CONTROL OF CORPORATE OWNERSHIP IN THE EVOLUTIONAL PORTFOLIO INTELLIGENT COMPLEX OPTIMIZATION (EPICO) MODEL

Nikolaos Loukeris*, Iordanis Eleftheriadis*

*University of Macedonia Department of Business Administration Egnatias 156, 540 06 Thessaloniki, Greece

OPEN ACCESS

How to cite this paper: Loukeris, N., & Eleftheriadis, I. (2017). Control of corporate ownership in the evolutional portfolio intelligent complex optimization (EPICO) model. *Corporate Ownership & Control, 14(4-1),* 301-313. doi:10.22495/cocv14i4c1art12

How to access this paper online: http://dx.doi.org/10.22495/cocv14i4c1art12

ISSN Online: 1810-3057
ISSN Print: 1727-9232

Received: 27.03.2017
Accepted: 02.06.2017

JEL Classification: G0, G020, G110, G12, G17
DOI: 10.22495/cocv14i4c1art12

Abstract

We introduce a new methodology that incorporates advanced higher moments evaluation in a new approach of the Portfolio Selection problem, supported by effective Computational Intelligence models. The Evolutional Portfolio Intelligent Complex Optimization (EPICO) model extracts hidden patterns out of the numerous accounting data and financial statements filtering misguiding effects such as noise or fraud, offering an optimal portfolio selection method.

Keywords: Isoelastic Utility finction, Higher moments, Recurrent Neural Networks, Time-Lag Recurrent Neural Networks, Jordan-Elman Neural Networks, MLP, Hybrid Networks, Genetic Algorithms, Portfolio Optimization

1. INTRODUCTION

The portfolio selection problem is a two phase process. Initially, the feasible set is created after the evaluation of portfolios, usually rejected by the risk-averse investors. Then the efficient portfolios guided by a utility function, are ranked and the model receives the optimal portfolios that reflect the behavior of investors. Secondly, the risk expressed in further higher moments (variance, volatility, hyperkurtosis, ultrakurtosis, hyperultrakurtosis) is minimized.

The objective of this research is to examine the first step of the problem, providing a general solution on the second step. We create an integrated system that optimizes portfolios in advanced methods of Computational Intelligence and Finance. The single period model is examined, as we evaluate various artificial intelligence models of the FeedForward Networks family: the Voted Perceptrons, the Multilayer Perceptrons, the Generalized Feed Forwards to the Bayesian Logistic Regressions as a benchmark.

The models, either neural or hybrid neuro-genetics of different topologies each, were: 1 on the Voted Perceptron, 46 on the MLPs, 66 on the GFFs, and 1 on the Bayesian Logistic Regression to create the efficient portfolio set. The scope is quintuple: i) the thorough investigation of the investors behavior in higher moments, retrieving in depth information on earnings and preferences of risk, ii) to develop the isoelastic utility as a function that supports higher moments, iii) to improve Markowitz's portfolio theory, after incorporating fundamentals evaluation, in the domain of Fractal

Markets Hypothesis and Chaos Theory, to clear noise, to exclude fraud, and to introduce the shelf-organising chaotic patterns in the random walk of returns, iv) to examine the efficiency of various networks of the Feed Forward family in neural or hybrid neuro-genetic nets concluding the optimal classifiers for an integrated investment model, v) to introduce the integrated model EPICO as a cutting-edge solution to portfolio optimization problem. The research framework, inspired by the [1] initial work, and its current advancements [2], [3], upgrades the Portfolio Theory, in new tools of Heuristics further than [4], [5], etc to a cutting edge integrated solution to the problem.

This paper is organized as follows: in Section 1 is the introduction, in 2 the literature on

higher order moments, in 3 the wavelet theory, in 4 the advances on Utility Functions, in 5 the problem's definitions, in 6 the integrated EPICO model, in 7 the Computational Intelligence methodology, in 8 the data, in 9 the results, and in Section 10 the Concluding Remarks.

2. HIGHER MOMENTS

Since the returns distributions are not n.i.i.d., and EMH fails in the markets as various non-financial effects alter the stock prices, whilst it also neglects the existence of extreme events. The Fractal Markets Hypothesis-FMH is a more robust approach as it introduces the dominance of short time horizons during turbulent periods. The FMH accepts that (1) a market consists of many investors with different investment horizons, and (2) the information set that is important to each investment horizon is different. The market remains stable during the period that it maintains the fractal structure, without a characteristic time scale. The market becomes unstable in case its investment horizon becomes uniform, because all the investors are trading based upon the same information set. The Global Financial Crisis (2007 onwards) has noticed that the growth of financial asset prices is a rather cyclical than a linear one. During the most turbulent times, major indices lost around 50% of their 2007 peak values during the 1.5 year long decline. The EMH, can't describe the crisis such because investors are assumed to price the assets rationally with respect to all available information. The investors are falsely assumed to be homogeneous on the effect of information, although it is well known that they are heterogenous with a wide range of investment horizons [6]. The investors with short investment horizons focus on technical information and crowd behavior of other market participants, whilst investors with long investment horizons decide on fundamental information and neglect the crowd behavior. The high complexity of the financial markets requires more advanced methods than the introductory linear methods. Thus the fractal markets hypothesis (FMH), emphasize on empirical observations of the markets that are consisted of heterogenous agents who react to the inflowing information with respect to their investment horizon.

The central idea on wavelets into the financial processes is their ability to analyze the underlying process both in the time and frequency domain. Similarly to the standard Fourier analysis, the series is decomposed into frequencies and a scale-specific power is obtained. The wavelets improve the Fourier analysis informing on the scale-specific power evolution in time. The financial wavelet power, [6], is scale-specific variance that informs about time evolution of variance as well as its distribution across frequencies (scales). Increased power is noticed at high frequencies (low scales) during the critical periods, and a changing structure of variance across frequencies, might be observed, before the turbulences due to the changing structure of investors' activity. [6], showed that the most turbulent times of the current Global Financial Crisis can be very well characterized by the dominance of short investment horizons in parallel to the FMH. Misbalance between short and long investment horizons created a tension between supply and demand, leading to decreased liquidity which has been repeatedly shown to lead to occurrence of extreme events.

In terms of behavior, [7], observed that investors are more sensitive to their potential losses, we will try to model the overall preferences, even those that incorporate the sub-conscious trends that guide them. The investors distribute their utility balancing perceptions and fears, on the one hand, and earnings on the other. Their logic expects a rational amount of return, but the fear of loss subconsciously magnified, produces remarkable decisions. The majority of investors are risk averse or risk neutral, hence the fear parameter is easy to manipulate behaviors. In bullish periods the fear of losing excess profits, whilst in bearish the fear of maximising losses, can influence non rational herding behaviors.

