to train the prediction model. In order to achieve a small scale evaluation of the proposed approach experiments have been contacted for the prediction of the stock of the National Bank of Greece using a time window of 5 days (a week) of past data. For the purpose of evaluation, comparison experiments have been conducted against an alternation of ESVM Stock Predictor with data collected for 5 years.

The proposed approach is presented in detail in the rest of the paper. In Section II a literature review regarding the machine learning approaches in stock exchange prediction problem is presented along with the description of the benchmarking algorithm that will be used for evaluating the proposed LSTM network approach. Section III describes the financial data that was used. Section IV presents the LSTM network approach for training the prediction model while in section V depicts the experimental process for the evaluation of the proposed approach. Finally, Section VI concludes the article and presents future work.

II. Related Work And Benchmarking

The complex and uncertain nature of the modelling and trading financial indexes problem forced the replacement of the traditional statistical model with prediction models extracted by artificial and computational intelligence algorithms. Especially, Artificial Neural Networks is one of the most common technique in the modern solutions. Variations such as Multilayer Perceptron Neural Networks, Higher Order Neural Networks and Recurrent Neural Networks [10, 11] are some paradigms of Artificial Neural Networks utilization for predicting financial indexes. Nevertheless, the large number of inputs on such Neural Networks approaches raise the problem of overfitting which restricts the performance of these solutions. These constraints led the researchers to the utilization of methods that include Support Vector Machine (SVM). SVM and their regression extension (Support Vector Regression) are extensively applied in modern modelling and trading financial indexes omit of promising results [12, 13]. One promising approach for market stock prediction in FTSE/ ASE20 is presented in [14], where the authors introduced the ESVM Stock Predictor, an evolutionary approach deploying genetic algorithms to tune the parameters of a Support Vector Machine Model and select its inputs optimally in order to achieve the highest statistical and trading performance while keeping the final model simple to avoid overfitting. The ESVM Stock Predictor is benchmarked with five traditional methods (Naïve Strategy, Buy and Hold Strategy, MACD strategy, ARMA plus a Multilayer Perceptron neural network) and the results obtained seem very promising in terms of both statistical and trading metrics.

Thus, an alternative implementation of ESVM Stock Predictor was used for the comparison with the proposed LSTM network approach in the experimental validation process. On the same concept the benchmarking GA-SVR algorithm combines effectively the Genetic Algorithms with Support Vector Regression (SVR), a subcase of SVM, which is trying to predict with the higher possible precision the dependent variables of a function in correspondence with the independent variables. During the implementation of GA-SVR, the GA chooses the dependent variables that affect the prediction of the independent variable. Furthermore, through the procedure, the best values for the parameter of normalization and (the parameter of the Gaussian kernel function) for the SVM algorithm are selected. Optimization of those two parameters is significantly important, as parameter defines the balance between complexity and performance and on the other hand, because of the non-linearity of the data a properly tuned kernel function is of high importance. An objective function is used for the implementation of the GA optimizing the suitable metrics for the evaluation of investing strategies (annualized return for instance).

III. Collected Financial Data

The dataset deployed in the present study incorporates information from 5 financial indexes, one exchange rate, 5 stock prices and gold price.

A. Under Study Stock

In the present paper the stock of National Bank of Greece (ETE) was studied (Fig. 1).

The ETE stock daily time series is non-normal (Jarque-Bera statistics confirms this at the 99% confidence interval), containing slight skewness and high kurtosis. In the present study arithmetic returns were used and they are estimated using the following procedure: Given the price level P1, P2,…,Pt, the arithmetic return at time t is formed by the eq. 1:

$$R_t = \frac{(P_t - P_{t-1})}{P_{t-1}} \qquad (eq.1)$$

Data is collected from Yahoo Finance website[1], exchange rate was downloaded from the Global-View Forex Forum[2] and

[1] https://finance.yahoo.com/
[2] http://www.global-view.com/

gold price was retrieved from Investing.com website[3]. It consists of 5 years (22/12/2009 until 11/12/2014).

