D. Bernardo, H. Hagras & E.P.K. Tsang, An Genetic Type-2 fuzzy logic based system for financial applications, modelling and prediction, Proceedings, IEEE International Conference on Fuzzy Systems (FU-IEEE), Hyderabad, India, 7-10 July 2013

A Genetic Type-2 Fuzzy Logic Based System for Financial Applications Modelling and Prediction

Dario Bernardo[*,**], Hani Hagras[*], Edward Tsang[*]
[*]The Computational Intelligence Centre
School of Computer Science and Electronic Engineering
University of Essex, Colchester, UK
[**]Logical Glue Ltd, UK
Email: dbernaa@essex.ac.uk

***Abstract*—Following the global economic crisis, many financial organisations around the World are seeking efficient frameworks for predicting and assessing financial risks. However, in the current economic situation, transparency became an important factor where there is a need to fully understand and analyse a given financial model. In this paper, we will present a Genetic Type-2 Fuzzy Logic System (FLS) for the modelling and prediction of financial applications. The proposed system is capable of generating summarized optimised type-2 FLSs based financial models which are easy to read and analyse by the lay user. The system is able to use the summarized model for prediction within financial applications. We have performed several evaluations in two distinctive financial domains one for the prediction of good/bad customers in a credit card approval application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed Genetic type-2 FLS has outperformed white box financial models like the Evolving Decision Rule (EDR) procedure (which is based on Genetic Programming (GP) and decision trees) and gave a comparable performance to black box models like neural networks while the proposed system provided a white box model which is easy to understand and analyse by the lay user.**

Keywords— type-2 fuzzy logic; financial applications;genetic algorithms; forecasting .

I. Introduction

The global economic meltdown of the late 2000s exposed many organisations around the world where every financial indicator was on a downward trend. As companies begin their slow recovery, they are increasingly looking for ways to reduce the risk associated with their business. Nowadays organisations have access to a quantity of data and information that was not possible to have 20 years ago. Looking at the current trend, in few years the amount of information will be even more. In addition, nowadays everything is online and in seconds it is possible to have huge amounts of information and it is also becoming much easier to store and maintain large amounts of data. Hence, different financial organisations are moving towards generating models based on data where these models are trying to predict the future by looking at the past.

Many organisations around the World still use statistical regression models which capture only information that can be refined into mathematical models to generate two outputs (0/1 or Good/Bad). Statistical regression encompass linear, multiple and logistic regression. Linear regression is widely used in financial applications where it was applied to “capital asset pricing model” [1] as well as for analysing and quantifying the systematic risk of an investment [2] as well as for financial time series modelling [1]. In addition, linear regression is also an important empirical tool in economics where it was employed to predict consumption spending [3], fixed investment spending, inventory investment, purchases of a country's exports [4], spending on imports [4], the demand to hold liquid assets [5], labour demand and supply [6]. Logistic regression is a variant of nonlinear regression that is appropriate when the target (dependent) variable has only two possible values (e.g., buy/don’t-buy, good/bad). As with general nonlinear regression, they are essentially black box models which cannot be easily understood and analysed by the normal user.

Some advanced machine learning and artificial intelligence techniques have been also applied in the financial domain. For example Support Vector Machines (SVMs) have been applied in [7] to forecast financial time series and in [8] to effectively manage governmental funds to small and medium enterprises by identifying those likely to default. Another machine learning technique is Neural Networks (NNs) which have been applied in big number of financial applications such as [9], [10], [11]. However, the drawback of such advanced machine learning techniques is that although they can give good prediction accuracies, they provide black box models which are very difficult to understand and analyse by a financial analyst where it is now becoming a common requirement to have an explanation or the reasoning behind a given financial decision.

There exist various white box transparent models, one of these models is decision trees. Various works have been reported using decision trees in financial applications such as [12], [13] and [14].

Fuzzy Logic Systems (FLSs) provide white box models which could be easily analyzed and understood by the lay user. However FLSs suffer from the curse of dimensionality problem which causes the FLS-based system to generate a big number of rules in order to give good model accuracy. Most recently type-2 FLSs that are capable of handling high uncertainty levels have been employed for the generation of classification models [15], [16]. However, the existing type-2 fuzzy classification systems are not suited for the financial domain where such type-2 FLSs generate big rule bases; besides, they make the assumption that all the possible rules are represented in the existing models which is impossible for systems with big number of inputs where the generated model will only cover a small subset of the search space. This means that, FLSs make the assumption that the obtained rule bases are always able to cover the decision space, which may not be always true.

In this paper, we will present a genetic type-2 FLS for the modelling and prediction of financial applications. The proposed system avoids the drawbacks of the existing type-2 fuzzy classification systems where the proposed system is able to carry prediction based on a relativity small pre-specified rule base size even if the incoming data vector does not match any rules in the FLS rule base. The proposed type-2 FLS aims to increase the understandability of the generated model by achieving the best performance with a limited and summarized number of rules in order to achieve simplicity and comprehensibility for the user. We have carried various evaluations where we are going to present in this paper results from two distinctive financial domains one for the prediction of good/bad customers for a credit card approval application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed system was able to use the generated optimised summarized models for the prediction within financial applications. The proposed system outperformed white box financial models like the Evolving Decision Rule (EDR) procedure (which is a white box model based on Genetic Programming (GP) and decision trees [14]) and gave a comparable performance to black box models like neural networks while the proposed system provided a white box model which is easy to understand and analyse by the lay user. It should be noted that due to the limited space, we will not be able to compare the performance of the type-2 and type-1 fuzzy based systems. The superiority of type-2 based classification systems over their type-1 counterparts have been shown in [15] and [16].