A more analytical tool that will focus in depth the details that define investing behaviors is introduced. The further higher moments detect the hidden aspects of investors' decision making. [8], [9] noticed that on the implied utility function of the HARA family (Hyperbolic Absolute Risk Aversion) the 5th of hyperskewness and the 6th of hyperkyrtosis moments should be used in the form of formula1 or formula 2.

$$U_t(R_{t+1}) = aE_t(R_{t+1}) - bVar_t(R_{t+1}) + cSkew_t(R_{t+1}) - dKurt_t(R_{t+1}) + eHypSkew_t(R_{t+1}) - fHypKurt_t(R_{t+1}) + gUltraSkew_t(R_{t+1}) - hUltraKurt_t(R_{t+1}) \quad (1)$$

or

$$U_T(R_{t+1}) = \alpha\mu + E(x_i-\mu)^2(-b + cE(x_i-\mu) - dE(x_i-\mu)^2 + eE(x_i-\mu)^3 - fE(x_i-\mu)^4 + eE(x_i-\mu)^5 - fE(x_i-\mu)^6) + W_x(u,s) \quad (2)$$

where,

$$Kurt_t(R_{t+1}) = Var_t^2(R_{t+1}) \quad (3)$$

$$HypKurt_t(R_{t+1}) = Kurt_t^2(R_{t+1}) = Var_t^4(R_{t+1}) \quad (4)$$

$$UltraKurt_t\left(R_{t+1}\right) = Kurt_t^4\left(R_{t+1}\right) = Var_t^8\left(R_{t+1}\right) \quad (5)$$

Thus (2) as a series of higher order moments can be extended to the level of analysis that is desired.

A general form of the utility function is:

$$U_t\left(R_{t+1}\right) = \sum_{\lambda_v=1}^{\omega} (-1)^{\lambda_v+1} \frac{\alpha\lambda_v}{n} \sum_{t=1}^{n} \left(x_i - \sum \frac{x_i}{n}\right)^n + W_x\ (u,s) \quad (6)$$

where λ_v is the depth of accuracy on investors utility preferences towards risk, depending on the behavior, $a_{\lambda v}$ a constant on investors profile: $a_{\lambda v} = 1$ for rational risk averse individuals, $a_{\lambda v} \neq 1$ for the non-rational, xi the value of return *i* in time t, $W_x(u,s)$ the continuous wavelet transform, s the scale of the wavelet, u its location.

3. WAVELET

The wavelet $\psi_{u,s}(t)$ is a real-valued square integrable function defined as:

$$\psi_{u,s}(t) = \frac{\psi\left(\frac{t-u}{s}\right)}{\sqrt{s}} \quad (7)$$

with scale *s* and location *u* at time *t*. Given the admissibility condition holds:

$$C_\psi = \int_0^{+\infty} \frac{|\psi(f)|^2}{f} df < +\infty \quad (8)$$

The time series can be reconstructed from its wavelet transform, where *Ψ(f)* the Fourier transform of a wavelet. The wavelet has a zero mean: $\int_{-\infty}^{+\infty} \psi^2(t)dt = 0$ normalized: $\int_{-\infty}^{+\infty} \psi(t)dt = 1$. The continuous wavelet transform $W_x(u,s)$, projects a wavelet *ψ(.)* onto the examined series *x(t)*:

$$W_x(u,s) = \int_{-\infty}^{+\infty} \frac{x(t)\psi^*\left(\frac{t-u}{s}\right)dt}{\sqrt{s}} \quad (9)$$

where, *ψ*(.)* a complex conjugate of *ψ(.)*. The continuous wavelet transform decomposes the series in frequencies and recreates the original series with no information loss, as the energy of the examined series is maintained:

$$x(t) = \frac{\int_0^{+\infty}\int_{-\infty}^{+\infty} W_x(u,s)\psi_{u,s}(t)\ duds}{s^2 C_\psi} \quad (10)$$

$$\|x\|^2 = \frac{\int_0^{+\infty}\int_{-\infty}^{+\infty} |W_x(u,s)|^2 duds}{s^2 C_\psi} \quad (11)$$

where, $|Wx(u,s)|^2$ is the wavelet power at scale s > 0.

The Morlet wavelet is selected, as a preferred form of financial applications [10] Aguiar-Conraria, L., [11]), [12], [13], [14]. The Morlet wavelet's central frequency preferably at ω_0=6 is:

$$\psi(t) = \frac{e^{iw_0}\ t - t^{2/2}}{\pi^{1/4}} \quad (12)$$

providing a nice balance between the time and frequency localization.

4. THE UTILITY

The Isoelastic Utility, a HARA function of CRRA, is on the risk averse investors:

$$U = \begin{cases} \frac{W^{1-\lambda}-1}{1-\lambda}, & \lambda \in (0,1)\cup(1,+\infty] \\ \log(x), & \lambda = 1 \end{cases} \quad (13)$$

where, W the wealth, λ a measure of risk aversion. [8], [9] indicated the Makowitz model can have a broader alternative relaxing its essential assumption on the normaly distributed prices.

The initial convex problem of quadratic utility maximization, [1],

$$\min_x f(x) = Var(r_p) \quad (14)$$

is inadequate in real markets [2] incorporated higher order moments:

$$\min_x f(x) = \lambda Var(r_p) - (1-\lambda)E(r_p) \quad (15)$$

$$r_p = \sum_i x_i r_i \quad (16)$$

$$x_i \geq 0 \quad (17)$$

$$\sum_i x_i = 1 \quad (18)$$

where, r_p the portfolio return, x_i the weight of asset *i*, r_i the return of i^{th} asset, μ the mean and σ^2 the variance.

5. PROBLEM DEFINITION

[8], [9] indicated the necessity of further higher moments into the model, to optimally describe investors' preferences. The problem is:

$$\min_x f(x) = \lambda \upsilon_\gamma \left[bVar_t(r_p) + d\ Kurt_t(r_p) + f\ HypKurt_t(r_p) - hUltraKurt_t(r_p) \right])] - (1-\lambda)\upsilon_\gamma \left[aE_t(r_p) + cSkew_t(r_p) + eHypSkew_t(r_p) + g\ UltraSkew_t(r_p) \right] \quad (19)$$

$$\upsilon_\gamma = 1 - \varepsilon_\tau \quad (20)$$

$$r_p = \sum_i x_i r_i^* \quad (21)$$

where, υ_γ the financial health of the company (binary: 0 towards bankruptcy, 1 healthy); ε_τ the heuristic model output that is the evaluation result (binary: 0 healthy, or 1distressed); r_i^* the return of stock i that belongs to the efficient frontier and is superior than the others; x_i their weights.

The superiority relation of the selected stocks within the portfolio is i^* *sup* j if and only if $R_t(i^*) > R_t(j)$, analysed into

$$Var_t R_t(i^*) < Var_t R_t(j) \quad (22)$$

$$Kur_t R_t(i^*) < Kur_t R_t(j) \quad (23)$$

$$HypKur_t R_t(i^*) < HypKur_t R_t(j) \quad (24)$$

$$UltraKurt_t R_t(i^*) < UltraKurt_t R_t(j) \quad (25)$$

The stocks that fail to fulfill all the previous superiority conditions are non-optimal stocks and are exempted from the optimal portfolio set, as a part of the efficient frontier. Thus given [8], [9]:

$$U_t(R_t(i)) = \sum_{\lambda_v=1}^{\omega} (-1)^{\lambda_v + 1} \frac{\alpha \lambda_v}{n} \sum_{t=1}^{n} \left(x_i - \sum \frac{x_i}{n} \right)^n + W_x(u,s) \quad (26)$$

then,

$$U_t(R_t(i^*)) > U_t(R_t(x_j)) \quad (27)$$

hence,

$$U_t(r_p) = \sum U_t(R_t(i^*)) \quad (28)$$

The previous is identical to:

$$max_x E\left(U_p(w,\lambda)\right) \quad (29)$$

in

$$E(U_p(w,\lambda)) = \max \left\{ \frac{\sum_i \left[1 + \exp(r_i x_i)\right]^{1-\frac{\upsilon_\gamma}{\lambda}}}{1-\frac{\upsilon_\gamma}{\lambda}} \right\} / N \quad (30)$$

