Review of an ANFIS Methodology-Based Stock Market Prediction System

Manal Alghieth
Faculty of Computer, Information Technology
Qassim University,
Qassim, Saudi Arabia
Email: mgietha@qu.edu.sa

***Abstract*—Stock market prediction is of immense interest to trading companies and buyers due to high profit margins. The majority of successful buying or selling activities occur close to stock price turning trends. This makes the prediction of stock indices and analysis a crucial factor in the determination whether the stocks will increase or decrease the next day. This paper describes an Adaptive Network based Fuzzy Inference System (ANFIS) and critically analyses its ability to improve prediction in Yahoo stock data. At present, the focus of research is on the improvement of prediction with low false prediction via the hybridization and extension of existing methodologies. The research results presented a low Mean-Square-Error (MSE) in both testing and validation processes.**

Keywords- Adaptive Network-Based Fuzzy Inference System (ANFIS); Prediction; Time series Stock market prediction; Yahoo! stock data.

I. INTRODUCTION

Stock price forecasting has long been a focus of intelligent soft computing techniques to improve the predictability of financial systems [1]. Due to rapidly changing trends in current global financial markets and the ongoing commercial uncertainties, accurate forecasting of time-based financial trends has become increasingly important. Stock market forecasting provides the investors with a general overview of the changing tendency of the stock markets. Based on the forecasts, the investors can make timely decisions on buying or selling stocks under bargains and avoid financial losses. A wide range of techniques applicable to stock market forecasting have been reported in the literature which are not just limited to econometric modelling but includes Artificial Intelligence (AI) – based soft-computing techniques- as well [2]. Indeed, Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS) are two well-known paradigms used in time-series design and prediction, and have their own strength and weaknesses in the forecasting of future data based on a finite set of previous time-based trends [3].

Research in fuzzy logic has drawn substantial attention during the past two decades and has now become a robust paradigm for the prediction of nonlinear and uncertain systems from a wide range of real-world domains including signal data mining [4], information retrieval [5], finance [6] and various real-world forecasting systems including stocks, resource demand and supply, power requirement, and sensor networks [7]-[10]. Despite a continued and high demand of this soft-computing technique, a number of limitations can be associated to it. FIS generally require a great deal of human intervention to accurately and realistically predict certain situations, which induces a high chance of human-based error in the system. Moreover, the increase in the system variables increases substantially the complexity of the system.

The majority of real-world forecasting systems cover application areas that require the knowledge of historic values to be incorporated into the model. This is because the outcome crucially depends upon historic data. Share prices, electricity consumption and weather forecasts are a few examples of such systems. Statistical Analysis (SA), ANN, Case Based Reasoning (CBR), FIS, Decision Trees (DT) and Support Vector Machines (SVM) are examples of a number of soft-computing and machine learning methodologies that are frequently used to implement time-series-based forecasting systems. A comprehensive review of applications of these techniques to financial time-series share market forecasting can be consulted in [38]. This review revealed ANNs to be the most frequently used technique in the financial forecasting sector followed by rough set (RS) theory, CBR, OR, FIS and SVM techniques. At present, the focus of research is on the improvement of prediction with low false prediction via the hybridization and extension of existing methodologies.

ANNs generally operate over an undefined dataset where, when subjected to training data, the technique learns from irregularities and thereby creates its own set of rules. The methodology heavily emphasises on comprehensiveness of data and, unlike fuzzy logic, is well known for its ability to withstand noisy data and outliers [11]. The methodology has the ability to predict missing, sparse or low-quality values, which makes it suitable for financial systems that meets with uncertain data. Moreover, this methodology is well known for its ability to handle input variables in parallel and thus it allows large datasets to be efficiently handled. These characteristics make ANN unique in its ability to generalise over a diverse range of input/output data pairs, making it an ideal candidate to replace the human-based expert rule-generation in fuzzy systems. Yet, this paradigm still has its own disadvantages in that over-training may result in unstable prediction capabilities. This shortcoming is generally overcome by dividing the dataset into three groups of training, test and validation sets where the algorithm is stopped if its error margin repeatedly increases over a consecutive number of iterations.

II. LITERATURE REVIEW

A. Prediction Systems in Literature

Time-series is regarded as a sequence of stochastic variables whose behaviour depends upon a number of real-world factors or dependent variables that decide the values of the next variables ahead of time based on past trends [12]. A number of soft-computing prediction methodologies have been reported in the literature, which are generally classified as statistical or AI–based domains type. Time-series analysis provides tools to select models that are then used to predict future events as a statistical time-series problem. These statistical predictions are based on the notion that the observations are based on a probability distribution function. Supporting and hybrid models are extensively reported in the literature to improve the forecasting performance via ANN classifiers [13] and network data flow prediction [14], signal synthesis [15], independent component analysis [16], locally linear embedded (LLE) in multivariate analysis [17] and logistic regression [18]. These statistical modelling algorithms are generally limited on the number of variables used and also tend to demonstrate increasing computational complexities with larger datasets. This is the reason why the majority of these models are used in conjunction with supporting soft-computing techniques including self-organising feature maps [19], Linear and Multiple Discriminant Analysis (LDA/MDA) [20], learning-vector quantization [21], case-based-reasoning, rough-sets, linear and quadratic programming and Support Vector Machines (SVM) [22].