Section II will provide a brief overview on type-2 FLS while Section III will present a brief overview on fuzzy logic based classification systems. Section IV will present the proposed genetic type-2 FLS for financial applications modelling and prediction. Section V will present the experiments and results. The conclusions and future work will be presented in Section VI.

II. A Brief Overview on Interval Type-2 Fuzzy Logic Systems

Interval type-2 FLSs employ type-2 fuzzy sets where an interval type-2 fuzzy set is characterized by a fuzzy membership function, i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1]. The membership functions of interval type-2 fuzzy sets include a Footprint of Uncertainty (FOU) which provide additional degrees of freedom that make it possible to directly model and handle uncertainties. In the interval type-2 fuzzy sets all the third dimension values are equal to one.

The proposed system in the paper is a type-2 fuzzy classification system and hence it does not follow the structure of the type-2 FLSs where the classification system process is summarized in the following section.

An interval type-2 fuzzy set denoted as $\tilde{A}$ is written as follows:

$$\mu_{\tilde{A}}(x) = \int_{x\in X}\int_{u\in\left[\overline{\mu}_{\tilde{A}}(x),\ \underline{\mu}_{\tilde{A}}(x)\right]} 1/u \tag{1}$$

$\overline{\mu}_{\tilde{A}}(x)$, $\underline{\mu}_{\tilde{A}}(x)$ represent the upper and lower membership functions respectively of the interval type-2 fuzzy set $\tilde{A}$.

III. A Brief Overview on Fuzzy Logic Classification Systems

In fuzzy logic classification systems, for a given c-class pattern classification problem with n attributes (or features), a given rule in the FLS rule base could be written as follows:

Rule R^j: If x_1 is A_1^j and ... and x_n is A_n^j then Class C_j with CFj, j = 1,2,...,N (2)

Where $x_1,\ldots,x_n$ represent the n-dimensional pattern vector, A_i^j is the fuzzy set representing the linguistic label for the antecedent pattern i, C_j is a consequent class (which could be one of the possible c classes), N is the number of fuzzy IF-THEN rules in the FLS rule base. CF_j is a certainty grade of rule j (i.e., rule weight). In case each input pattern is represented by K fuzzy sets and given that we have n input patterns, the possible number of rules that will cover the whole search space is K^n. However, in the vast majority of financial applications, we do not have enough data to generate this huge number of rules. Hence, there will be various cases where the incoming input vector will not fire any rule in the FLS rule base. In financial applications, we cannot discharge a given input pattern because there are no matching rules in the rule base. A technique to resolve this problem was proposed in [12], [13] and [14], this technique keeps in a rule repository all the rules for the minority class in unbalanced data sets. All the inputs that do not match any rule in the repository are considered belonging to the majority class. This technique can work in unbalanced data set but might not work in all cases.

IV. The Proposed Genteic Type-2 Fuzzy Logic Based System for Financial Applications Modelling and Predcition

In FLSs, the choice of the appropriate parameters of the fuzzy sets poses a major challenge to the design of a FLS. By simply changing the fuzzy sets parameters, it is possible to change the behaviour of a FLS, for example in the field of managing risk in financial systems it is possible to build riskier or risk-averse fuzzy systems by changing the parameters of the fuzzy sets to make the FLS passing more or less customers. It is extremely difficult though to find the optimal configuration using a simple manual or heuristic approach because of the number of the variables to be optimised and the interaction of these variables. In our work, Genetic Algorithms (GAs) were used to tune the parameters of the type-2 fuzzy sets of the FLS.

The GA uses a population where each chromosome describe a fuzzy set space, in other words the position of each membership function for each input. In simple terms, the GA is positioned before the rule extraction algorithm, and aim to find the best membership function configuration for describing in the best way the problem. The GA basically tries different configurations that through the specified rule extraction algorithm produce the best performances.

In the GA, each instance of the FLS is created by using each individual of the population and each instance generates a different fitness value. As shown in Fig. 1, the steps followed by the proposed genetic type-2 fuzzy system can be summarised as:

1. Initialize randomly the first generation to generate the type-2 fuzzy sets parameters.
2. Build a rule-base for each parameter configuration of the type-2 fuzzy sets as provided by a given chromosome. As the matter of the fact each chromosome describes the fuzzy membership functions configuration and this in conjunction with the training data is used to build the rule-base (the rule base generation process is discussed in Section (IV.B).
3. Evaluate the classification ability of the generated type-2 FLS over a validation data set and produce the fitness value for each individual.
4. If an individual reaches desired fitness value or the max number of iterations are reached the algorithm terminates.
5. The GA uses the population and their fitness values to evolve and produce a new population.
6. Go to step 2.

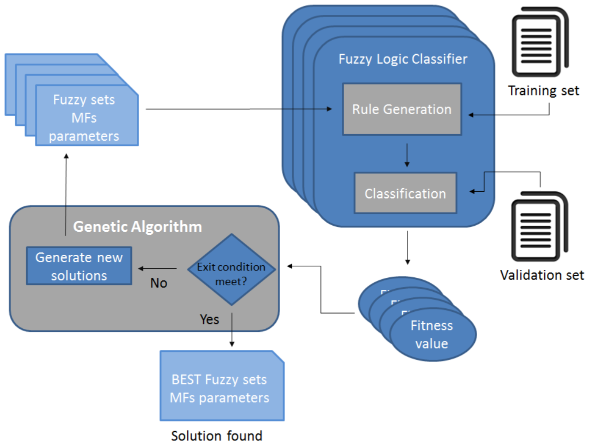

Fig. 1. An overview of the proposed genetic type-2 fuzzy logic system.

