D. Bernardo, H. Hagras & E.P.K. Tsang, An interval Type-2 fuzzy logic system for the modelling and prediction of financial applications, International Conference on Autonomous and Intelligent Systems (AIS), Aveiro, Portugal, 25-27 June 2012, 95-105 (ISBN 978-3-642-31368-4)

An Interval Type-2 Fuzzy Logic System for the Modeling and Prediction of Financial Applications

Dario Bernardo, Hani Hagras, Edward Tsang

The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK

Abstract. In the recent years, there has been growing interest in developing tools for the modeling and prediction of financial applications. The problem of financial applications is that there are huge data sets available which are sometimes incomplete, and almost always affected by noise and uncertainty. Some techniques used in financial applications employ black box models which do not allow the user to understand the behavior and dynamics of the given application. In this paper, we present a type-2 Fuzzy Logic System (FLS) for the modeling 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 pre-specified rule base size even if the incoming input vector does not match any rules from the given rule base. We have performed several experiments based on the London Stock Exchange data which was successfully used to spot ahead of time arbitrage opportunities. The proposed type-2 FLS has outperformed the existing type-2 fuzzy logic based classification systems and the type-1 FLSs counterparts when using pre-specified rule bases.

Keywords: Type 2 fuzzy logic systems, financial applications.

1. INTRODUCTION

With the current financial applications, there is a pressing need for new comprehensive and accurate approaches to capitalize on economic opportunity without incurring high levels of unexpected risk [1]. The majority of the commercial financial tools employ statistical regression techniques which capture only that information which can be refined into mathematical models to generate two outputs. Moreover, the regression techniques are essentially black box models which cannot be easily understood and analyzed by the normal user. Advanced machine learning and artificial intelligence techniques like Neural Networks suffer from the same problem where they can give good prediction accuracies, however they provide black box models which are very difficult to understand and analyze by a financial manager.

Fuzzy Logic Systems (FLSs) provide white box models which could be easily analyzed and understood by the layman 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 [2], [3]. 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 the huge financial data sets where the generated model will only cover a small subset of the search space. In this paper, we will present a type-2 FLS for the modeling and prediction of financial applications. The proposed system avoids the drawbacks of the existing type-2 fuzzy classification systems because it is able to carry prediction based on a pre-specified rule base size even if the incoming data vector does not match any rules in the FLS rule base. We have performed several experiments based on the London Stock Exchange data which was successfully used to spot ahead of time arbitrage opportunities. The proposed type-2 FLS has outperformed the existing type-2 fuzzy logic based classification systems and also the type-1 FLSs counterparts when using pre-specified rule bases.

In Section 2, we will present a brief overview on type-2 FLSs. Section 3 will present an overview on the fuzzy classification systems. Section 4 will present the proposed type-2 fuzzy based modeling and prediction system for financial applications. Section 5 will presents the experiments on the arbitrage data and the achieved results. Finally Section 6 will present the conclusions and future work.

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

In the recent years type-2 FLSs have grown in popularity due to their ability to handle high levels of uncertainties. Type-2 FLSs employ type-2 fuzzy sets as shown in Fig. 1 where a 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] [4]. The membership functions of type-2 fuzzy sets are three dimensional and include a footprint of uncertainty (shaded in grey in Fig. 1, it is the new third-dimension of type-2 fuzzy sets and the footprint of uncertainty that provide additional degrees of freedom that make it possible to directly model and handle uncertainties [4], [5]. The interval type-2 FLSs use interval type-2 fuzzy sets (such as the type-2 fuzzy set shown in Fig. 1 to represent the inputs and/or outputs of the FLS). In the interval type-2 fuzzy sets all the third dimension values are equal to one. The use of interval type-2 FLS helps to simplify the computation (as opposed to the general type-2 FLS).

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 reported in [4], [5] where the classification system process is summarized in the following section.