Despite the multitude of techniques available, the scope of this research focuses on two predominant AI paradigms in a bid to improve the overall prediction accuracy of the underlying system. As mentioned earlier, ANNs are known for their capabilities to understand and predict patterns in serial data whereas the FIS provides a platform to embed expert human knowledge thereby improving the overall prediction accuracy of uncertain, real-world systems. Based on their limitations and strengths, the next two subsections present their current state-of-the-art in order to elaborate further on various avenues of improvement.

B. Fuzzy classifiers in time-series-based financial forecasting

A tri-classifier clustering approach was implemented by Chang et al. [23] as a fuzzy neural network approach which segmented training data into historical clusters in an apparent bid to reduce the training overhead and predict short-length cases via a larger 5-yearly dataset. The approach claimed improved outcomes when compared to the proposed ANFIS methodology based on the forecasted Root-Mean-Squared-Errors (RMSE). Li et al. [24] presented a genetic particle swarm clustering methodology combined with a fuzzy c-means algorithm in a bid to use gradient method to improve the overall accuracy. Similar to other hybrid time-series systems, this methodology also presented high execution times when subjected to larger and multi-dimensional datasets.

A number of direct neuro-fuzzy approaches have been reported in literature with Tung et al. [25] using financial covariates, Yoshida [26] utilising the Black-Scholes formula, Castillo and Melin [27] reporting via fractal dimensions and Tang and Chi [28] using ROC analysis with Logit performance to improve time-series prediction with promising improvements.

The Taguchi method has been used in a number of forecasting investigations [23], [29]. The focus has predominantly been on the utilization of Grey Relational Analysis (GRA) and the utilization of Grey Extreme Learning Machine (GELM) technique against General Back Propagation Neural Networks (GBPN) methods. The methodologies have also been used to predict the most optimal number of neural parameters for improved prediction rate. However, there is a consensus that an increase in the optimization parameters for these algorithms to control hidden nodes, layers and activation functions generally result in a reduced overall performance of the system being optimised.

C. Neural Systems in time-series-based financial forecasting systems

As discussed earlier, ANNs are known to improve prediction accuracies of time-series-data forecasting systems in financial and other trading applications. Their ability to generalise in the presence of noisy feature sets and outliers makes them ideal for share market price prediction, asset allocation and portfolio change forecasting.

Martinetz et al. [30] compared an unsupervised technique based on K-means clustering against methodologies including Kohonen-maps, K-means and Maximum-entropy. The classifier presented outstanding minimization in vector quantization coding distortion error and a faster convergence at a controllable cost of higher computational effort. ANNs were initially employed by Connor et al. [31] with outliers “softly” removed from the data when the training was performed over the “outlier-filtered” data. This technique substantially improved the prediction accuracy of the system. However, in large-scale real-world systems, it is generally impractical to use “pre-training” clustering techniques for outlier removal. Moreover, there is a high probability that such a technique may also eliminate valid feature samples from the database as well. In order to address this issue, a hybrid ANN technique was proposed by Castillo and Melin [27] via a neuro-fuzzy technique. The technique regulated the fuzzy membership functions by means of a single-layer feed-forward neural network. The outcome of this work was far superior than the one obtained with generalised regression-based models. A similar work by Zhang and Berardi [32] utilised varied ANN structures over varied data partitions via varied initial random weights, random architectures and variable data and reported a considerable accuracy over conventional neural architectures.

Research has lately moved into the analysis of noisy chaotic time-series prediction. According to Soofi and Cao [33] chaotic and non-linear time series prediction

 ISBN: 978-1-61208-695-8

has a significant effect on the economic and financial time series prediction. This is particularly prevalent in stock market prediction where the nonlinear feature data is normally marred by excessive noise. Leung et al. [34] addressed the optimum prediction of noisy time-series data via a Radial Basis Function (RBF) neural network classifier, where the issue of generalization against a large dataset was tackled using a "cross-validated sub-space" method to identify a suitable number of hidden neurons to efficiently handle noise within the datasets. Recently, in-architecture neural network updates have been explored with Goh [35] creating a neuron-level hyper-plane to separate noise from genuine feature samples. This technique, when combined with the nonlinear subspace, creates an optimal RBF predictor for variable signal-to-noise ratios (SNR).

Improvements in the neural architecture also involve the utilization of the so-called "recurrent" ANN (RPNN) that facilitates long-term prediction [36] and local linear and wavelet-based transforms [37]. Additionally, generalised regression-based ANN, counter-propagation technique, neural adaptive resonance classifiers, CART DT, TreeNet-based data mining and random forests have also been used [38].