$$E(U_p(w,\lambda)) = \max\{\Sigma_i [1 + exp(r_i x_i)]^{1-\upsilon\gamma/\lambda} / (1-\upsilon_\gamma/\lambda)\}/N \quad (31)$$

let,

$$Var_t^2(r_p) = z \quad (32)$$

and

$$Var_t\ (R_{t+1}) = y \quad (33)$$

as

$$z = y^2 = \sigma^4 \quad (34)$$

then,

$$min_x f(x) = \lambda \upsilon_\gamma Var_t(r_p)[b + dz + fz^2] \quad (35)$$

The non-convex problem, requires strong heuristics to be solved. The new contribution is that we extract hidden accounting and financial patterns to a thorougher stock's evaluation. Fraud and manipulation are a significant risk to investors. Thus, under (20) and (35) I filter the distressed companies with no significant potentials from portfolios. The evaluation $\upsilon\gamma$, in (20) is more important than the investor's risk behavior, as they have a reverse influence in $\upsilon\gamma/\lambda$. The minx f(x) equality in (35) declares a categorical, objective influence of an asset is more influential than subjective investors' behavior. The flow chart of processes is described in figure 1.

6. THE EPICO MODEL

The Evolutional Portfolio Intelligent Complex Optimization (EPICO) is an integrated model that further develops the EPOS – Evolutional Portfolio Optimization System, **[3]**.

The EPICO model on the first step reads the fundamentals, the accounting data, the market prices and the preferred optimization period t.

Then it proceeds by selecting the initial method to evaluate the companies whose stocks are candidate in the portfolio. On this step the individual investor's risk profile is given and the λ is selected for the Isoelastic utility.

On the next step, the system examines if this is the last firm to be examined, and if the condition for the optimal portfolio as an efficient portfolio is satisfied. Else we proceed to the next of the initial evaluation that uses a Computational Intelligence model, to create two subsets: Subset A of the healthy companies, and Subset B of the distressed firms.

In the specific model we select the best network among the models, the ε_τ value is calculated (0, for the healthy and 1 for the distressed firms).

If the health index ε_τ is the same to the index of the previous year then the model moves on the next step.

Else the firm is fluctuating and must be removed from the portfolio.

If $\varepsilon_\tau = 1$ then the firm is distressed and it is removed, else if $\varepsilon_\tau = 0$ the firm is healthy being candidate for the optimal efficient portfolio.

On the next step the $U_t(R_t())$ *the* utility function of (105) is calculated per firm.

Next, firms are ranked according to their utility score.

Then, the Efficient Frontier is calculated.

Next, the firms with the higher utility score are selected into the efficient portfolio.

The sub-optimal firms as well as the non-optimal firms are revaluated with potential new data on the step 4 of Neural Nets evaluation, following all the steps.

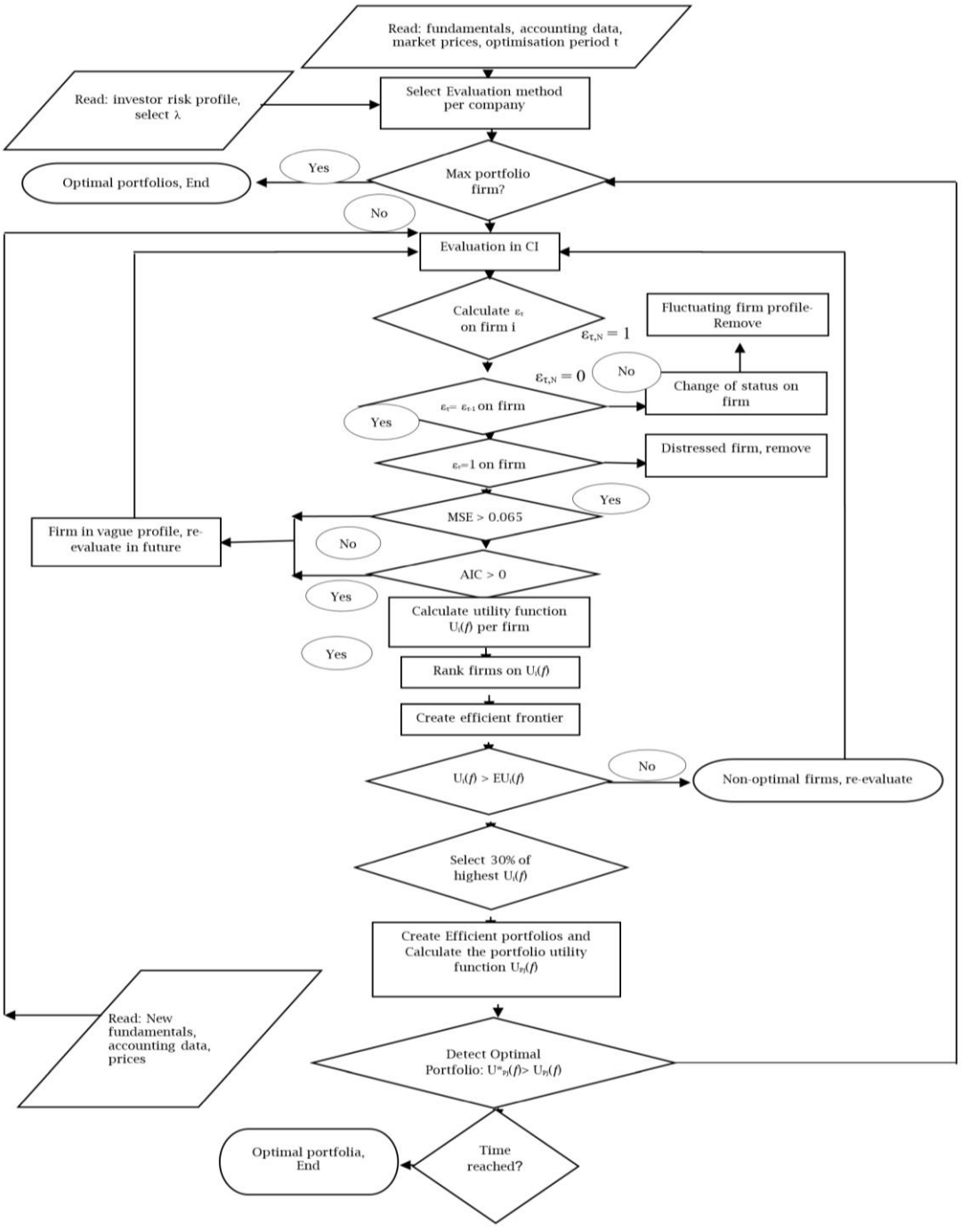
Next, after the efficient portfolio is created, its Utility Function is calculated $U_{Pj}(f)$.

Then, the optimal overall portfolio $U^*_{Pj}(f)$ whose utility is the maximum available, is detected, if possible, by all the available efficient portfolios utilities $U_{Pj}(f)$ recorded in $U^*_{Pj}(f) > U_{Pj}(f)$.

The process stops when the time limit is reached and the PI has the optimal portfolio.

The key idea is to filter fraud and speculative noise that interfere on the price and disorient investors. Thus examining recent accounting entries and through their financial indexes we can define the real financial health of the firm. After the real healthy firms are selected then their returns are considered on the model and we proceed on the main core of the Markowitz initial approach, the detection of efficient frontier and the creation of the efficient portfolio. The model's flow chart is in Figure 1.