Fig. 1. ETE Stock daily time series

B. *Additional Input Financial Data*

As mentioned before, apart from the under study stock index, as inputs, additional financial data is used. ETE, OPAP, PEIR, OTE and INLOT are stocks that belongs to Athens Stock Exchange. ETE, OPAP, PEIR, OTE also belongs to FTSE/ASE Large Cap. FTSE100 consists of the 100 companies listed on the London Stock Exchange with the highest market capitalization. DJIA is a stock market index that consists of 30 large publicly owned companies based in the United States. GDAX consists of the 30 major German companies trading on the Frankfurt Stock Exchange. The Nikkei is a stock market index for the Tokyo Stock Exchange (TSE) and consists of the 225 larger companies.

Forecasting and trading the ETE was attempted using an extended universe of inputs (16 inputs) containing autoregressive inputs. This dataset includes inputs and from a variety of financial time-series including ETE, OPAP, PEIR, OTE, INLOT, FTSE/ASE 20, FTSE100, DJIA, GDAX, NIKKEI 225, EUR/USD EXCHANGE RATE and GOLD PRICES. ETE index can also be dependent on an enormous number of indices. The data consists of 5 years. Each financial year is 252 days. So, we have 5*252 = 1260 rows of data. We calculated 16 inputs and 1 output. Total dataset elements are 1260*17 = 21420 elements. From this scope the problem of modelling and forecasting the ETE index from this scope of view can be seen as a big data problem. The weights are trained in order to find the dependencies between indices and eliminate indexes that have no dependencies. This is the way we deal with big data problem.

TABLE I. INPUT TO ALGORITHM

#	Variable	Description
1	DayLast(ETE)	ETE stock return
2	DayHigh(ETE)	ETE higher intraday stock return
3	DayLow(ETE)	ETE lower intraday stock return
4	DayVolume(ETE)	ETE day volume
5	DayLast ADJ(ETE)	ETE Adj Close
6	DayLast(OPAP)	OPAP stock return
7	DayLast(NLOT)	NLOT stock return
8	DayLast(OTE)	OTE stock return
9	DayLast(PEIR)	PEIR stock return
10	DayLast(FTSE/Athex Large Cap)	FTSE/Athex Large Cap index return
11	DayLast(FTSE100)	FTSE100 index return
12	DayLast(DJIA)	DJIA index return
13	DayLast(GDAX)	GDAX index return
14	DayLast(NIKKEI 225)	NIKKEI 225 index return
15	EUR/USD EXCHANGE RATE	EXCHANGE RATE
16	GOLD PRICES	GOLD PRICES

[3] https://gr.investing.com/

We trained our model with 1,2..10 day/s and predicted the next 1 day. The best results was for 4 days data and predicted the 5 day. The education schema may be related to the fact that a financial week is 5 days. We used total 252 days of data for training and the model occurred predicted the next day or days according the test case. The number of 252 days corresponds to the numbers of days that the Athens Stock Exchange operates normally within a year. It was used sliding window approach. This means that for the prediction of the second day we used 2-253 days and we predicted the 254 day. The process is repeated until we reach the last window.

We used different testing cases in order to find the best forecasting model. Test cases included 252 training days or 504 and sliding windows of1, 2, 5 and 10.

IV. Forecasting Model

A. *Recurrent Neural Networks with LSTM units*

The proposed forecasting model is based on Recurrent Neural Networks composed of LSTM units [15], also known as LSTM networks, which widely used for processing and prediction time series given time lags of unknown size and duration between important events. Recurrent Neural Networks [16] are based on the same principles as those behind Feedforward Neural Networks (FFNNs). The two main differences between FFNNs and RNNs are that RNNs uses sequences as inputs in the training phase, and memory elements. Memory is defined as the output of hidden layer neurons, which will serve as additional input to the network during next training step. In FFNN the output at any time t, is a function of the current input and the weights. This can be easily expressed using the following equation (eq.2):

$$\bar{y}_t = F(\bar{x}_t, W) \quad (eq.2)$$

In RNNs, the output at time t, depends not only on the current input and the weight, but also on previous inputs. In this case the output at time t will be defined as:

$$\bar{y}_t = F(\bar{x}_t, \bar{x}_{t-1}, \bar{x}_{t-2}, \cdots, \bar{x}_{t-t_0}, W) \quad (eq.3)$$