A. Employing Genetic Algorithms to determine the type-2 Fuzzy Sets Parameters

The GAs implementation of [17], [18] was used to encode the chromosomes using real numbers.

The GA objective function is to optimize the Recall of the classifier on each class in a weighted way. The Recall is defined as the proportion of the class cases that were correctly identified. We employ a weight w_i to give a weighted importance to a given class i, hence the GA fitness could be written as follows:

$$Fitness\ score = \frac{\sum_i^C Recall_i * w_i}{C} \quad (3)$$

C is the number of classes for the given problem.

By default in a standard configuration all weights are set to 1 in order to give the same importance to all classes.

Each input for the FLS is represented by five type-2 fuzzy sets, which need 17 parameters to be represented as shown in Fig. 2. To fully build the fuzzy sets needed by the system, the total number of parameters (genes) to be optimised can be found as follows:

$$Number\ of\ parameters = 17\ x\ F \quad (4)$$

Where F is the number of inputs (or features). For a dataset with 7 input features the total number of parameter to tune will be 119 parameters thus creating a chromosome composed of 119 genes.

An example of the Membership Functions (MFs) that can be generated by the GA is shown in Fig. 3. As can be seen, the membership functions produced are not equally spaced and they are also asymmetric, thus the use of the GA allows to better model the uncertainity available in the data set.

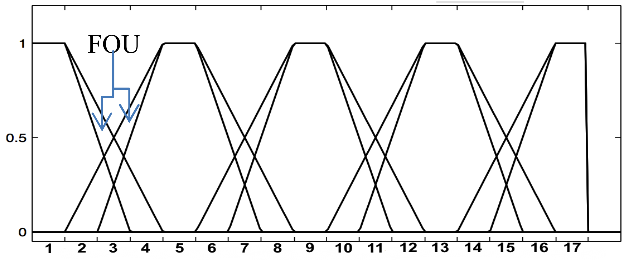

Fig. 2. The number of type-2 fuzzy sets 17 parameters to be tuned for each input.

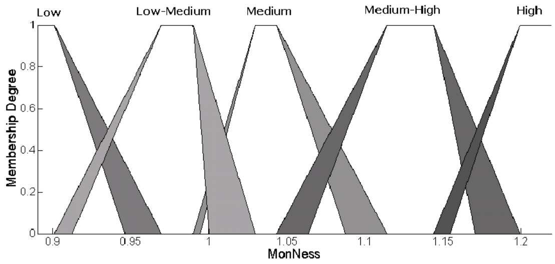

Fig. 3. The MFs learnt by the GA for MonNess input of arbitrage dataset.

Fig. 3 shows the learnt membership functions for the MonNess feature in the arbitrage dataset used in this paper.

This input feature is defined as the division between strike price and the underlying index level. It is interesting to note that the shapes of the produced membership functions are asymmetric and they are actually different from each other in the same dimension. This degree of freedom allows having more fine grained rules in areas of the decision space where is more needed.

B. Rule generation in the proposed Genetic Type-2 FLS

This subsection will show how the rules of the type 2 FLS are modelled taking as an input a dataset and the fuzzy sets whose parameters were optimised by the GA. This is called modelling phase. In the modelling phase the rule base of the type-2 fuzzy classification system is constructed from the existing training dataset. Once the model has been built the FLS can be used to predict new inputs. This is called prediction phase. In the prediction phase, the generated type-2 FLS is used to predict the incoming input vectors. Fig. 4 shows an overview on the modelling and prediction phases.

1) The modelling phase

The modelling phase operates according to the following steps (as shown in Fig. 4):

Step 1- Raw Rule Extraction: For a fixed input-output pair ($x^{(t)}$, $C^{(t)}$) in the dataset, $t=1,...T$ (T is the total number of data training instances available for the modelling phase) compute the upper and lower membership values $\overline{\mu}_{A_s^q}$, $\underline{\mu}_{A_s^q}$ for each antecedent fuzzy set $q=1,...K$ (K is the total number of fuzzy sets representing the input pattern s where $s=1...n.$). Generate all rules combining the matched fuzzy sets A_s^q (i.e. either $\overline{\mu}_{A_s^q} > 0$ or $\underline{\mu}_{A_s^q} > 0$) for all $s=1...n$. Thus the rules generated by ($x^{(t)}$, $C^{(t)}$) will have different antecedents and the same consequent class $C^{(t)}$. Thus each of the extracted raw rules by ($x^{(t)}$, $C^{(t)}$) could be written as follows:

$$R^j: \quad IF\ x_1\ is\ \tilde{A}_1^{qjt}\ and\ ...\ and\ x_n\ is\ \tilde{A}_n^{qjt}$$

$$THEN\ Class\ C_t,\ t = 1,2,...,T \tag{5}$$

For each generated rule, we calculate the firing strength F^t. This firing strength measures the strength of the points $x^{(t)}$ belonging to the fuzzy region covered by the rule. F^t is defined in terms of the lower and upper bounds of the firing strength $\underline{f^{(t)}}, \overline{f^{(t)}}$ of this rule which are calculated as follows:

$$\overline{f^{jt}}\left(x^{(t)}\right) = \overline{\mu_{A_1^{qjt}}}(x_1) * \cdots * \overline{\mu_{A_n^{qjt}}}(x_n) \tag{6}$$

$$\underline{f^{jt}}\left(x^{(t)}\right) = \underline{\mu_{A_1^{qjt}}}(x_1) * \cdots * \underline{\mu_{A_n^{qjt}}}(x_n) \tag{7}$$