Figure 1. The EPICO model



7. THE COMPUTATIONAL INTELLIGENCE

The emphasis on the classifier gill be given on the Feed forward networks family. A thorough investigation of i) the Voted Perceptron, ii) the Multi-Layer Perceptron, and iii) the Generalised Feed Forward and the MLPs as a measure of comparison will take place. All the models will be examined in neural net and hybrid neuro-genetic form, on various topologies. In the following sub-sections a detailed presentations of these models takes place.

7.1. The Voted Perceptron

The Voted Perceptron-VP is a refinement of the Perceptron algorithm preferably in cases of noisy or inseparable data, [15], [16], [17], [18], [3]. The training phase similar, whilst a change relies on the test examples. The algorithm can be considered to build a series hypotheses $G^t(x)$ for $t = 1....n$ where G^t is the scoring function from the algorithm trained on just the first t training examples. The output of a model trained on the first examples for a sentence s is:

$$G^t(x) = \sum_{(i \le t,\, j)} \alpha_{i,j} (h(x_{i1})\ h(x) - h(x_{ij})\ h(x)) \quad (36)$$

Thus the training algorithm can be considered to construct a sequence of n models, V_1, ...V_n. On a test sentence s, each of these n functions will return its own parse tree, Vt(s) for $t = 1...n$. The Voted Perceptron picks the most likely tree as that which occurs most often in the set $\{V_1(s) \ldots V_n(s)\}$. The G^t is easily derived from G^{t-1}, though the identity;

$$G^t(x) = G^{t-1}(x) + \sum h_{.}\, h_{.i} \quad (37)$$

hence, the Voted Perceptron can be implemented with the same number of kernel calculations, under roughly the same computational complexity, as the original Perceptron, [19].

7.2. The Multi-Layer Perceptron

The Multi-Layer Perceptron–MLP, [4], [19], [18], [8], [3], is a widely used neural network, whose inputs are processed in a numerous layers [20], [18], [3] that contain artificial neurons. The number of neurons in the input is identical to the variables, as the outputs are equal to the number of classes, and nonlinear intermediate neurons form the hidden layers, [21]. The neurons except those on the input, produce a linear combination of the outputs of previous layers plus a bias. The synaptic weights, between different neurons, normalized with the output classes to 0-1 so that the MLP concludes to the optimal performance of the maximum ex-post receiver in classifications, [22]. Next neurons in the hidden layer process a non-linear sigmoid function of their input.

$$\varphi(x) = 1/(1+exp(-x)) \quad (38)$$

The output neurons produce a result equal to the linear combination. The output is discrete in {0,1}, whilst the output c of each neuron is:

$$c = \varphi\ (\sum_i w_i\, \alpha_i + b) \quad (39)$$

where, a_i - the inputs of the neuron and w_i - the weights of the neuron. MLPs (figure 1), can approximate arbitrary functions, when they are trained with the backpropagation algorithm, [23]. The errors in the network are minimized through the backpropagation rule, allowing adaptation of the hidden neurons. Multi-Layer Perceptron with nonlinear neurons have a smooth nonlinearity as the logistic function and the hyperbolic tangent, whilst their massive interconnectivity permits the computation non linear functions. The Multi-Layer Perceptron is trained with error correction learning, where the desired response for the system must be known [24], [25]. A bias affects biological neurons in extreme weather conditions or in physiological disorders, thus a bias input is given to each one of the artificial neurons, The MLPs in a hybrid form optimised by Genetic Algorithms were elaborated into NeuroSolutions software. Genetic Algorithms select the most significant inputs among the 16 financial inputs in the hybrid MLP. The network, through multiple training, finds the inputs combination with the lowest error. Genetic Algorithms were used on each layer in the MLPs with different topologies. Batch learning were chosen to update the weights of hybrid neuro-genetic MLP, in an accumulative manner. Genetic Algorithms optimised the sub-problems regarding the: a) number of Processing Elements, b) Step Size, and c) Momentum Rate. The output layer optimized the value of Step size and Momentum through Genetic Algorithms.

Figure 2. Multi-Layer Perceptron biased with n hidden nonlinear layers

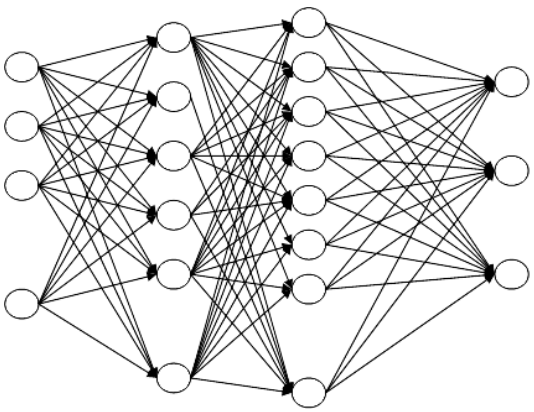
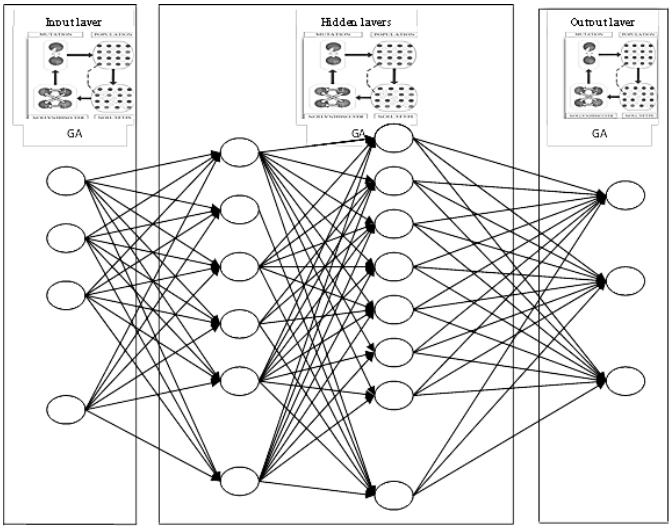


Figure 3. The Hybrid Multi-Layer Perceptrons, of Genetic Algorithms optimization in all layers and Cross Validation



7.3 THE HYBRID GENERALISED FEED FORWARD NETWORKS

The Generalised Feed Forward networks, (Fig.2) are a generic form of the MLPs being able to let their synapses jump over one or more layers. In the GFF we used an initial MLP is created where each layer feeds forward its output signal to all subsequent layers. The real performance of the GFFs showed that they resolve the problems much more efficiently then the MLPs, in a fraction of the time, on the same number of neurons.