Backpropagation is also used for RNNs training, but with a conceptual difference. The process is similar to the one in the FFNN, with the exception of considering previous time steps, as the system has memory. This process is called Backpropagation Through Time (BPTT). The LSTM cell allows a recurrent system to learn over many time steps without the fear of losing information due to the vanishing gradient problem. It is fully differentiable, therefore gives us the option of easily using backpropagation when updating the weights.

B. LSTM Network Structure

The network structure was determined after the execution of a series of experiments in order to find the best fit parameters. The best number of hidden layers of the network is 64. For the memory, the best fit was to have as input the last five values in the time series. This denotes that for input we have the last five days for each one of the 16 input parameters of Table I. The time series is normalized and then separated in 1553 windows of 5 days each. The model has as an input the first 4 rows of each window and predicts from the 5th row of the window the closing price. The method is purely real time supervised learning in which the in which the initial time set is a stock year of 252 days and the test set is the next day. The training starts from the first 252 windows, for each window we train the model to predict the 5th day based on the previous 4.

Finally the model makes a prediction for the 253 window which has never seen before in the training procedure and the result is been collected for evaluation. Next the model is getting trained on the windows from 2 to 253 and predicts the 254th. The procedure is been repeated until the end of all the training set windows. The procedure is been repeated until the last window, and the whole of predictions is been compared with the true values.

The implementation of LSTM Network was done with the TensorFlow library [17]. The TensorFlow is an open source software library for learning machines in a series of tasks. It is a symbolic library of mathematics and is also used as a system for the building and training of neural networks for the detection and decryption of patterns and correlations, commensurate with human learning and reasoning. Used for both research and production on Google.

V. EVALUATION

The evaluation procedure of the proposed methodology include the execution of trading use cases that will show its performance in terms of financial gains. Also, the same use cases executed for the ESVR algorithm that presented in Section II, in order to have a benchmark against state-of-art approaches.

A. Financial Metrics

For the proposed LST-RNN methodology assessment, seven well known financial metrics were used:

- **Annualized Return**. One of the best indicators of how good an investment is considered is the average annual return. It takes into account not only the cumulative profit, but also the trading days in which it took place, and calculates the corresponding annual profit for a trading year. The formula (eq. 4) that calculates this index is as follows, where the sum of the numerator gives the total return and the denominator the time of this return:

$$Annual\ Return = \left(\frac{\sum_{i=1}^{n} R_i}{n}\right) * 252 \quad (\text{eq. } 4)$$

- **Annualized Return with Transaction Costs**. This metric is the same with • Annualized Return, but is calculated taking accounts the costs of the financial transactions, i.e. commission from purchases and sales of stocks.
- **Annualized Volatility**. Volatility is defined as the amount of uncertainty or risk associated with the magnitude of fluctuations at an exchange rate. Annual variability describes the change of an index. A large volatility in an exchange rate implies that the price of the currency may change dramatically in a short time and in any direction. Conversely, in the case of low volatility, the exchange rate will not fluctuate dramatically but will gradually change over a long period of time. From a technical point of view, the term refers to the standard deviation of a change in the value of a financial instrument over a certain period of time (usually this time is a stock market). Just because it represents uncertainty and risk, a great value for this indicator is considered a negative factor. The formula (eq.5) through which annual volatility is calculated is as follows, where the numerator C_i represents the closing value of the i-th day and the denominator C_{i-1} the closing value of the immediately preceding one.