The * denotes the minimum or product t-norm. Step 1 is repeated for all the t data points from 1 to T to obtain generated rules in the form of Equation (5). If there are two or more rules generated which have the same antecedents and consequent classes, we will aggregate these rules in one rule having the same antecedents and the same consequent class with the associated $\overline{f^{jt}}$ and $\underline{f^{jt}}$ which result in the maximum average ($\overline{f^{jt}}$ +$\underline{f^{jt}}$)/2 amongst these rules.

The financial data is usually highly imbalanced (for example in a lending application it is expected that the majority of people will be good customers and a minority being bad customers and usually the interesting class is the minority class). Hence, we will present a new approach called "**weighted scaled dominance**" which is an extension of our previous work on "scaled dominance" and the "weighted confidence" work introduced by Ishibuchi in [19]. This method tries to handle imbalanced data by trying to give minority classes a fair chance when competing with the majority class. In order to compute the scaled dominance for a given rule having a consequent Class C_j, we divide the firing strength of this rule by the summation of the firing strengths of all the rules which had C_j as the consequent class. This allows handling the imbalance of data towards a given class. We scale the firing strength by scaling the upper and lower bounds of the firing strengths as follows:

$$\overline{fs^{jt}} = \frac{\overline{f^{jt}}}{\sum_{j \in Class j} \overline{\overline{f^{j}}}} \tag{8}$$

$$\underline{fs^{jt}} = \frac{\underline{fs^{jt}}}{\sum_{j \in Class j} \underline{fs^{jt}}} \tag{9}$$

Step 2- Scaled Support and Scaled Confidence Calculation: Many of the generated rules will share the same antecedents but different consequents. To resolve this conflict, we will calculate the scaled confidence and scaled support which are calculated by grouping the rules that have the same antecedents and conflicting classes. For given m rules having the same antecedents and conflicting classes, the scaled confidence ($\tilde{A}_q \Rightarrow C_q$) (defined by its upper bound $\overline{c}$ and lower bound $\underline{c}$, it is scaled as it involves the scaled firing strengths mentioned in the step above) that class C_q is the consequent class for the antecedents $\tilde{A}_q$ (where there are m conflicting rules with the same antecedents and conflicting consequents) could be written as follows:

$$\overline{c}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in\ Class\ C_q} \overline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \overline{fs^{jt}}(x_s)} \tag{10}$$

$$\underline{c}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in\ Class\ c_q} \underline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \underline{fs^{jt}}(x_s)} \tag{11}$$

The scaled confidence can be viewed as measuring the validity of $A_q \Rightarrow C_q$. The confidence can be viewed as a numerical approximation of the conditional probability [20]. The scaled support (defined by its upper bound $\overline{s}$ and lower bound $\underline{s}$, it is scaled as it involves the scaled firing strengths mentioned in the step above) is written as follows:

$$\overline{s}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in\ Class\ C_q} \overline{fs^{jt}}(x_s)}{m} \tag{12}$$

$$\underline{s}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in\ Class\ c_q} \underline{fs^{jt}}(x_s)}{m} \tag{13}$$

The support can be viewed as measuring the coverage of training patterns by $A_q \Rightarrow C_q$. The scaled dominance, (defined by its upper bound $\overline{d}$ and lower bound $\underline{d}$) can now be calculated by multiplying the scaled support and scaled confidence of the rule as follows:

$$\overline{d}\left(\tilde{A}_q \Rightarrow C_q\right) = \overline{c}\left(\tilde{A}_q \Rightarrow C_q\right) \cdot \overline{s}\left(\tilde{A}_q \Rightarrow C_q\right) \tag{14}$$

$$\underline{d}\left(\tilde{A}_q \Rightarrow C_q\right) = \underline{c}\left(\tilde{A}_q \Rightarrow C_q\right) \cdot \underline{s}\left(\tilde{A}_q \Rightarrow C_q\right) \tag{15}$$

For rules that share the same antecedents and have different consequent classes, we will replace these rules by one rule having the same antecedents and the consequent class which will be corresponding to the rule that gives the highest weighted scaled dominance value. The "weighted scaled dominance" (defined by its upper bound $\overline{wd}$ and lower bound $\underline{wd}$) is calculated using interval mathematics employing the upper and lower bounds as follows:

$$wd\left(\tilde{A}_q \Rightarrow C_q\right) = d\left(\tilde{A}_q \Rightarrow C_q\right) - d_{ave} \tag{16}$$

Where d_{ave} is the average dominance (defined by $\underline{d_{ave}}$, $\overline{d_{ave}}$) over fuzzy rules with the same antecedent $\tilde{A}_q$ but different consequent classes.