Figure 4. The Generalised Feed Forward Network

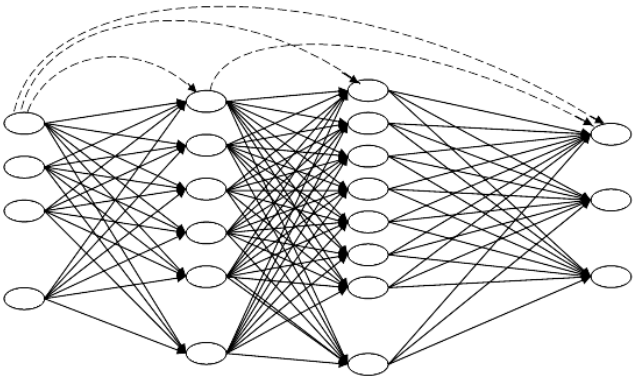
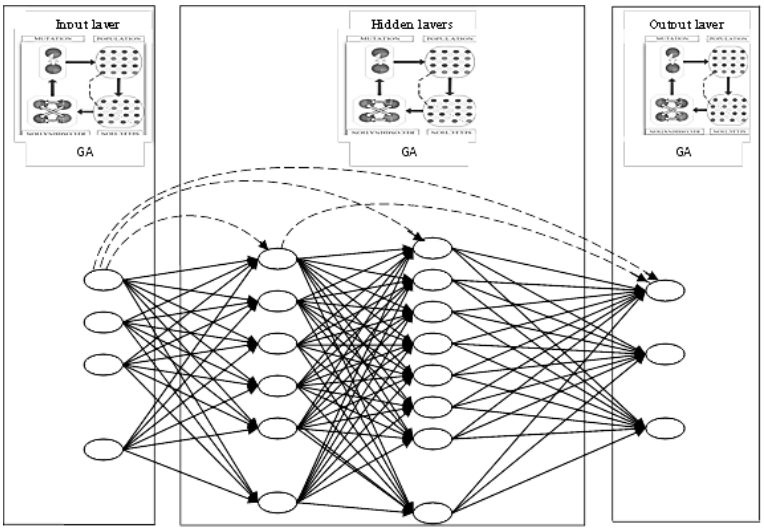


Figure 5. The Hybrid Generalised Feed Forward networks, of GA optimization in all layers and CV



7.4 THE BAYESIAN LOGISTIC REGRESSION

The Bayesian Logistic Regression for both Gaussian and Laplace Priors, [26], [18], [3], was implemented by the Weka platform. The Logistic regression is discriminative probabilistic linear classification in

$$p(C_1| \text{x}) = \sigma(\text{w}^T\varphi) \quad (40)$$

though the exact Bayesian inference for the Logistic Regression is inflexible, an evaluation of posterior distribution p(w|t) takes place, whilst a step of normalisation to prior is required as:

$$p(w)=N(\text{w}|\text{m}_0,\text{S}_0) \quad (41)$$

under the likelihood multiplication generated by the sigmoids, results to:

$$p(t|w) = \prod_{n=1}^{N} y_n^{tn}\{1 - y_n\}^{1-tn} \quad (42)$$

The Bayesian Logistic Regression can apply Laplace approximation to get the Gaussian q(w). The evaluation of the predictive distribution comes as a convolution:

$$p(C_1 |\varphi, \text{t}) = \int\sigma(\text{w}^T\varphi,t)q(\text{w})d\text{w} \quad (43)$$

of the sigmoid and Gaussian, and the approximate Sigmoid by Probit function. The Laplace Approximation requires the mode w_0 of the posterior distribution p(w|t) and it is being done by a numerical optimization algorithm. It also fits a Gaussian centered at the mode:

$$q(\text{w}) = \text{N}(\text{w}|\text{w}_0,\text{A}^{-1}) \quad (44)$$

whilst there is a need for second derivatives of log posterior that is equivalent to find a:

$$\text{A} = -\nabla\nabla\ln f(\text{w}) |_{\text{w}=\text{w}0} \quad (45)$$

Hessian matrix:

$$\text{S}_\text{N} = -\nabla\nabla\ln p(\text{w}|\text{t}) \quad (46)$$

8. DATA OF NEURAL COMPUTATION

Data came by 1411 companies from the loan department of a Greek commercial bank, with the following 16 financial indices:

1) EBIT/Total Assets;
2) Net Income/Net Worth;
3) Sales/Total Assets;
4) Gross Profit/Total Assets;
5) Net Income/Working Capital;
6) Net Worth/Total Liabilities;
7) Total Liabilities/Total assets;
8) Long Term Liabilities /(Long Term Liabilities + Net Worth);
9) Quick Assets/Current Liabilities;
10) (Quick Assets-Inventories)/Current Liabilities;
11) Floating Assets/Current Liabilities;
12) Current Liabilities/Net Worth;
13) Cash Flow/Total Assets;
14) Total Liabilities/Working Capital;
15) Working Capital/Total Assets;
16) Inventories/Quick Assets.

And the17th index with initial classification, done by bank executives, based on [27]. Test set was 50% of overall data, and training set 50%. The 1411 companies are unique on the dataset, each index value is the 3 years average, [27], and the dependent value ετ is binary, in 0 for the healthy, and 1 for the distressed companies. The frequency of observation is discrete, having 3 different values in most of the firms, offering the average elaborated by the models. The resources used was an AMD Athlon II of 4 cores at 2.61 GHz in Win XP SP2, X32 bits, 4GB RAM for all the models.

9. RESULTS

The classification evaluation process of the EPICO model is advanced by the appropriate classifier:

9.1. Results of Multi-Layer Perceptrons

The MLP in WEKA software [18], [17], [8], [3], used the backpropagation to train, providing a graphical representation of the network, having the ability to be altered only while the network is not running. The number of epochs to train was 500. The training set had 706 companies (50%) and the test set 705 (50%). The test set results provide information on the most efficient MLP topology, because the synaptic weights were adjusted by the previous learning process of the training set. Thus Multi-Layer Perceptron with 6 hidden layers had the optimal classification convergence, since it offered the highest number of correct classifications, 650 (92.14%), that is to say 593 healthy companies were categorized as healthy out of 706, and 57 as in distress. The incorrect classifications of MLP with 6 layers were the lowest at 55 cases (7.8%). The interobserver agreement expressed by the Kappa statistic was the highest at 0.6344 for the MLP of 6 layers, and the cost function in the form of MSE was at the second lowest place (the lowest was for the MLP with A: 9 layers) indicating an excellent fitness of the network output to the desired output. MLP with number of hidden layers A=(attribs + classes) / 2 = (16+2)/2 = 9, had a very good performance as well, slightly inferior than the optimal MLP of 6 layers, followed by the MLP of 2 layers.

9.2. Results of hybrid neuro-genetic Multi-Layer Perceptrons.

The results from hybrid neuro-genetic MLP lacked convergence accuracy as they only classified correctly the healthy at a rate 100%, [28], [29]. There were 16 MLP neural networks examined in two different programmes where 6 and 10 topologies were examined respectively. The MLP hybrids were 30, and of 3 different architectures in 10 alternative topologies. The Hybrid MLP optimised by GA on the input layer only, deciding the importance of the financial indices to the model and their use, provided the best performance the in hybrid MLPs and in every model examined in this research overall, on the net of 1 layer, where the healthy firms were classified correctly at 99.66%, and the distressed at 94.17%, in the lowest MSE at 0.034, and the best fitness at 0.958, the lowest error at 3.16%, in a very low Akaike Criterion revealing an impartial model, but exposed to overtraining, in a fast time of 51 minutes and 2 seconds.