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

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

$\overline{\mu}_{\tilde{A}}(x)$, $\underline{\mu}_{\tilde{A}}(x)$ represent the upper and lower membership functions respectively of the interval type-2 fuzzy set $\tilde{A}$. The upper membership function is associated with the upper bound of the footprint of uncertainty $FOU(\tilde{A})$ of a type-2 membership function. The lower membership function is associated with the lower bound of $FOU(\tilde{A})$ [4].

In our system, the generation process of the employed interval type-2 fuzzy sets starts by generating type-1 fuzzy sets which equally partition the input universe of discourse into a given number of partitions. We then blur the type-1 fuzzy sets to the left and the right equally by a given uncertainty factor as shown in Fig. 1a to generate the type-2 fuzzy sets. In the application shown in this paper, we have employed 4 fuzzy sets as shown in Fig. 1b to represent each input variable.

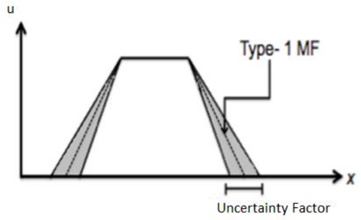
3. A Brief Overview on Fuzzy Logic Classification Systems

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:

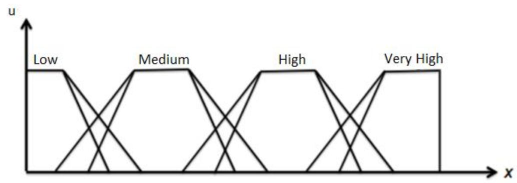
Rule R^j: If x_1 is A_1^j and … and x_n is A_n^j then Class C_j with CFj, j = 1,2,…,N (2)

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. In the arbitrage application presented in this paper, we have 7 inputs where each input is represented by 4 fuzzy sets, hence the needed number of rules to cover the whole search space for this given application is 4^7= 16384 rules. In our given application (which applies to 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.

Several type-1 fuzzy classification systems have been reported in the literature such as [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] and [16]. However, in the vast majority of these papers, the data was relatively pretty easy to partition, and if an input pattern does not match any of the decision areas previously labeled, the input is discharged. In financial applications this cannot be done where if a new pattern that have never been seen before is proposed, a decision need to be made anyway, and unfortunately discharging it a priory cannot be the solution. A technique to resolve this problem was proposed in [17], [18], 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 will might not work in all cases.



(a)



(b)

Fig. 1 - a) The process followed to generate a type-2 fuzzy set from a type-1 fuzzy set. b) The employed interval type-2 fuzzy sets in the application reported in this paper.

4. The Proposed Type-2 Fuzzy Modeling and Prediction System for Financial Applications

The proposed system has two phases, a modeling phase and a prediction phase. In the modeling phase the rule base of the type-2 fuzzy classification system is constructed from the existing training dataset. In the prediction phase, the generated rule base is used to predict the incoming input vectors.

4.1 The Modeling Phase

The modeling phase operates according to the following steps (as shown in Fig. 2):

Step 1- Raw Rule Extraction: For a fixed input-output pair ($x^{(t)}$, $C^{(t)}$) in the dataset ,$t=1,\ldots T$ (T is the total number of data training instances available for the modeling phase) compute the upper and lower membership values $\overline{\mu}_{A_s^q}$, $\underline{\mu}_{A_s^q}$ for each antecedent fuzzy set $q=1,\ldots K$ (K is the total number of fuzzy sets representing the input pattern s where $s=1\ldots n.$). 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 $s=1\ldots n$. 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:

$$R^j: \quad \text{If } x_1 \text{ is } \tilde{A}_1^{qjt} \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{A}_n^{qjt} \text{ then Class } C_t,\ t = 1,2,\ldots,T \tag{3}$$

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:

$$\overline{f^{jt}}\left(x^{(t)}\right) = \overline{\mu_{A_1^{qjt}}}(x_1) * \cdots * \overline{\mu_{A_n^{qjt}}}(x_n) \tag{4}$$

$$\underline{f^{jt}}\left(x^{(t)}\right) = \underline{\mu_{A_1^{qjt}}}(x_1) * \cdots * \underline{\mu_{A_n^{qjt}}}(x_n) \tag{5}$$