$$Annualized\ Volatility = \sqrt{252} * \sqrt{\frac{\sum_{i=1}^{n}(R_i - R_{avg})^2}{n-1}} \quad (eq.5)$$

$$where\ R_i = \ln\left(\frac{C_i}{C_{i-1}}\right) \quad (\text{eq.}6)$$

- **Sharpe Ratio.** This metric is the most widely used method for measuring risk-adjusted earnings. It was introduced in 1966 by William F. Sharpe, professor at Stanford University, from whom took its name. The idea behind this metric is to provide a clear picture to the investor about the percentage of the expected profit in an environment of increased risk in relation to an environment of limited risk. In this metric, the highest possible values are desirable. The formula used to calculate this metric is depicted in the equation 7. The numerator consists of the expected return on profits minus the risk free return for the period under consideration and the denominator from the average standard deviation of our case. To sum up, we would say that this metric indicates to the investor what proportion of the money he is willing to invest is associated with risk-taking on his part [18].

$$Sharpe\ Ratio = \frac{\overline{R_p} - \overline{r_f}}{\widehat{\sigma_p}} \quad (eq.7)$$

- **Sharpe Ratio with Transaction Costs**. This metric is the same with Sharpe Ratio, but is calculated taking accounts the costs of the financial transactions.
- **Positions Taken**. The changes made by each forecasting model during its implementation period.
- **Correct Directional Change**. This metric denotes as a percentage the correct or not prediction in the movement (rise or fall) of the index from the model used.

B. Evaluation Results

In this section we present the evaluation of LSTM network versus ESVR. Both algorithms were tested on trading use cases against the forecast of ETE stock return. The trading strategy for the examined models is simple and identical for both of them: go or stay long when the forecast return is above zero and go or stay short when the forecast return is below zero. Also a confirmation filter was used which disables changing positions when the forecasted value is below the threshold 0.001.

To take more accurate results each test case repeated 10 time. The mean of the results are presented. The test cases was training for 252 (1 year) and 504 (2 years) days with sliding windows of 1, 2, 5 and 10 days. The following tables presents the results of the trading scenarios that executed.

TABLE II. TRAINING DAYS: 252, SLIDING WINDOW: 1 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	23,61%	**27,41%**
Annualized Return with transaction costs	18,54%	-5.3%
Annualized Volatility	99,96%	99.96%
Sharpe Ratio	0,24	-0,02
Sharpe Ratio with Transaction Costs	0,19	-0,06
Positions Taken	44	41
Correct Directional Change	44,89%	44.65%

TABLE III. TRAINING DAYS: 252, SLIDING WINDOW: 2 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	3,39%	**85,23%**
Annualized Return with transaction costs	-0,51%	75.41%
Annualized Volatility	99,90%	99.84%
Sharpe Ratio	0,03	0,81
Sharpe Ratio with Transaction Costs	-0,01	0,76
Positions Taken	34	38
Correct Directional Change	45,99%	47.21%

TABLE IV. TRAINING DAYS: 252, SLIDING WINDOW: 5 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	26,65%	**75,52%**
Annualized Return with transaction costs	24,26%	78,54%
Annualized Volatility	99,80%	99.86%
Sharpe Ratio	0,28	0.42
Sharpe Ratio with Transaction Costs	0,25	0.87
Positions Taken	21	33
Correct Directional Change	44,91%	41,82%

TABLE V. TRAINING DAYS: 252, SLIDING WINDOW: 10 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	-5,95%	**12,34%**
Annualized Return with transaction costs	-7,52%	8.15%
Annualized Volatility	99,61%	99.52%
Sharpe Ratio	-0,06	0.02
Sharpe Ratio with Transaction Costs	-0,08	0.23
Positions Taken	14	27
Correct Directional Change	45,27%	42.12%