Table 1. Results of MLP, [18], [17], [8]

MLP Test Set	*0→0*	*0→1*	*1→0*	*1→1*	*Misclas.*	*Cor. cl.*	*K-stat*	*MAE*	*MSE*	*RMSE*	*RAE*	*RRSE*	*Time*
MLP A layers	592	4	52	57	56 7.94%	649 92.05%	0.6295	0.0808	0.0614	0.2479	30.86%	68.57%	10.45 sec
MLP 1	594	2	64	45	66 9.36%	639 90.63%	0.5335	0.1089	0.0840	0.2899	41.56%	80.18%	2.27 sec
MLP 2	593	3	59	56	56 7.94%	649 92.05%	0.6261	0.0897	0.0659	0.2569	34.24%	71.05%	3.28 sec
MLP 3	594	2	63	46	65 9.21%	640 90.78%	0.5428	0.0986	0.0817	0.2859	37.64%	79.07%	4.39 sec
MLP 4	593	3	57	52	60 8.51%	645 91.48%	0.5918	0.0905	0.0731	0.2704	34.53%	74.78%	5.06 sec
MLP 5	593	3	58	51	61 8.65%	644 91.34%	0.5831	0.0888	0.0718	0.268	33.89%	74.13%	6.02 sec
MLP 6	*593*	*3*	*52*	*57*	*55 7.80%*	*650 92.19%*	*0.6344*	*0.0819*	*0.0627*	*0.2505*	*31.27%*	*69.28%*	*7.08 sec*

Table 2. Results of Hybrid MLPs, [17], [8] (Part 1)

Neural Network		*Active Confusion Matrix*				*Performance*						
MLP Neural Network	*Layers*	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
	1	*100*	*0*	*98.16*	*1.83*	*0.417*	*0.986*	*0.121*	*19.399534*	*-471.148*	*-377.72*	*22"*
	2	100	0	98.16	1.83	0.417	0.986	0.123	19.348611	-431.387	-312.365	23"
	3	100	0	100	0	0.422	0.999	0.153	19.603922	-381.471	-236.853	26"
	4	100	0	100	0	0.425	1.007	0.136	21.926317	-336.485	-166.270	29"
	5	100	0	100	0	0.643	1.521	0.084	39.624512	-5.3443	190.466	26"
	6	100	0	100	0	0.755	1.785	0.004	44.192734	147.891	369.297	30"
	7	0	100	0	100	1.281	3.030	0.0002	59.147186	561.348	808.351	32"
	8	0	100	0	100	1.167	2.761	0.128	56.429844	535.628	808.22	31"
	9	100	0	100	0	0.797	1.886	0.049	45.72015	306.568	604.763	30"
	10	100	0	100	0	0.486	1.149	0.0003	30.310116	-3.21259	320.578	37"
Hybrid MLP, inputs GA	*Layers*											
	1	*99.66*	*0.33*	*5.82*	*94.17*	*0.034*	*0.083*	*0.958*	*3.167633*	*-2224.53*	*-2135.24*	*51' 2"*
	2	99.16	0.83	52.42	47.57	0.191	0.470	0.746	12.983215	-923.615	-774.377	59' 35"
	3	100	0	100	0	0.432	1.062	0.183	26.092693	-329.259	-164.715	1h 25' 25"
	4	100	0	100	0	0.405	0.996	0.153	19.834393	-334.673	-144.61	1h 29' 04"
	5	100	0	100	0	0.557	1.371	0.095	35.649826	-54.8485	170.922	1h 40' 36"
	6	100	0	100	0	0.434	1.068	0.224	26.069294	-189.470	61.8107	1h 48' 26"
	7	100	0	100	0	0.513	1.263	0.097	32.985878	-48.4362	210.151	1h 44' 54"
	8	0	100	0	100	1.353	3.329	0.155	60.953899	637.915	909.605	2h 41' 22"
	9	100	0	100	0	0.537	1.32	0.037	34.459007	78.78643	406.6	2h 0' 45"
	10	0	100	0	100	0.866	2.13	0.344	48.194672	453.9556.	807.28	1h 47' 17"
Hybrid MLP, GA all layers	*Layers*											
	1	99.16	0.83	58.71	41.28	0.209	0.495	0.750	14.547907	-1065.435	-1041.1	2h 13' 16"
	2	96.31	3.68	18.34	81.65	0.162	0.384	0.798	11.69929	-1060.96	-918.91	2h 50' 06"
	3	100	0	92.66	7.33	0.396	0.938	0.253	18.93849	-535.591	1295.79	7h 29' 18"
	4	100	0	100	0	0.422	0.999	0.211	19.58418	378.505	1009.45	7h 43' 56"
	5	100	0	99.08	0.91	0.419	0.992	0.088	19.43412	79.4628	522.276	9h 44' 43"
	6	100	0	100	0	0.422	0.998	0.196	19.56872	2745.85	4892.8	2 h 39' 59'"
	7	100	0	100	0	0.422	1	-0.056	19.58313	370.563	996.389	13h 05' 54"
	8	100	0	100	0	0.422	0.999	0.174	19.58316	2818.56	5010.87	26h 49' 31"
	9	100	0	100	0	0.422	0.998	0.196	19.56876	2745.85	4892.09	21h 39' 59"
	10	100	0	100	0	0.422	0.999	0.128	19.583370	2076.55	3794.05	29h 57' 06"

Table 2. Results of Hybrid MLPs, [17], [8] (Part 2)

Neural Network		*Active Confusion Matrix*				*Performance*						
MLP Neural Network	*Layers*	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
Hybrid MLP, GA all, Cross Validation												
	1	99.15	1.84	25.68	74.31	0.149	0.352	0.804	8.08449	-1300.83	-1273.9	2h 32' 30"
	c.v.	98.32	1.67	31.19	68.80	0.167	0.395	0.780	7.922915	-1217.69	-1190.8	
	2	100	0	98.16	1.83	0.497	1.176	0.126	8.203679	-274.749	-135.25	2h 56' 17"
	c.v.	100	0	99.08	0.91	0.499	1.179	0.090	7.983900	-271.110	-131.68	
	3	99.49	0.50	62.13	37.86	0.217	0.535	0.723	15.057777	878.727	2119.82	11h 31' 29"
	c.v.	99.83	0.16	65.13	34.86	0.230	0.544	0.734	14.784730	911.851	2156.4	
	4	100	0	99.08	0.91	0.418	0.990	0.099	19.432472	1459.53	2786.69	16h 22' 59"
	c.v.	100	0	100	0	0.422	0.998	0.064	19.47684	1467.29	2793.72	
	5	100	0	100	0	0.422	1	-0.018	19.575001	1002.56	2832.81	26h 09' 45"
	c.v.	100	0	100	0	0.423	1	-0.063	19.712772	1004.27	2033.94	
	6	100	0	100	0	0.422	0.999	0.033	19.583233	1754.55	3266.01	27h 43' 20"
	c.v.	100	0	100	0	0.423	0.999	0.020	19.589403	1756.23	3266.85	
	7	100	0	100	0	0.422	0.999	0.061	19.583591	1946.557	3580.87	43h 52' 18"
	c.v.	100	0	100	0	0.423	0.999	0.102	19.589190	1948.20	3581.61	
	8	100	0	100	0	0.423	1	-0.023	19.589117	160.2418	650.13	22h 08' 52"
	c.v.	100	0	100	0	0.422	1	-0.008	19.583158	158.563	648.72	
	9	100	0	98.16	1.83	0.414	0.980	0.141	19.357584	780.613	1677.75	30h 25' 37"
	c.v.	100	0	99.08	0.91	0.419	0.989	0.101	19.434509	789.1412	1685.78	
	10	100	0	100	0	0.423	1	-0.274	19.583002	2380.58	4292.61	15h 02' 44"
	c.v.	100	0	100	0	0.423	1	-0.314	19.588856	2382.25	4293.22	

Table 3. The Voted Perceptron results [17], [8], [3]