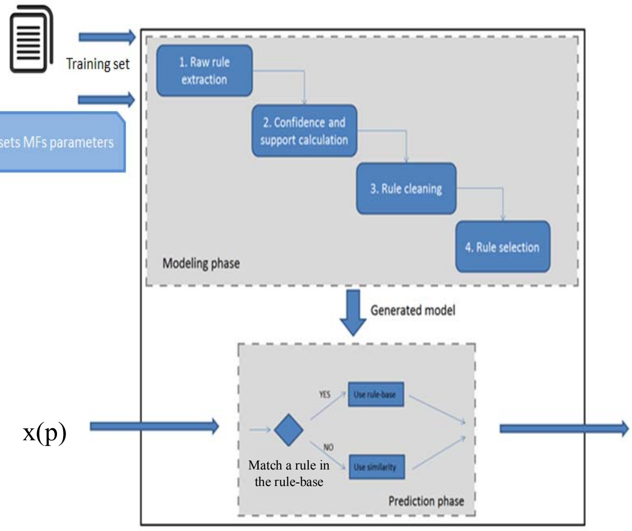

Fig. 4. An overview on the modelling and prediction phases.

Step 3- Rule Selection: As fuzzy based classification methods generate a large number of rules, this could cause major problems for financial applications where the users need to understand the system. Hence, in our method, we will reduce the rule base to a relatively small pre-specified size of rules that generates a summarized model which could be easily read, understood and analysed by the user. In this step, we select only the top Y rules per class (Y is pre-specified by the given financial application) which has the rules with the highest weighted scaled dominance values. This selection is useful because rules with low weighted scaled dominance may not actually be relevant and possibly introduce errors. This helps to keep the classification system more balanced between the majority and minority classes. By the end of this step, the modelling phase is finished where we have $X = nC \cdot Y$ rules (with nC the number of classes) ready to classify and predict incoming patterns as discussed below in the prediction phase.

2) The prediction phase

When an input pattern is introduced to the generated model, two cases will happen: the first case is when the input $x^{(p)}$ matches any of the X rules in the generated model, in this case we will follow the process explained by *case 1*. If $x^{(p)}$ does not match any of the existing X rules, we will follow the process explained by *case 2.*

a) Case 1: The input matches one of the existing rules

In case the incoming input $x^{(p)}$ matches any of the existing X rules, we will calculate the firing strength of the matched rules according to Equations (6) and (7), this will result in $\overline{f^j}\left(x^{(p)}\right)$, $\underline{f^j}\left(x^{(p)}\right)$. In this case, the predicted class will be determined by calculating a vote for each class as follows:

$$\overline{Z}Class_h\left(x^{(p)}\right) = \frac{\sum_{j\in \mathrm{h}} \overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q)}{\max j \in \mathrm{h}\ (\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))} \tag{17}$$

$$\underline{Z}Class_h\left(x^{(p)}\right) = \frac{\sum_{j\in \mathrm{h}} \underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q)}{\max j \in \mathrm{h}\ (\underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q))} \tag{18}$$

In the above equations $\max j \in \mathrm{h}\ (\overline{f^j}(x^{(p)}) * \overline{wd}\left(A_q \to C_q\right))$ and $\max j \in \mathrm{h}\left(\underline{f^j}\left(x^{(p)}\right) * \underline{wd}\left(A_q \to C_q\right)\right)$ represent taking the maximum of the product of the upper and lower firing strengths and the weighted scaled dominance respectively among the "K" rules selected for each class. The total vote strength is then calculated as:

$$ZClass_h = \frac{\overline{Z}Class_h(x^{(p)}) + \underline{Z}Class_h(x^{(p)})}{2} \tag{19}$$

The class with the highest $ZClass_h$ will be the class predicted for the incoming input vector $x^{(p)}$.

b) Case 2: Input does not match any of the existing rules

In case the incoming input vector $x^{(p)}$ does not match any of the existing X rules, we need to decide the output class for the input. The first step is to build all the rules that are possible to be generated from the given input, using the matched fuzzy sets. Let's suppose we have a classification problem with two inputs X_1 and X_2. Let's suppose that a given input $x^{(p)}$ will match overall four different fuzzy sets. Let $MR\left(x^{(p)}\right)$ the set of rules obtained by combining the matched fuzzy sets. So let's suppose there will be four matching fuzzy sets which will generate four different rules: R_1 = {*Small*, *Medium*} R_2 = {*Small*, *Large*} R_3 = {*Medium*, *Medium*} R_4 = {*Medium*, *Large*}. Each of these rules will have an associated firing strength but not an output class. The following step is to find the closest rule in the rule base for each rule in $MR\left(x^{(p)}\right)$. In order to do this, we need to calculate the similarity (or distance) between each of the fuzzy rules generated by $x^{(p)}$ and each of the X rules stored in the rule base. Let's define "k" to be the number of *rules* generated from the input $x^{(p)}$. Let the linguistic labels that fit $x^{(p)}$ be written as $v_{inputr} = (v_{input1r},$

$v_{input2r},\ldots,v_{inputnr})$ where r is the index of the r-th rule generated from the input. Let the linguistic labels corresponding to a given rule in the rule base be $v_j = (v_{j1},v_{j2},\ldots,v_{jn})$. Each of these linguistic labels (*Low, Medium, etc*) could be decoded into an integer. Hence the similarity between the rule generated by $x^{(p)}$ and a given rule in the rule base could be calculated by finding the distance between the two vectors as follows:

$$Similarity_{input\ r\leftrightarrow j} = ((1-\left|\frac{vinput1r-vj1}{V1}\right|)*(1-\left|\frac{vinput2r-vj2}{V2}\right|)*\ldots.*(1-\left|\frac{vinputnr-vjn}{Vn}\right|) \tag{20}$$