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 (3). 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 loan approval applications, we find that the vast majority of the customers are good customers and a minority are bad customers who might never pay their loan back). Hence, we will present a new approach called ***"scaled dominance"*** which tries to handle imbalanced data by trying to increase the confidence and support for the minority 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:

$$\overline{fs^{jt}} = \frac{\overline{f^{jt}}}{\sum_{j\in Class j}\overline{f^{j}}} \quad (6)$$

$$\underline{fs^{jt}} = \frac{\underline{fs^{jt}}}{\sum_{j\in Class j}\underline{fs^{jt}}} \quad (7)$$

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 [2]:

$$\overline{c}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in Class\ C_q} \overline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \overline{fs^{jt}}(x_s)} \quad (8)$$

$$\underline{c}\left(\tilde{A}_q \Rightarrow C_q\right) = \frac{\sum_{x_s \in Class\ c_q} \underline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \underline{fs^{jt}}(x_s)} \quad (9)$$

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 [8]. 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) of $A_q \Rightarrow C_q$ is written as follows:

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

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

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

$$\overline{d}\left(\tilde{A}_q \Rightarrow C_q\right) = \overline{c}\left(\tilde{A}_q \Rightarrow C_q\right) \cdot \overline{s}\left(\tilde{A}_q \Rightarrow C_q\right) \quad (12)$$

$$\underline{d}\left(\tilde{A}_q \Rightarrow C_q\right) = \underline{c}\left(\tilde{A}_q \Rightarrow C_q\right) \cdot \underline{s}\left(\tilde{A}_q \Rightarrow C_q\right) \quad (13)$$

Step 3- Rule Cleaning: 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, will be corresponding to the rule that gives the highest scaled dominance value. In [19], the rule generation system generates only the rule with the highest firing strength, however in our method, we generate all rules that are generated by the given input patterns, and this allows covering a bigger area in the decision space.

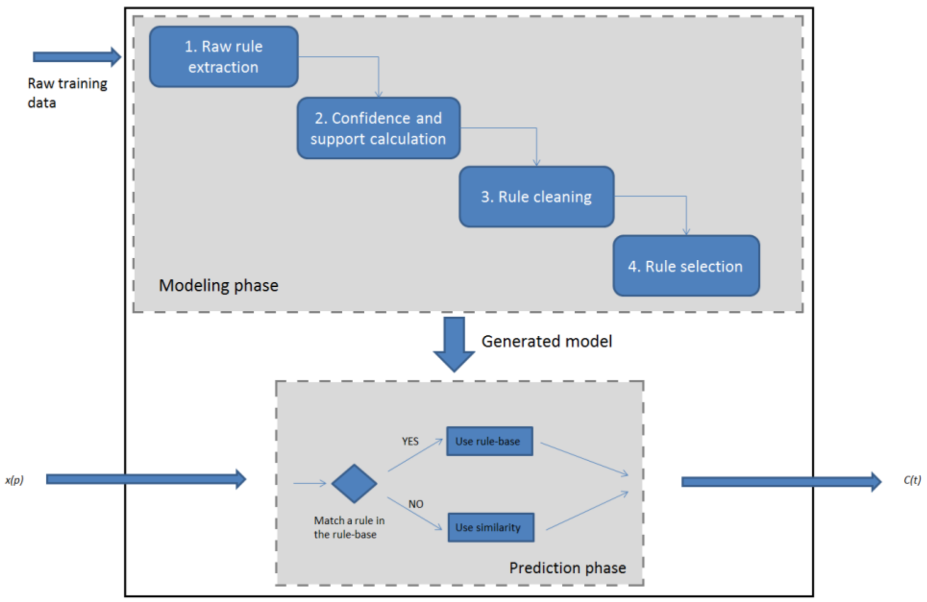


Fig. 2 - An overview of the proposed modeling and prediction system.