0→0	*0→1*	*1→0*	*1→1*	*Misclassifications*	*Correct classif.*	*K-stat*	*MAE*	*RMSE*	*RAE*	*RRSE*	*Time*
601	5	59	40	64 9.07%	641 90.92%	0.5128	0.0908	0.3013	34.60%	86.42%	0.17 sec

Table 4. Overall ranking of the optimal GFF models

Models		*Active Confusion Matrix*				*Performance*						
	Layers	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
GFF input-output GA	1	98.90	1.085	11.465	88.52	0.072	0.170	0.908	5.776	-1907.09	-1796.44	3h 19' 25"
GFF GA all	3	97.14	2.845	17.885	82.10	0.128	0.304	0.834	8.343	-786.38	284.34	4h 20' 25"
	1	97.56	2.425	18.805	81.18	0.133	0.315	0.827	8.243	-723.47	-271.82	3h 19' 25"
GFF GA all,	7	96.64	3.35	19.26	80.73	0.136	0.323	0.825	9.119	1541.07	3429.31	25h 46' 34"
CV		98.32	1.67	29.355	70.63	0.149	0.353	0.812	7.023	1608.29	3495.49	
GFF NN	1	97.73	2.26	21.095	78.89	0.138	0.328	0.821	9.675	-1225.82	-1111.95	14"
GFF NN, CV	8	98.23	1.755	26.14	73.85	0.143	0.338	0.814	9.284	709.44	2041.35	42.5"
CV		98.23	1.755	26.14	73.85	0.143	0.338	0.814	9.284	709.44	2041.35	
GFF GA inputs	10	97.98	2.005	26.6	73.16	0.144	0.341	0.812	9.469	1219.39	2873.69	7h 44' 32"
GFF GA all	8	98.57	1.42	26.6	73.39	0.14	0.329	0.821	8.329	1262.65	2959.69	29h 50' 17"
GFF GA all, CV	1	97.98	2.005	24.305	75.68	0.145	0.343	0.810	8.646	-1219.07	-1126.3	2h 27' 41"
CV		98.4	1.59	24.765	75.22	0.139	0.330	0.821	8.686	-1242.55	-1149.79	
GFF NN	10	98.65	1.34	31.185	68.80	0.147	0.348	0.811	8.454	1557.50	3419.165	57"

Table 5. Optimal outcomes MLP NN, Hybrid MLP, and the Voted Perceptron, [17], [3]

Models	***Active Confusion Matrix***					***Performance***						
	Layers	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
Hybrid MLP, inputs GA	1	99.66	0.33	5.82	94.17	0.034	0.083	0.958	3.167633	-2224.53	-2135.24	51' 2"
Hybrid MLP, GA all, CV	1	99.15	1.84	25.68	74.31	0.149	0.352	0.804	8.08449	-1300.83	-1273.9	2 h 32' 30"
c.v.		98.32	1.67	31.19	68.80	0.167	0.395	0.780	7.922915	-1217.69	-1190.8	
Hybrid MLP, GA all lay.	2	96.31	3.68	18.34	81.65	0.162	0.384	0.798	11.69929	-1060.96	-918.91	2 h 50' 06"
		0→0	*0→1*	*1→0*	*1→1*	*MAE*	*RMSE*	*K-stat*	*RMSE*	*RAE*	*RRSE*	*Time*
Voted Perceptron		85.24	0.7	8.36	5.67	0.0908	0.3013	0.5128	0.3013	34.60%	86.42%	0.17"
	Layers	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
MLP Neural Network	1	100	0	98.16	1.85	0.417	0.986	0.121	19.399534	-471.148	-377.72	22"

Table 6. Overall Optimal results of the FeedForward Models: VP, MLPs, GFFs

	Layers	*0→0*	*0→1*	*1→0*	*1→1*	*MSE*	*NMSE*	*r*	*%error*	*AIC*	*MDL*	*Time*
Hybrid MLP, inputs GA	1	99.66	0.33	5.82	94.17	0.034	0.083	0.958	3.167	-2224.53	-2135.24	51' 2"
GFF input-output GA	1	98.90	1.08	11.46	88.52	0.072	0.170	0.908	5.776	-1907.09	-1796.44	3h 19' 25"
GFF GA all	3	97.14	2.84	17.88	82.10	0.128	0.304	0.834	8.343	-786.38	284.34	4h 20' 25"
GFF GA all	1	97.56	2.42	18.80	81.18	0.133	0.315	0.827	8.243	-723.47	-271.82	3h 19' 25"
Hybrid MLP, GA all, CV	1	99.15	1.84	25.68	74.31	0.149	0.352	0.804	8.084	-1300.83	-1273.9	2h 32' 30"
CV		98.32	1.67	31.19	68.80	0.167	0.395	0.780	7.922	-1217.69	-1190.8	
Hybrid MLP, GA all lay.	2	96.31	3.68	18.34	81.65	0.162	0.384	0.798	11.699	-1060.96	-918.91	2h 50' 06"
		0→0	*0→1*	*1→0*	*1→1*	*MAE*	*RMSE*	*K-stat*	*RMSE*	*RAE*	*RRSE*	*Time*
Voted Perceptron		85.24	0.7	8.36	5.67	0.090	0.3013	0.512	0.3013	34.60%	86.42%	0.17"
	Layers	***0→0***	***0→1***	***1→0***	***1→1***	***MSE***	***NMSE***	***r***	***%error***	***AIC***	***MDL***	***Time***
MLP Neural Network	1	100	0	98.16	1.85	0.417	0.986	0.121	19.399	-471.14	-377.72	22"

The Hybrid MLP of 1 layer in GA optimization of all layers and Cross Validation had an optimal performance, where 99.15% of the healthy firms were correctly classified, and 74.31% of the distressed in the initial classification whilst the CV classifications was altered in 98.32% and 68.80% respectively, though the MSE was at 0.149, the fitness to the model at 0.804, in 8.08% error, converging in 2 hours 32 minutes and 30 seconds. These models are more reliable as the Cross Validation reassures non exposure to over-optimisation.

Finally the Hybrid MLP that is optimised in all layers by GA offered the optimal performance on the 2 layers network, in 96.31% healthy companies correctly classified and 81365% distressed, as the error reached 0.162 in MSE, and 0.384 in NMSE, whilst the fitness of the model was high at 0.798, in a significant percentage error at 11.69% consuming a considerable amount of time at 2 hours 50 min., 6 s.

9.3. Results of the Voted Perceptron

The Voted Perceptron [17], [8], [3], had a moderate performance since the classification offered a 90.92% to the correctly classified companies by the executives, and 9.07% misclassifications, whilst the interobserver agreement level at 0.51 was adequate, and the errors level was significant as MAE was 0.09. The processing time was very fast at 0.17 seconds (see Table 3).

As [17], [3] concluded, The most efficient method among all the different types examined in this research is the Hybrid MLP neural network with genetic optimisation into the inputs layer with 1 hidden layer that had the optimal overall performance of all heuristic models examined. Its classification results were superior, in the highest fitness of the data to the model, providing a very low error, and an adequate computing time of 51 minutes approximately. In to the second place was ranked the Hybrid MLP Networks of 1 layer with Genetic Algorithms optimising all their parameters and Cross validation, whilst an almost identical to that model was the Hybrid MLP Networks of 2 layers and GAs that optimise all of their layers. The fourth place was taken by the Voted Perceptron network of an adequate performance in terms of fitness to the data and error. The last rank was given to the MLP Neural Network of 1 layer with quite inferior results to all the model types.