TABLE VI. TRAINING DAYS: 504, SLIDING WINDOW: 1 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	36,58%	**63,21%**
Annualized Return with transaction costs	31,94%	72,23%
Annualized Volatility	99,96%	99,95%
Sharpe Ratio	0,37	0,02
Sharpe Ratio with Transaction Costs	0,32	0,21
Positions Taken	40	41
Correct Directional Change	46,46%	48,82%

TABLE VII. TRAINING DAYS: 504, SLIDING WINDOW: 2 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	-1,28%	**12,32%**
Annualized Return with transaction costs	-2,73%	-3,34%
Annualized Volatility	100,04%	92,95%
Sharpe Ratio	-0,01	-0,01
Sharpe Ratio with Transaction Costs	-0,03	-0.04
Positions Taken	12	25
Correct Directional Change	44,91%	41,82%

TABLE VIII. TRAINING DAYS: 504, SLIDING WINDOW: 5 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	**18,56%**	7,21%
Annualized Return with transaction costs	16,38%	14,34%

Annualized Volatility	99.04%	82,95%
Sharpe Ratio	0,08	0.02
Sharpe Ratio with Transaction Costs	0,06	0.07
Positions Taken	19	20
Correct Directional Change	45,28%	48,23%

TABLE IX. TRAINING DAYS: 504, SLIDING WINDOW: 10 DAYS

Financial Metrics	GA-SVR	LSTM Network
Annualized Return	**-1,28%**	-3,12%
Annualized Return with transaction costs	-2,73%	-2,34%
Annualized Volatility	100,06%	25,95%
Sharpe Ratio	-0,01	0.03
Sharpe Ratio with Transaction Costs	-0,03	0.12
Positions Taken	13	19
Correct Directional Change	44,91%	49,32%

C. Discussion

From the above results, we conclude that GA-SVR and LSTM Network have similar behaviour but there are significant differences in specific metrics for some scenarios. Regarding the Annualized Return, an outperformance of LSTM Network is noted in two cases, where the difference is more than 50%, and we can assume that LSTM is overfitting in this dataset. Nevertheless, in the rest of the use cases, it performs better in four out of six scenarios. Regarding Annualized Volatility, both algorithms behave the same at 252 training days scenarios while GA-SVR shows a better behaviour in the 504 training days scenarios, meaning that LSTM waited more dramatically changes of the stock's value and thus it made more transactions, which is verified by the Positions Taken values. The high Annualized Volatility, presented in both models can be explained by the dataset synthesis. All data is collected at the years which economic crisis hits Greece. These 5 years Greece government synthesis changed 4 times and a lot of economic measures were taken, According Correct Directional Change both algorithms have the same mediocre results. About this metric, it is difficult to predict when the price will change, f today the price has risen and tomorrow and the next few days fall, the algorithm may not catch the next day price fall, but in the next one it has been adjusted and predicted the fall. For the other metrics, both methodologies where more or less the same.

As a general conclusion, the LSTM is able to behave better and generate more financial gains in most of the scenarios. Nevertheless, this inference must be used very carefully taking into account the possibility of overfitting when the profits are very high. Furthermore, using high number of training days, shows that LSTM network approach behaves as it operates in a more risky environment than the reality indicates.

VI. CONCLUSION

The presented work focuses on the problem of modelling and trading financial indices. Due to the complexity of the problem, the large volume of financial data and the existence of dependencies among various indices that have not been interpreted yet, the proposed solution introduced LSTM networks as a suitable solution. The approach is assessed running trading scenarios in ETE stock of FTSE/ASE Large Cap group of Athens Stock Exchange (Greece) utilising collected data from approximately 5 years. Also the LSTM Network approach benchmarked against a hybrid approach combining Genetic Algorithms and Support Vector Machines (GA-SVR).

From the experiments' results we can see that regarding the annual gain, LSTM network approach is advantageous in the most of the scenarios against the GA-SVR approach. Also, it seems that the proposed approached works better in use cases with a small number of training days, but it needs very cautious use in order to avoid overfitting.