III. DESIGN AND ANALYSIS

FIS can be classed as of Mamdani type or Takagi-Sugeno Kang (TSK) type. Mamdani FIS is mostly used in practice, although TSK FIS is well known for its computational efficiency and compactness, and it derives a set of rules from input/output training data pairs. Indeed, an important aspect of TSK FIS is its crisp outcome, which significantly reduces its computational complexity when compared to its Mamdani counterpart. A typical TSK FIS rule is given below:

$$i : IF\ x\ is\ A_i\ and\ y\ is\ B_i\ THEN\ f_i = p_i(x) + q_i(y) + r_i \quad (1)$$

where i stands for the rule number, A_i and B_i are corresponding fuzzy sets to each linguistic label domain, f_i is the output set covered by the fuzzy rule in the fuzzy region and p_i, q_i and r_i are the design parameters.

In the equation (1), the values for parameters p_i, q_i and r_i are obtained by training input/output pairs via an ANN.

First order TSK FIS can be defined and visualised as a moving pointer that moves linearly in an outer space based on the value of the antecedent variables. As each rule in the FIS database is associated to the input variables, the TSK FIS is suitable for systems requiring interpolation of multiple linear inputs. A Sugeno system interpolates linear gains from multiple input parameters that would be applied across the input space. This gives a Sugeno system a smooth curve-based change, which is very close to real-world conditions. For instance, due to input-space interpolation, a Sugeno model demonstrates a Gaussian transition between various states. A real-world example of this phenomenon can be that of a temperature control and monitor mechanism in a boiler system where a Sugeno type controller is used to adjust power levels when temperature changes. Instead of defining heat conditions as Very High, High, Medium, Low and Very Low, a Sugeno system can actually interpolate the intermediate values to show an asymptotic decline or incline from very hot to very cold conditions (Matlab, R2014b).

A. FIS rule-base generation via subtractive clustering and grid-partitioning

Expert engineers with in-depth knowledge of the underlying domain generate FIS rules, either when a good database is not available or does not cover the whole modelling scenario. However, in order to generate a comprehensive rule-base that portrays the exact relationship between the input/output feature sets, the variable space must be efficiently clustered.

In a fuzzy c-means clustering algorithm, each data point belongs to each of the clusters based on some degree of membership. Therefore, the closer a point is to the mean position of a cluster, the higher its membership to that cluster is. For instance, the weight of a person may be attributed to two different clusters of individuals with one cluster classified as those being obese and the other being of average weight. Based upon a specific data point's (person's) weight's distance to the centre point of both of these clusters, the data points membership could be 0.33 Obese and 0.67 Average_Weight, effectively assigning him/her to belong predominantly to an Average_Weight cluster.

In the time-series-based stock value prediction case, rules are drawn from multiple variables including opening, high, low and trading volume values. These variables can be bound to the input-space via a number of partitioning methodologies including grid [39], tree [40] and scatter partitioning [41]. Grid-based clustering is generally deemed appropriate for systems with low number of membership functions and input variables. This is primarily due to the fact that the methodology's computational complexity increases exponentially with the increase in the number of membership functions and input variables (Mathworks, 2014b).

A complete FIS with the proposed two input variables, trade volume (ϑ_t) and stock value (δ_t) at a time-instance t, and three membership functions, namely LOW, MEDIUM and HIGH consists of a total number of 9 rules. In general, a complete FIS with p input variables, each one with its domain divided into $N_1, \ldots, N_p$ fuzzy labels, will consists of the following number of rules (2):

$$N_1 * N_2 * \ldots * N_p \quad (2)$$

When all input variables are associated the same number of linguistic labels (N) then the total number of rules possible is p^N, and therefore the number of rules will increase exponentially with respect to the number of input variables and the number of linguistic labels. To reduce the number of rules, alternative techniques such as subtractive clustering was proposed on the basis of a single-pass algorithm for number of cluster and centre estimation [42].

 ISBN: 978-1-61208-695-8

B. Formulation of a neuro-fuzzy approach for financial time-series estimation:

The proposed system implements a neuro-fuzzy approach where the ANN technique is used to tune the FIS parameters. The resultant methodology is widely known as an Adaptive Network based Fuzzy Inference System (ANFIS), which utilises training feature data to induce fuzzy rules via neural training-based weight adjustment.

A wider framework for the proposed TSK ANFIS to predict stock prices is shown in Figure 1, where each layer is further explained below:

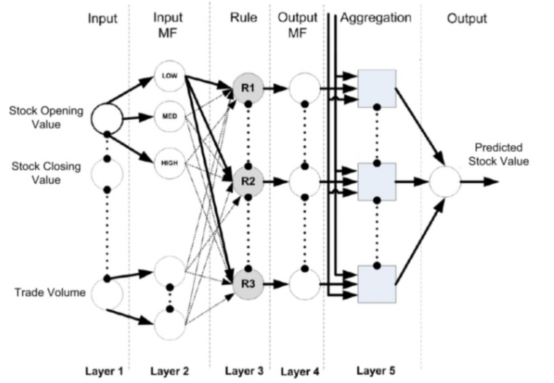

Figure 1. The design of a proposed TSK FIS based ANFIS framework utilising 4 input variables, respective input membership functions, rules and aggregation as hidden layers and output stock prices as the predictive outcome.