Step 4- Rule Selection: As fuzzy based classification methods generate big number of rules, hence this can cause major problems for financial applications where it is needed to understand the market or user behavior and dynamics. Hence, in our method, we will reduce the rule base to a pre-specified size of rules that could be easily read, understood and analyzed 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 scaled dominance values. This selection is useful because rules with low dominance may actually be not relevant and actually introduce errors. This helps to keep the classification system more balanced between the majority and minority classes. By the end of this step, the modeling 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.

4.2 Prediction Phase

When an input pattern is introduced to the generated model, two cases will happen, the first case will happen when the input $x^{(p)}$ matches any of the X rules in the generated model, in this case we will follow the process explained in subsection (4.2.1). If $x^{(p)}$ does not match any of the existing X rules, we will follow the process explained in subsection (4.2.2).

4.2.1 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 (4) and (5), this will result in $\overline{f^j}(x^{(p)}), \underline{f^j}(x^{(p)})$. In this case, the predicted class will be determined by calculating a vote for each class as follows:

$$\overline{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in \mathrm{h}} \overline{f^j}(x^{(p)}) * \overline{d}(A_q \to C_q)}{\max j \in \mathrm{h}(\overline{f^j}(x^{(p)}) * \overline{d}(A_q \to C_q))} \quad (14)$$

$$\underline{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in \mathrm{h}} \underline{f^j}(x^{(p)}) * \underline{d}(A_q \to C_q)}{\max j \in \mathrm{h}(\underline{f^j}(x^{(p)}) * \underline{d}(A_q \to C_q))} \quad (15)$$

The total vote strength is then calculated as:

$$ZClass_h = \frac{\overline{Z}Class_h(x^{(p)}) + \underline{Z}Class_h(x^{(p)})}{2} \quad (16)$$

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

4.2.1 Case 2: The input does 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 find the closest rule in the rule base that matches $x^{(p)}$. In order to do this, we need to calculate the similarity (or distance) between each of the fuzzy rule generated by $x^{(p)}$ and each of the X rules stored in the rule base. The rules generated by $x^{(p)}$ are found by taking each element in $x^{(p)}$ and taking all matching fuzzy sets with either $\overline{\mu_{A_1^{qj}}}(x_i)$ or $\underline{\mu_{A_1^{qj}}}(x_i)$ greater than 0. At this point there will be k 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:

$$Similarity_{input\ r \leftrightarrow j} = ((1-\left|\frac{vinput1r - vj1}{V1}\right|) * (1-\left|\frac{vinput2r - vj2}{V2}\right|) * \ldots * (1-\left|\frac{vinputnr - vjn}{Vn}\right|) \quad (17)$$

In the equation V_s represents the number of linguistic labels representing each variable *s*. Each rule in the rule-base will have at this point a similarity associated with the *r-th* rule generated form the input. In this case, the predicted class will be determined by firstly selecting the rules with the highest similarity with the *r-th* generated rule. There will be more than one rule with the same similarity. Considering the rules that will have the same similarity with the *r-th* rule, the winning class for the *r-th* generated rule is calculated as a vote for each class as follows:

$$\overline{Z}Class_{h\,r}(x^{(p)}) = \sum_{j \in \mathrm{h}} \overline{d}(A_q \to C_q) \quad (18)$$

$$\underline{Z}Class_{h\,r}(x^{(p)}) = \sum_{j \in \mathrm{h}} \underline{d}(A_q \to C_q) \quad (19)$$

The total vote strength is then calculated as:

$$ZClass_{h\,r} = \frac{\overline{Z}Class_{h\,r}(x^{(p)}) + \underline{Z}Class_{h\,r}(x^{(p)})}{2} \quad (20)$$

The class with the highest $ZClass_{h\,r}$ will be the class associated with the *r-th* rule generated from the input. From all the *k* rules generated from $x^{(p)}$, the final output class is calculated by applying Equations (14), (15) and (16).