Where V_1 .... V_n represent the number of linguistic labels representing each variable. Each rule in the rule-base will have at this point a similarity associated with the *r-th* rule generated form the input. For each rule in $MR(x^{(p)})$ the most similar rule in the rulebase, using Equation (20) will be found to decide on the output class. There will be "*k*" rules (the most similar rules to the *k* rules in $MR(x^{(p)})$) selected to decide for the $x^{(p)}$ input the output class. The predicted class will be determined as a vote for each class as follows:

$$\overline{Z}Class_h\left(x^{(p)}\right) = \frac{\sum_{j\in \mathrm{h}} \overline{wd}(A_q\to C_q)*\overline{f^j}(x^{(p)})}{\max j\in \mathrm{h}\ (\overline{f^j}(x^{(p)})*\overline{wd}(A_q\to C_q))} \tag{21}$$

$$\underline{Z}Class_h\left(x^{(p)}\right) = \frac{\sum_{j\in \mathrm{h}} \underline{wd}(A_q\to C_q)*\underline{f^j}(x^{(p)})}{\max j\in \mathrm{h}\ (\underline{f^j}(x^{(p)})*\underline{wd}(A_q\to C_q))} \tag{22}$$

Where $\underline{f^j}\left(x^{(p)}\right)$ and $\overline{f^j}\left(x^{(p)}\right)$ are the lower and upper firing strength of the most similar rule in the rule base and $\overline{wd}\left(A_q \to C_q\right)$ and $\underline{wd}\left(A_q \to C_q\right)$ are the upper and lower interval of the weighted scaled dominance of the most similar rule of the rule considered in $MR\left(x^{(p)}\right)$. In the above equation $\max j \in h\ (\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))$ and $\max j \in h\left(\underline{f^j}\left(x^{(p)}\right) * \underline{wd}\left(A_q \to C_q\right)\right)$ represent taking the maximum of the product of the upper and lower firing strengths and the weighted scaled dominance respectively among the most similar rules to the "*k*" rules, this measure is used to scale the lower and upper voting strength of each class. The total vote strength is then calculated as:

$$ZClass_{h\ r} = \frac{ZClass_h\left(x^{(p)}\right) + \underline{Z}Class_h\left(x^{(p)}\right)}{2} \tag{23}$$

The class with the highest $ZClass_h$ will be the class associated with $x^{(p)}$.

V. Evaluations and Results

The classification ability of the proposed system has been tested with two different datasets. The first sets of experiments are based on the data which has been used for spotting arbitrage opportunities in the London International Financial Futures Exchange (LIFFE) market [12]. The second sets of experiments were performed on a credit card approval application. In all the data sets below, we divide the data into 3/4 for the training set and 1/4 for the testing set. The 40% of the training set is used during the GA tuning as validation set.

A. Performance on Arbitrage dataset

We have evaluated the proposed approach to predict arbitrage opportunities in the London International Financial Futures Exchange (LIFFE) market. The pre-processed data comprised of 11 input features and 1641 instances of which only 401 representing arbitrage opportunities and the rest representing non-arbitrage opportunities. We have compared the proposed genetic type-2 FLS approach with one of the most powerful white box financial modelling and prediction systems for spotting arbitrage opportunities which is Evolving Decision Rule (EDR) procedure [14]. The EDR method evolves a set of decision rules by using Genetic Programming (GP) and it receives feedback from a key element that is called repository. We have compared as well the proposed approach against Neural Networks which was found to be the best black box model available for this data set.

The proposed genetic type-2 FLS aims to fulfil two objectives: The first one is to get good results on both RECALL and false positive rate, the second objective is to use small number of rules to model and predict the arbitrage opportunities, thus presenting a white box model which could be easily understood and analyzed by the lay user. The perfect ideal classifier is able to have a RECALL (True Positive Rate) of 1 and a False Positive Rate (FPR) of 0, thus the more predictive the given model is, the higher is its Receiver Operating Characteristic (ROC) curve which means that the ROC curve for the given prediction model approaches the ideal classifier. Hence, the more predictive a given model is, the more the area under its curve will approach the ideal classifier whose area under its ROC curve equals to one. Fig. 5 shows the ROC curve obtained by the proposed genetic type-2 FLS plotted against the ROC curves obtained by the EDR procedure [14] and the Neural Networks respectively. From Fig. 5, it is obvious that the proposed genetic type-2 FLS gives a much better ROC curve than the EDR procedure and the Neural Networks while the type-2 FLS presents the user with a small number of rules which summarizes the model and explains the system behaviour to the lay user in an understandable and comprehensible way. Fig. 5 shows the results obtained when employing the proposed genetic type 2 FLS with only 200 and 40 rules. The best results are obtained using 200 rules; the genetic type-2 FLS with just 40 rules has slightly worst performance than the performance with 200 rules but still produces much better results than the EDR procedure and also producing a slightly better performance when compared to Neural Networks. The selected 40 or 200 rules have been selected taking those with highest weighted scaled dominance.

The best average recall overall obtained on this dataset is 96.22% with the genetic type-2 FLS with 200 rules. The genetic FLS with 40 rules best average recall performance was 94.64%. The best average recall obtained by the neural network was 94.04%. The EDR best performances were around 78%.