Layer – 1: Calculation of membership values for the premise parameter

The nodes in this layer are adaptive and the node output is the extent up to which the given input fulfils the underlying (associated) linguistic variable associated with this node as per the following expression:

$$\mu\, A_i(x_1) = {}^{1}\!\Big/_{1 + \left|x_1 - {}^{c_i}\!/_{a_i}\right|^{2b_i}} \tag{3}$$

where x_1 is the input to the node and a, b, c are adjustable factor variables termed as premise parameters. The layer outputs the membership values of the premise part where an ANN back propagation algorithm is used during the learning stage. The premise parameters are used to define membership functions that are generally fine-tuned via a Gradient-Descent method. As the subsequent values of the parameters change, the linguistic term's membership function $\mu\, A_i(x_1)$ changes as well. That is, the closer a parameter is to a certain membership, the clearer its association to a certain group is. In other words, the membership grade of a fuzzy set specifies the degree up to which the given input satisfies the quantifier. As shown in Figure 2 as the value of the parameters change between parameters a_1, a_2 and a_3, its membership projection (see y axis) changes between 0 and 1.

In the proposed stock price prediction problem, if closing price at time instance t is δ_t^i, which is an input variable with three membership values of HIGH, MEDIUM and LOW, then the three nodes are kept in the Layer – 1 and denoted via various membership function types.

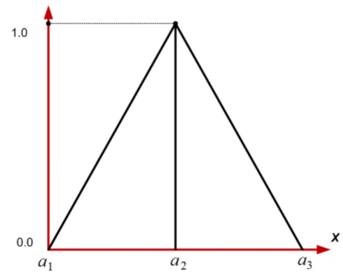

Figure 2. A triangular membership function used for prediction.

For the proposed case of close, low, open and volume variables, the membership functions can be formulated as follows:

$$\mu_{(\delta,\nu,\mathrm{X})} = \begin{cases} 0,\ x < a_1 \\ {}^{x - a_1}\!/_{a_2 - a_1},\ a_1 \le x \le a_2 \\ {}^{a_3 - x}\!/_{a_3 - a_2},\ a_2 \le x \le a_3 \\ 0, x > a_3 \end{cases} \tag{4}$$

As an example, if the value of $x = 3.5$ then its membership value would be 0.75, which is calculated as follows:

$$ {}^{x - a_1}\!/_{a_2 - a_1} = {}^{3.5 - 2}\!/_{4 - 2} = {}^{1.5}\!/_{2} = 0.75$$

Layer – 2 : The fuzzification layer

In Layer – 2, the nodes are kept fixed with each expressing one linguistic variable (e.g., MEDIUM) mapped to one input variable in layer 1. The output at this layer is a membership value specifying the extent up to which an input variable belongs to a specific set. This extent is also regarded as the firing strength of the rules, and it is obtained by multiplying the input signals from the preceding layer (ANFIS 2013):

$$\omega_1 = \mu\, A_i(x_1)\mu\, B_i(x_2) \tag{5}$$

For instance, for a FIS containing 3 rules with each containing membership values (See calculation shown previously in Layer 1) as Rule 1: if $x\ is\ A1\ and\ y\ is\ B1\ then\ f1 = p1x + q1y + r1$ $\omega_1 = 0.75 \text{ x } 0.67 = 0.5025$. Similarly, for assumed Rule 2: if $x\ is\ A2\ and\ y\ is\ B2\ then\ f2 = p2x + q2y + r2$ $\omega_2 = 0.25 \text{ x } 0.33 = 0.0825$ and Rule 3: if $x\ is\ A3\ and\ y\ is\ B3\ then\ f3 = p3x + q3y + r3$ $\omega_3 = 0.25 \text{ x } 0.3 = 0.075$. Based on the rule firing values, rule 1 will fire as it has the highest weight value.

A complete Layer – 2 with 4 variables and three linguistic labels each will require a total of $3^4 = 81$ rules. The rule strength is calculated where a clustering algorithm decides the initial number and type of membership function to be allocated to each of the variable type.