In order to improve the constraints of the proposed model, as a future direction, we scope to follow a similar to GA-SVR hybrid approach using evolutionary calculation. We believe that by setting up a suitable fitness function we could optimize our model parameters and minimize the overfitting problem. Indicatively, Genetic Algorithms or Diversity or Optimization clusters could be used towards to achieve this scope.

REFERENCES

[1] Dunis, C., Likothanassis, S., Karathanasopoulos, A., Sermpinis, G., and Theofilatos, K.: Computational Intelligent Techniques for Trading and Investment, Taylor and Francis (Series: Advances in Experimental and Computable Economics), ISBN: 978-0-415-63680-3, (2013).

[2] Chen, C.C., Tsay, W.J.: A Markov regime-switching ARMA approach for hedging stock indices. Journal of Futures Markets. 31(2), 165–191 (2011).

[3] Sermpinis, G., Laws, J., Dunis, C.L.: Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks. European Journal of Finance, 19(3), 165-179 (2011).

[4] Hsieha, T.J., Hsiaob H.F., Yeh W.C.: Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. Applied Soft Computing. 11(2), 2510–2525 (2011).

[5] Dunis, C., Likothanassis, S., Karathanasopoulos, A., Sermpinis, G. and Theofilatos. K.: A hybrid genetic algorithm–support vector machine approach in the task of forecasting and trading. Journal of Asset Management, 14, 52–71 (2012).

[6] Doman, M., and Doman, R.: Dependencies between Stock Markets During the Period Including Late-2000s Financial Crisis. Procedia Economics and Finance 1, 108-117 (2012).

[7] Graham, M., Kiviaho, J., and Nikkinen, J.: Short term and long-term dependencies of the S&P500 index and commodity prices, 13(4), 583-592 (2013).

[8] Amorgianiotis, T., Mitra, S., Theofilatos,K., Likothanassis, S. (2014) Uncovering Dependencies between European Markets using ESVM Fuzzy Inference Trader, In: Proceedings of Forecasting Financial Markets Conference, Marseilles, May 2014.

[9] Boyd D., Crawford K.: CRITICAL QUESTIONS FOR BIG DATA, Information, Communication & Society, 15(5), 662-679(2012).

[10] Theofilatos, K., Georgopoulos, E., Likothanassis, S., Mavroudi S. (2014) Computational Intelligence: Recent advances, perspectives and open problems, In: Computational Intelligence for Trading and Investment, Routledge. DOI: 978-0- 415-63680-3

[11] Theofilatos, Konstantinos, Andreas Karathanasopoulos, Georgios Sermpinis, Thomas Amorgianiotis, Efstratios Georgopoulos, and Spiros Likothanassis. "Modelling and Trading the DJIA Financial Index using neural networks optimized with adaptive evolutionary algorithms." In

International Conference on Engineering Applications of Neural Networks, pp. 453-462

[12] Cao, L. and Tay, F. (2003) Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks. 14(6), 1506 – 1518.

[13] Choudhury. S., Ghosh. S., Bhattacharya. A., Fremandes. K.R. and Tiwari. K. M., (2014). A real Time Clustering and SVM Based Price-Volatility Prediction for Optimal Trading Strategy. Neurocomputing 131. 419-426.

[14] Karathanasopoulos, A., Theofilatos, K. A., Sermpinis, G., Dunis, C., Mitra, S., and Stasinakis, C. (2016) Stock market prediction using evolutionary support vector machines: an application to the ASE20 index. European Journal of Finance, 22(12), pp. 1145-1163

[15] Medsker, L.R. and Jain, L.C., 2001. Recurrent neural networks. Design and Applications, 5.

[16] Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". Neural Computation. 12 (10): 2451–2471.

[17] Sharpe, W.F., 1994. The sharpe ratio. Journal of portfolio management, 21(1), pp.49-58.

[18] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016, November. TensorFlow: A System for Large-Scale Machine Learning. In OSDI (Vol. 16, pp. 265-283).