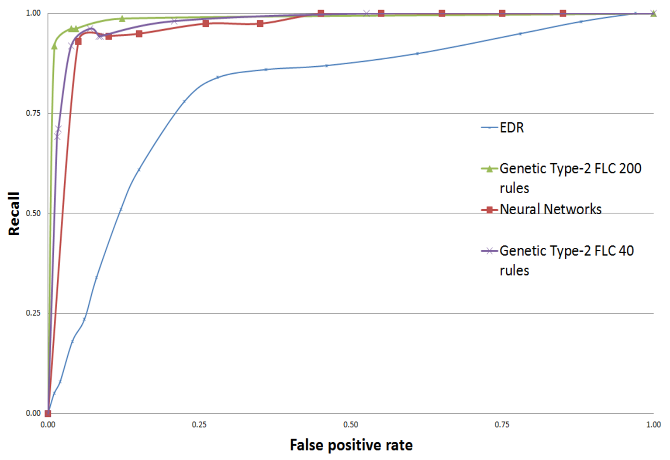

Fig. 5. ROC curve on arbitrage dataset.

These results shows that the proposed genetic type-2 FLS gives a better performance when compared to a white box model like the EDR procedure. The proposed genetic type-2 FLS gave also better performance when compared to black box models which shows that the proposed genetic type-2 FLS can achieve a similar or even better performance than black box models while providing a transparent white box model with a summarised number of rules which are easy to understand and analyse by the lay user.

B. *Performance on Credit Card Approval dataset*

The proposed system has been tested using data gathered from a credit card company. The dataset built includes the information that a customer would provide when asking for credit card. The dataset is composed from 123116 records and 11 features. The system need to be able to find bad credit requests but as well need not to decline too many requests. As the matter of the fact a simple approach to avoid all bad credit requests would be simply of not accepting any requests at all, but of course this extreme scenario is not feasible because this does not match with the business model of credit lenders. The data is extremely unbalanced which consists of 98.54% good customers and 1.46% bad customers. It is worth mentioning that the data is quite different from the Arbitrage data set mentioned in the previous subsection where this data set is very noisy and sparse the hence the prediction accuracy of any prediction system will be rather limited.

The genetic type 2 FLS was compared to neural networks which were found to be the best black box model suited for this data set. Fig. 6 shows the ROC curve of the proposed genetic type-2 FLS plotted against the Neural Networks ROC curve. It was found that the neural networks gave best average recall of 76.25 % while the genetic type-2 FLS gave a close best average recall of 74.17 %. The best performances in this case on this dataset are obtained by the neural network.

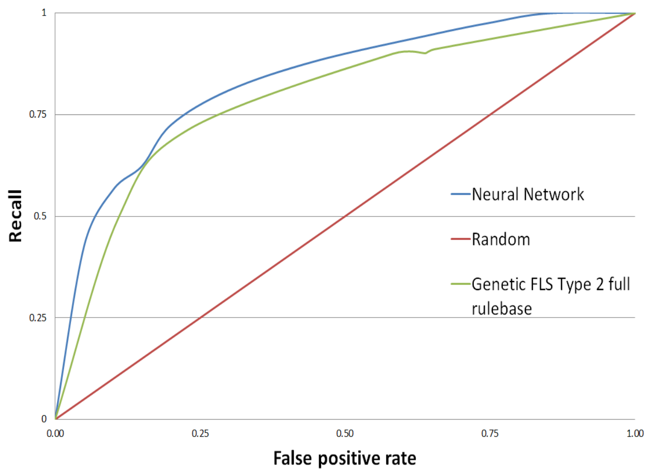

Fig. 6. ROC curve on credit card approval dataset

C. *Scaling technique performances*

In this paper, we have presented a new measure called ***weighted scaled dominance.*** The aim of this new measure is to give more weight to associations that do not occur very often in the dataset, especially those associations affiliated with the minority classes and hence described in the decision space by few samples. In most of the cases the minority classes are usually the relevant class to identify, and scarcity of the samples makes the problem more challenging. There exist in the literature other measures that aim to give more importance to infrequent but still important associations; one of these measures is the *lift*. The *lift* can be defined as the combined support of the consequent and the antecedent of a rule, over the support of the antecedent multiplied by the support of the consequent [21].

Another important measure is the weighted dominance which is obtained simply by the multiplication of scaled confidence and scaled support, explained in [19] and [20]. We compared the results obtained by the different metrics using the same fuzzy sets. Fig. 7 summarise the results obtained with the different data mining measures with varying the size of the rule bases used for classification on the credit card approval dataset.

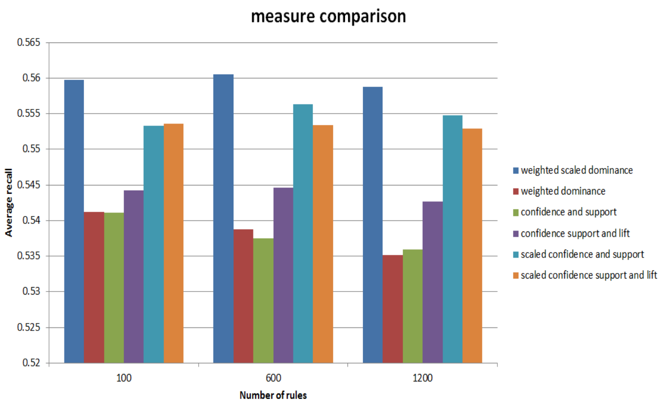

Fig. 7. Result summary on scaling technique.