Layer – 3: Rule-strength normalization:

The output to this layer, represented by a fixed number of nodes, is the rule's antecedent part that is the firing strength of the fuzzy rule in its normalised form represented as a $\mathrm{t-norm}$. The $\mathrm{i^{th}}$ node in this layer calculates the $\mathrm{i^{th}}$ rule's firing strength ratio to the firing strength of the sum of all rules as follows (ANFIS 2013).

$$\overline{\omega}_i = \frac{\omega_i}{\sum_{j=1}^{R} \omega_j} \tag{6}$$

where ω_i is the firing strength of the $\mathrm{i^{th}}$ rule computed in the previous Layer – 2. Following-up from the previous 3-rule example, the normalization (for Rule 1) is as follows:

$$\overline{\omega}_1 = \frac{\omega_1}{\sum_{j=1}^{3} \omega_j} = \frac{0.5025}{0.5025 + 0.0825 + 0.075} = \frac{0.5025}{0.66} = 0.7613$$

Layer – 4: The Rule-Consequent Layer

The nodes in this layer are not fixed and adaptively change where, for every $\mathrm{i^{th}}$ node, a linear function is computed whose coefficients are adapted by an error function. The error function is a multi-layer feed-forward neural network as described below:

$$\overline{\omega_i} * f_i = \overline{\omega_i} * (p_i x_1 + q_i x_2 + r_i) \tag{7}$$

where $\overline{\omega_i}$is the weight output of the input layer (Layer – 2), whereas $\mathrm{p_i, q_i, r_i}$ are the parameter set where i represents various the total inputs to the system. These parameters are also called the "consequent parameters" where at this stage the overall subsequent output is computed by summing all the input signals. Thus, the final output for the given input in Layer-1 will be:

$$\sum_i \overline{\omega_i} f_i = {\sum_i \omega_i f_i}\Big/{\sum_i \overline{\omega_i}} \tag{8}$$

Clearly (8) demonstrates the ability of a multivariate time-series system based on a sliding-window.

IV. "YAHOO" CASE STUDY

The Yahoo dataset is regarded as a standard stock dataset and it is widely used as a benchmark to evaluate a wide range of machine learning algorithms. The sample stock data to explain the underlying concept was downloaded from Yahoo! Finance [43]. The data contains a daily trading of stock volume and prices from 12/04/1996 to 31/08/2012 consisting of the following five parameters:

- Open (share price)
- Low (share price)
- High (share price)
- End-of-day Close (share price)
- Volume (trade volume in US$)

The data is extracted for adaptive neuro-fuzzy training based on a sliding-window operation: Based on the single-step (one-day) sliding window operation, a feature vector containing a set of input vectors and the output (closing value) will be obtained in a row-wise fashion. Each row represents a single day prediction based on the previous 'n' number of days.

This study uses experimental data from Yahoo Finance to evaluate the performance of the proposed methodology. The closing, low and high stock values for the entire duration are shown in Figure 3, which shows substantial fluctuations in stock market values during the daily operating hours. This measures a significant justification for the utilization of all the four (i.e., close, low, high and adj close) values in classifier training in addition to the trading volume measure. The justification lies in the fact that the opening stock price of a share may substantially change by the end of the trading day and may therefore change the closing stock price drastically.

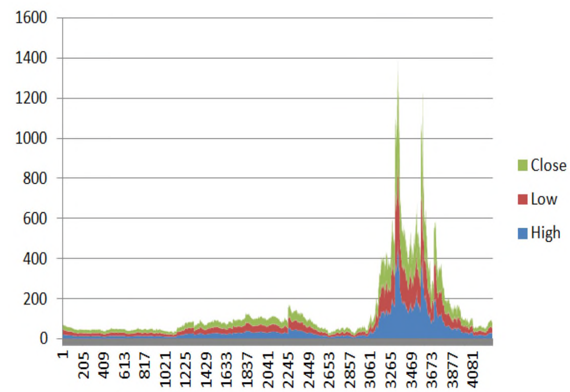

Figure 3. Closing, low and high share value limits during the entire 18-year duration of the stock market data.

The system was trained against the AForce.Neuro neural computation library with extensions made to the AForge.Fuzzy computations library for the hybridised implementation of the ANFIS framework. The training was based on a 10-day-delay with 10 neurons via a nonlinear autoregressive classification [44]. The data was divided into three randomly selected groups with training, testing and validation data selected at 75%, 15% and 15%, respectively. The 75-15-15 is a standard machine learning training practice used in research that was adopted from standard Matlab ANN toolbox (Matlab, 2014a). It must be noted that validation data group was only used to measure network generalization where the training was halt if the generalization stopped improving for at-least 5 consecutive epochs. An epoch in ANN terminology is the completion of a single training iteration leading either to the termination of the training sequence or the start of the next iteration based upon the criteria set in the initialization stage of the training process. The data division left 17980 target time steps of data for training, and 3853 days each for validation and testing purposes. The non-linear auto-regression for this training is described by the equation given below where d = 10 days (Mathworks, 2014c):

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-d)) \qquad (9)$$

Equation (9) shows a sliding window operation based upon previous d=10 values to predict share prices on the 10^{th} day.

The algorithm was run over a range of randomly selected data combinations and generated promising regression outcomes particularly over test and validation data, as shown in Figure 4. A regression value closer to 1 means a close regression relationship between outputs and targets whereas a value closer to zero shows a poor correlation and therefore a poorly trained system.