9.4. Results of the Generalised Feed Forward models

The GFF hybrid of I layer in GA optimization on the inputs and outputs layers only had the optimal performance in high convergence on the correct classification of the healthy and the distressed companies, 98.9% and 88.52% respectfully, very high fitness of the model to the data as r was 0.908, the lowest error as MSE was 0.072, NMSE 0.170 and percentage error 5.77, whilst the AIC was very low at -1907.09 indicating impartiality and a time of 3h 19m 25s.

The next rank was taken by the hybrid GFF of 3 layers in GA optimization in all layers, in an almost fine classification, a high fitness of the model to the data at r 0.834, low error, and impartiality in 4h 20m 25s.

A slightly inferior performance had the hybrid GFF of 1 layer and GA optimization in all layers, in terms of classification, fitness, and error, whilst the impartiality was higher, in a time of 3h 19m 25s (see Table 4, 5, 6).

10. CONCLUDING REMARKS

The EPICO model is an integrated system on the portfolio selection problem. Its module-based structure provides a flexible platform for demanding environments. The fundamentals filter the fraud more effectively. The complex isoelastic utility function, is suited for customer-tailored solutions in higher efficiency. The optimal classifier of the FeedForward family was the Hybrid MLP of 1 layer GA optimization into inputs only, in a fine classification, an ideal fitness to the model at 0.958, the least error, impartial processing, in a fast time.

The Hybrid GFF of 1 layer and GAs on the inputs and outputs got the second rank of the FF family, in a lower classification performance, fine fitness, higher error, impartiality, in a higher computing time.

Thirdly the Hybrid GFF of 3 layers and GA optimization in all the layers had a quite good classification, good fitness, slightly higher error, impartial model, and a higher processing time.

The research is not limited by the data parameters as far as the model is connected on line to the internet, or more specialized data bases. Future directions could include the infiltration of publicly shared opinions (sentiments) of investors so that the model has a stronger point on the behavioral aspect.

REFERENCES

1. Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance, 7(1),* 77–91.
2. Maringer, D., & Parpas, P. (2009), Global optimization of higher order moments in portfolio selection. *Journal of Global Optimization, (43),* 2-3.
3. Loukeris, N., & Eleftheriadis, I. (2015b). Further Higher Moments in Portfolio Selection and A-priori Detection of Bankruptcy, under Multi Layer Perceptron Neural Networks, Hybrid Neuro-Genetic MLPs, and the Voted Perceptron. *International Journal of Finance and Economics, 20(4).*
4. Lippman R. (1987). An introduction to computing with neural nets. *IEEE Assp magazine, 4,* 4-22.
5. Holland, J. H. (1975/1992). *Adaptation in Natural and Artificial Systems.* Cambridge, MA, MIT Press, Second edition (1992). (First edition), University of Michigan Press.
6. Kristouflek, L. (2013). Fractal markets hypothesis and the global financial crisis: Wavelet power evidence. Scientific Reports, 3, 2857, doi:10.1038/srep02857.
7. Subrahmanyam, A. (2007).Behavioral Finance: A Review and Synthesis. *European Financial Management, 14(1),* 12–29.
8. Loukeris, N., Eleftheriadis, I., & Livanis, S. (2014a). Optimal Asset Allocation in Radial Basis

Functions Networks, and hybrid neuro-genetic RBFNs to TLRNs, MLPs and Bayesian Logistic Regression. World Finance Conference, July 2-4, Venice, Italy.
9. Loukeris, N., Eleftheriadis, I., & Livanis, E. (2014b). Portfolio Selection into Radial Basis Functions Networks and neuro-genetic RBFN Hybrids. IEEE 5th Int. Conference IISA, July 7-9, Chania Greece.
10. Aguiar-Conraria, L., Azevedo, L., & Soares, M. (2008). Using wavelets to decompose the time-frequency effects of monetary policy. *Physica A*, 387, 2863–2878.
11. Rua, A., & Nunes, L. (2009) International Co-Movement of Stock Market Returns: A Wavelet Analysis. *Journal of Empirical Finance, 16,* 632 639.
12. Vacha, L., & Barunik, J. (2012). Co-movement of energy commodities revisited evidence from wavelet coherence analysis. *Energy Economics*, *34,* 241–247.
13. Vacha, L., Janda, K., Kristoufek, L., & Zilberman, D. (2013). Time-frequency dynamics of biofuel-fuel-food system. *Energy Economics*, 40, 233 241.
14. Freund Y., & Schapire, R. (1997). A decision theoretic generalization of online learning and application to boosting. European Conference on Computational Learning Theory, 1995. *Journal of Computer and System Sciences 55, 119-139.*
15. Freund Y. & Schapire, R. (1996*). Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference*, 148-156.
16. Freund, Y., & Schapire, R. (1999). Large margin classification using heperceptron algorithm. *Machine Learning*, *37(3),* 277-296
17. Loukeris, N., & Eleftheriadis, I. (2012a). Bankruptcy Prediction into Hybrids of Time Lag Recurrent Networks with Genetic optimisation, Multi Layer Perceptrons Neural Nets, and Bayesian Logistic Regression, Proc. Int. Summer Conference of the International Academy of Business and Public Administration Disciplines (IABPAD), Honolulu, Hawaii, USA (August 1- 5) - Research Paper Award.
18. Loukeris N. (2008). Comparative Evaluation of Multi Layer Perceptrons to hybrid MLPs with Multicriteria Hierarchical Discrimination and Logistic Regression in Corporate Financial Analysis. 11th International Conference on Computers CSCC, Elounda Agios Nikolaos, Crete, Greece 26-28 July.
19. Hornik K., Stinchcombe M., & White, H., (1989). Multilayer feedforward networks are universal approximators, *Neural Networks*, *2(5),* 359-366.
20. Lapedes A. & Farber, R. (1987). Nonlinear signal processing using neural networks: prediction, and system modelling. LA-VR-87-2662, Los Alamos.
21. Makhoul J. (1992). *Pattern recognition properties of neural networks, Proc. 1991 IEEE Workshop on Neural Networks for Signal Processing*, 173-187.
22. Rumelhart, D., Hinton, G., & Williams, R. (1986). *Learning internal representations by error back-propagation, in Parallel distributed processing: explorations in the microstructure of cognition.* (Rumelhart, D. & McClelland, J., eds.), MIT Press, Cambridge, MA.
23. Principe J., deVries B., Kuo J., & Oliveira P., (1992*), Modeling applications with the focused gamma network, Neural Information Processing Systems 4.* (eds. Moody, Hanson, Touretsky), 121-126, Morgan Kaufmann.
24. Principe J., Euliano, N. R, & Lefebvre, W. C. (1999). *Neural and adaptive systems: Fundamentals through simulations.* New York: Wiley.
25. Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale Bayesian logistic regression for text categorization. *Technometrics*, *49,* 291-304.
26. Courtis, J. K. (1978). Modeling a Financial Ratios Categoric Framework. *J. Bus.Fin. & Acc., 5(4),* 371-386.
27. Loukeris N., & Matsatsinis, N. (2006a). Corporate Financial Evaluation and Bankruptcy Prediction implementing Artificial Intelligence methods. *WSEAS, Transactions of Business and Economics*, *4(3).*
28. Loukeris, N., & Matsatsinis, N. (2006b). Hybrid Neuro-Genetic Systems as Effective Analysis schemes of Financial statements. *WSEAS, Transactions on Business and Economics, 5(3).*