As shown in Fig. 7, the best result (with best average recall) was obtained employing the suggested ***weighted scaled dominance*** measure as described in Equation (16) or in general whenever the ***scaling*** procedure is used in any technique. The scaling procedure is applied by employing the scaled firing strength described in Equation (8) – (9). When the scaling is not used the next best results are obtained using *lift*. This underlines the point that infrequent association can as well be very important, and this is extremely true in imbalanced datasets. The ***weighted dominance*** without using the scaling procedure did not get good results, as the matter of the fact this measure did not use the scaled firing strength as in the ***weighted scaled dominance***. As can be seen, the comparisons have been conducted with different pre specified rule sizes. The rule selection has been performed by selecting those only with highest metrics in question.

VI. Conclusions and future work

In this paper, we have presented a genetic type-2 FLS capable of generating optimised summarized models from pre-specified number of linguistic rules, which enables the user to understand the generated financial model, thus generating a transparent and easy to read and analyse model.

We have shown how the proposed genetic system allows learning the various parameters of the input type-2 fuzzy sets which cannot be easily designed or manually tuned.

We have presented two novel measures, the first measure is a data-mining measure called weighted scaled dominance where we have shown that it outperformed other widely used measures. The second presented measure was called similarity measure which is a technique used to be able to provide a classification even when the inputs do not match any rules from the rule base.

We have performed several evaluations in two distinctive financial domains one for the prediction of good/bad customers in a financial real-world credit card approval application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed genetic type-2 FLS has outperformed white box models like the Evolving Decision Rule (EDR) procedure (which is a white box financial model based on Genetic Programming (GP) and decision trees) and gave a comparable performance to black box models like neural networks while the proposed genetic type-2 FLS provided a white box model which is easy to understand and analyse by the lay user.

For the future work, we will aim to optimize also the length of the rules and use do not care conditions to make the genetic type-2 FLS easier to read by the lay user. We intend also to perform some evaluations on the similarity technique proposed in this paper validating its performances against a default rule approach.

References

[1] J. Cohen, P. Cohen, S. West and L. Aiken, “Applied multiple regression/correlation analysis for the behavioral sciences.” (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates 2003.

[2] M. Levinson, “Guide to Financial Markets. London: The Economist”. Profile Books, pp. 145–6, 2006.

[3] A. Deaton, "Understanding Consumption", Oxford University Press, 1992.

[4] P. Krugman and M. Obstfeld, "International Economics: Theory and Policy". Glenview: Scott, Foresman, 1988.

[5] D. Laidler, "The Demand for Money: Theories, Evidence, and Problems", 1993.

[6] A. Ehrenberg and Smith, "Modern Labor Economics", HarperCollins, 2008.

[7] K. Kim, “Financial time series forecasting using support vector machines”, Neurocomputing, Vol. 55, No. 2, pp. 307–319, 2003.

[8] H.S. Kim and S.Y. Sohn, “Support vector machines for default prediction of SMEs based on technology credit”, European Journal of Operational Research, Vol 3, pp. 838–846, 2010.

[9] E. Giacomini, "Neural Networks in Quantitative Finance", Master Thesis, Humboldt-Universit¨at zu Berlin, 2003.

[10] R. Lawrence "Using Neural Networks to Forecast Stock Market Prices", University of Manitoba, 1997.

[11] K. Kwong, "Financial Forecasting Using Neural Network or Machine Learning Techniques", University of Queensland, 2001.

[12] A. Garcia-Almanza and E. Tsang, “Forecasting stock prices using genetic programming and chance discovery”, Proceedings of the 12th International Conference On Computing In Economics And Finance, 2006.

[13] A. Garcia-Almanza and E. Tsang, “Evolving decision rules to predict investment opportunities,” The International Journal of Automation and Computers, Vol. 5, No. 1, pp. 22-31, January 2008.

[14] A. Garcia-Almanza, “New classification methods for gathering patterns in the context of genetic programming”, PhD Thesis, Department of Computing and Electronic Systems, University of Essex, 2008.

[15] J. Sanz, A. Fernandez, H. Bustince and F. Herrera, “Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning”. Information Sciences, Vol. 180, pp. 3674-3685, 2010.

[16] J. Sanz, A. Fernandez, H. Bustince and F. Herrera, “A genetic tuning to improve the performance of fuzzy Rule-Based Classification Systems with Interval-Valued fuzzy sets: Degree of ignorance and lateral position”. International Journal of Approximate Reasoning, Vol 52, pp. 751-766, 2011.

[17] S. Shakya, “Markov Random Field Modeling of Genetic Algorithms,” Progress Report submitted to The Robert Gordon University to make the case for transfer from MPhil to PhD, The Robert Gordon University, 2004.

[18] S. Kassem, “ A Type2 Fuzzy Logic System for Workforce Management in the Telecommunications Domain”, MSc. Thesis, University of Essex, 2012.

[19] H. Ishibuchi and T. Yamamoto, “Rule Weight Specification in Fuzzy Rule-Based Classification Systems”, IEEE Transactions On Fuzzy Systems, Vol. 13, No. 4, pp. 428-435, August 2005.

[20] H. Ishibuchi, T. Nakashiman and T. Murata “Three-objective genetic-based machine learning for linguistic rule extraction”, Information Science, Vol. 136, No.4, pp. 109-133, August 2001.

[21] S. Tufféry, “Data Mining and Statistics for Decision Making”, Chichester, 2011.