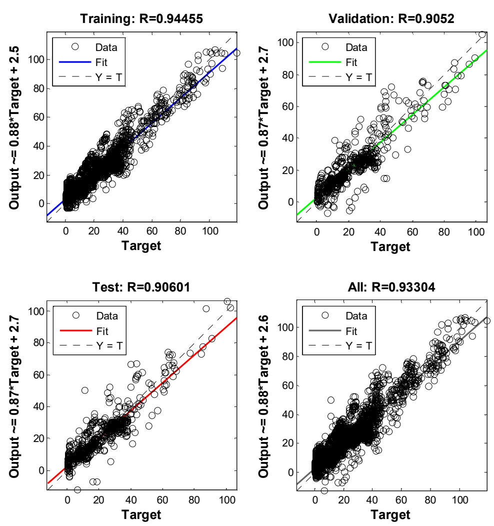

Figure 4. High regression closure values depicting a robustly trained ANN classifier.

The validation performance plotted during the 20-epochs training cycle generated a low Mean-Square-Error (MSE) pattern, which also demonstrates an optimally converged network. Indeed, Figure 5 demonstrates the ability of the underlying training sequence to have improved the overall actual-to-predicted Mean-Square-Error (MSE). The best prediction outcome was shown to be from training data. This is obvious due to the fact that training sequences are already used and known to the system, which is a clear indication of why the overall training error is lower when compared to validation. The highest validation MSE is attributed mainly to the fact that it is obtained when the trained classifier is used against unseen data. On top of it, validation is also used to terminate the training sequence when it sees 5 consecutive MSE increments in continuous epochs. The test performance is still better than the remaining two datasets. This may be attributed to the fact that test sequences generally see a trained classifier and do not tend to see an uncertain classifier which is being trained.

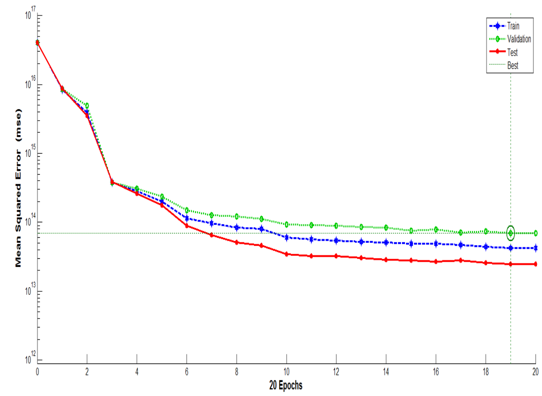

Figure 5. Validation MSE performance during network training.

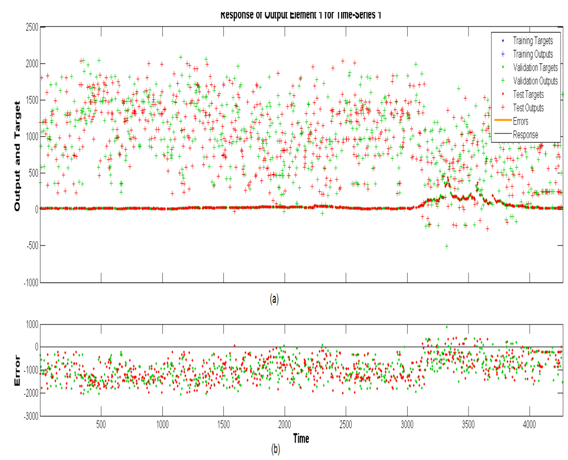

Figure 6. (a) Output and target plot of testing (red markers) and validation data (blue markers) and (b) the respective error plot.

The overall system outcome presents outstanding classification accuracy as evident from Figure 6. The markers for both '.' and '*' represent the target and output comparison for both validation (blue) and test data (red). In Figure 6, the majority of error values can be seen during the 2006 global recession time (see right-most part of Figure 6 (a). Nonetheless, the majority of correct classifications are shown as test values, which demonstrate the viability of this classifier to predict stock data. A sparse spread shows outstanding neural classification accuracy. A sparse error basically indicates a better-trained classifier, which is expected to demonstrate higher prediction accuracy when subjected to unseen data sequences. The overall accuracy of the system was evaluated against two standard testing methodologies of k-fold and jack-knife-based techniques with k = 5. The overall accuracy of these measures is shown in TABLE 1. The k-fold validation randomly divided unseen data into 5 unique sets out of which 4 were used for localised training, testing and validation. Once trained, the trained classifier was then used against a totally unseen (5^{th}) dataset with the prediction outcome recorded. In the next cycle, "group 2" was used as a baseline group against a classifier trained on group 1, 3, 4, 5. The overall accuracy is shown in TABLE I. Nonetheless, the overall system accuracy provides a promising venue for the underlying system to be further improved and extended.

TABLE I. OVERALL ANFIS PREDICTION ACCURACY BASED ON 5-FOLD CROSS-VALIDATION:

Group	5-fold validation (%)
1	92.71
2	85.71
3	92.87
4	89.54
5	88.95
Average	89.956

V. CONCLUSION

This work particularly evaluated the most commonly employed soft-computing paradigms in stock market prediction that include fuzzy logic and neural networks. An in-depth analysis of the current state-of-the-art introduced significant potential in the utilization of hybridised classification systems. The proposed approach utilised the generalization capabilities of neural networks to improve the automated rule-generation capability of Adaptive Network based Fuzzy Inference System (ANFIS) framework. The approach utilised data from Yahoo stock data to train a 10-day-delay back-propagation algorithm that converged with a very promising value greater than 0.8.

The large dataset generated by Yahoo contained a total of 4281 days comprising of an estimated 11 years. In order to evaluate the overall consistency of reporting, the proposed technique employed a data evaluation technique which presented a rounded identification accuracy of 90% with k-fold validation. Despite the promising prediction outcome, the technique could still be improved with a varied number of neurons, activation functions, training algorithm types, number of neurons and the induced training delay. It was envisaged that an improvement in these values can be brought-in via a number of existing optimization techniques. As discussed in the literature review, genetic algorithms, particle swarm optimization, tabu-search and other similar optimization algorithms can be employed to induce an automated, hill-climbing heuristic for the methodology to further improve the system outcome.

REFERENCES

[1] Y. Chen, "Classifying credit ratings for Asian banks using integrating feature selection and the CPDA-based rough sets approach," Knowledge-Based Systems, 26, pp. 259-270, 2012.

[2] L. Dymowa,"Soft computing in economics and finance," Springer, 2011.

[3] E. Blandis, and R. Simutis,"Using Principal Component Analysis and Neural," Network for Forecasting of Stock Market Index. Bizinesa augstskola Turiba SIA, Riga, 2002.

[4] D. Zhu, X. Wang, and R. Ren, "Heuristics R and D projects portfolio selection decision system based on data Mining and fuzzy logic," Intelligent Computation Technology and Automation (ICICTA), 11-12 May, pp. 118-121, 2010.

[5] H. Yih-Jen, "A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques." Fuzzy Systems, IEEE Transactions on, v. 13, n. 2, p. 216-228, 2005. ISSN 1063-6706.2005.

[6] M. A. Lee and M. H. Smith, "Handling uncertainty in finance applications using soft computing", Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society, Proceedings of ISUMA - NAFIPS, Third International Symposium on 17-19 September, pp. 384-389, 1995.

[7] Y. A. Hiemstra, "Stock market forecasting support system based on fuzzy logic," System Sciences, pp. 281-287, 1994.

[8] M. Ben Ghalia and P. Wang, "Intelligent system to support judgmental business forecasting: the case of estimating hotel room demand," Fuzzy Systems, IEEE Transactions on v. 8, n. 4, pp. 380-397, 2000.

[9] D. Qiaolin, T. Jing, and L. Jianxin, "Application of new FCMAC neural network in power system marginal price forecasting," Power Engineering Conference, IPEC, 29 November – 2 December, pp. 1-57, 2005.

[10] G. Ollos and R. Vida, "Adaptive Event Forecasting in Wireless Sensor Networks," Vehicular Technology Conference (VTC Spring), IEEE, 15-18 May, pp. 1-5, 2011.

[11] T. Liu, "Optimizing mining association rules based on Artificial Neural Network," World Automation Congress (WAC), 24-28 June, pp. 1-4, 2012.

[12] N. Balakrishnan, "Methods and applications of statistics in business, finance, and management science," Hoboken, NJ: Wiley, 2010.

[13] R. Kozma, M. Kitamura, M. Sakuma, and Y. Yokoyama, "Anomaly detection by neural network models and statistical time series analysis," Neural Networks, IEEE World Congress on Computational Intelligence 27 June -2 July, pp. 3207-3210, vol.5., 1994.

[14] K. Huang, Z. Qi, and B. Liu, "Network anomaly detection based on statistical approach and time series analysis," Advanced Information Networking and Applications Workshops, Waina 26-29 May 2009, pp. 205-211, 2009.

[15] H. Akaike, "Use of statistical models for time series analysis," Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86, pp. 3147-3155, April 1986.

[16] E. Oja, K. Kiviluoto, and S. Malaroiu, "Independent component analysis for financial time series," Adaptive Systems for Signal Processing, Communications, and Control Symposium, IEEE, pp. 111-116, 2000.

[17] Z. Dazhuo, L. Jinxia, and M. Wenxiu, "Clustering based on LLE for financial multivariate time series," management and service science, 20-22 September, pp. 1-4, 2009.

[18] Q.-Z. Li and W.-C. Shi, "Research in financial risk prediction on biochemical industry of China listed companies," Management Science and Engineering (ICMSE) 20-22 September, pp. 1517-1521, 2012.

[19] F. E. H. Tay and L. J. Cao, "Application of support vector machines in financial time series forecasting," Omega-International Journal of Management Science, v. 29, n. 4, pp. 309-317, 2001.

[20] J. Keyes, "Financial services information systems," Boca Raton: Auerbach, 2000.

[21] L. S. Lopes, N. Lau, P. Mariano, and L. M. Rocha, "Progress in Artificial Intelligence," 14th Portuguese Conference on

Artificial Intelligence, EPIA 2009, Aveiro, Portugal, October 12-15, 2009.
[22] L. Yuling, G. Haixiang, and H. Jinglu, “An SVM-based approach for stock market trend prediction,” Neural Networks (IJCNN), 4-9 August, pp. 1-7, 2013.
[23] P. Chang, C. Liu, and C. Fan, “Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry,” Knowledge-Based Systems, v. 22, n. 5, pp. 344-355, 2009.
[24] C. Li, J. Zhou, P. Kou, and J. Xiao, “A novel chaotic particle swarm optimization based fuzzy clustering algorithm. Neurocomputing”, 83, pp.98-109.2012.
[25] W. L. Tung, C. Quek, and P. Cheng, “GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures,” Neural Networks, v. 17, n. 4, pp. 567-587, 2004.
[26] Y. Yoshida, “The valuation of European options in uncertain environment,” European Journal of Operational Research, v. 145, n. 1, pp. 221-229, 2003.
[27] O. Castillo and P. Melin, ”Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory,” IEEE Transactions on Neural Networks, v. 13, n. 6, pp. 1395-1408, 2002.
[28] T. C. Tang and L. C. Chi, “Predicting multilateral trade credit risks: comparisons of Logit and Fuzzy Logic models using ROC curve analysis,” Expert Systems with Applications, v. 28, n. 3, pp. 547-556, 2005.
[29] F. L Chen, and T. Y Ou, “Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry.” Expert Systems with Applications, 38(3), pp. 1336-1345.2011.
[30] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, “Neural-gas network for vector quantization and its application to time-series prediction,” IEEE Transactions on Neural Networks, v. 4, n. 4, pp. 558-569, 1993.
[31] J. T. Connor, R. D. Martin, and L. E. Atlas, “Recurrent neural networks and robust time-series prediction,” IEEE Transactions on Neural Networks, v. 5, n. 2, pp. 240-254, 1994.
[32] G. P. Zhang and V. L. Berardi, “Time series forecasting with neural network ensembles: an application for exchange rate prediction,” Journal of the Operational Research Society, v. 52, n. 6, pp. 652-664, 2001.
[33] A. S. Soofi and L. Cao, “Modelling and forecasting financial data: techniques of nonlinear dynamics,” Boston, Mass.: Kluwer Academic Publication, 2002.
[34] H. Leung, T. Lo, and S. C. Wang, “Prediction of noisy chaotic time series using an optimal radial basis function neural network,” IEEE Transactions on Neural Networks, v. 12, n. 5, pp. 1163-1172, 2001.
[35] C. K. Goh and K. C. Tan, “Evolutionary multi-objective optimization in uncertain environments: issues and algorithms,” Berlin: Springer-Verlag, 2009
[36] M. Han, J. Xi, S. Xu, and F. Yin, “Prediction of chaotic time series based on the recurrent predictor neural network,” IEEE Transactions on Signal Processing, v. 52, n. 12, pp. 3409-3416, 2004.
[37] Y. Chen, B. Yang, and J. W. Dong, “Time-series prediction using a local linear wavelet neural network,“ Neurocomputing, v. 69, n. 4-6, pp. 449-465, January 2006.
[38] P. R. Kumar and V. Ravi, “Bankruptcy prediction in banks and firms via statistical and intelligent techniques - a review,” European Journal of Operational Research, v. 180, n. 1, pp. 1-28, 1, 2007.
[39] H. Ishibuchi and T. Nakashima, ”A study on generating fuzzy classification rules using histograms,” pp. 132-140 vol.1, 21-23 April 1998.
[40] Y. JaeHung, andI. K Sethi, “Design of radial basis function networks using decision trees.” Neural Networks, pp. 1269-1272 vol.3. Nov/Dec 1995.
[41] H. Shinn-Ying, “Design of accurate regressions with a compact fuzzy-rule base using an evolutionary scatter partition of feature space.” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, v. 34, n. 2, p. 1031-1044. ISSN 1083-4419, 2004.
[42] S. A. Chiu, “Cluster extension method with extension to fuzzy model identification. Fuzzy Systems,”. IEEE World Congress on Computational Intelligence, Proceedings of the Third IEEE, pp. 1240-1245, vol. 2, June 1994.
[43] Yahoo (2014) Yahoo Inc. stock data YAHOO [Online] Available at <https://uk.finance.yahoo.com/q/hp?s=YHOO> [Accessed: 20/06/2014].
[44] S. A Yusuf, D. J Brown, A. Mackinnon, R. Papanicolaou "Application of dynamic neural networks with exogenous input to industrial conditional monitoring," Neural Networks (IJCNN), The 2013 International Joint conference, pp.1, 8, 4-9 Aug. 2013.