5. Experiments and Results

We have tested the proposed type-2 fuzzy logic system mentioned in this paper to model and predict arbitrage opportunities. Computers today are able to spot stock misalignment in the market and in milliseconds. This would allow them to make almost risk-free profits. There are two main challenges in this type of operation. Firstly, arbitrage situation do not occur very often. Secondly, the operator must act ahead of others, so the competition is reduced to how fast a computer is, and how fast its connection to the stock exchange is. . Tsang et al [17] showed that arbitrage opportunities do not appear instantaneously. There are patterns in the market which can be recognized 10 minutes ahead.

The proposed system is trained to identify ahead of time arbitrage opportunities, as it is done in [17]. The data reported in this paper was further developed in [18], [21], in order to identify arbitrage situations by analyzing option and futures prices in the London International Financial Futures Exchange (LIFFE) market. The pre-processed data comprised 1641 instances of which only 401 representing arbitrage opportunities and the rest representing non-arbitrage opportunities. The data was split into 2/3 for modeling and 1/3 for testing.

The type-2 fuzzy modeling and prediction system employed a 20 % uncertainty factor to generate the type-2 fuzzy sets as shown in Fig. 1a. The system was trained with the training data to generate a model of 100 rules per class.

For performance evaluation, the achieved results are compared based on average RECALL on minority and majority class. In pattern recognition and information retrieval, RECALL is calculated as the fraction between the numbers of correctly recognized inputs belonging to the minority (majority) class over the total number of inputs actually belonging to minority (majority) class. In our case, high recall means that an algorithm returned most of the arbitrage opportunities [20]. Recall is important in this application because arbitrage opportunities are rare, hence we cannot afford to lose too many of them.

Table 1. The achieved results.

	RECALL-minority	RECALL-majority	AVG-RECALL
HFS method	99.79%	63.52%	**81.66%**
AMRG method / scaled dominance	94.97%	84.03%	**89.50%**
HFS method / scaled dominance on Type1	81.76%	91.81%	**86.78%**

AVG-RECALL is the average recall of the $RECALL_{minority}$ and $RECALL_{majority}$. Table 1 shows the results obtained on the testing data when the proposed type-2 fuzzy based system employ the approach which generate in the modeling phase only the rule with Highest Firing Strength (HFS method) where we can see how our approach which involves generating at the modeling phase All the Matched Rule Generated (AMRG method) and scaled dominance have a superior performance which is better by *7.84%* than the HFS method. Table 1 also shows when using our proposed approach how the type-2 fuzzy logic based system is able to produce *2.72 %* better average RECALL than the type-1 based system using the HFS method and scaled dominance. In addition, it is obvious that the proposed system was able to provide prediction for all the incoming input vectors thus avoiding the drawbacks of the fuzzy classification systems counterpart.

6. Conclusions and Future Work

In this paper, we have presented a type-2 FLS for the modeling and prediction of financial applications. The proposed system included new techniques such as scaled dominance and similarity matching which enable us to produce predictions when the input vector does not match the rules in the rule base.

We have performed several experiments based on the arbitrage data which was used successfully to spot ahead of time arbitrage opportunities [17]. We have shown that the proposed type-2 FLS outperformed the existing type-2 fuzzy logic based classification systems and the type-1 FLSs counterparts when using pre-specified rule bases.

For our future work, we aim to employ genetic algorithms to tune the type-2 fuzzy sets in order to get better results. We also intend to build a more flexible tool to move along the Receiver Operating Characteristic (ROC) curve with more detail [18].

REFERENCES

1. BBC: Meltdown Losses of $4 trillion. BBC News, news.bbc.co.uk, 22 April 2009
2. Sanz, J., Fernandez, A., Bustince, H., Herrera, F.: Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. Information Sciences, 180, 3674-3685, (2010).

3. Sanz, J., Fernandez, A., Bustince, H., Herrera, F.: 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. 52, 751-766, (2011)
4. Hagras, H.: A hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. IEEE Transactions on Fuzzy Systems, 12, 4, 524-539, (2004).
5. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, (2001).
6. Ishibuchi, H.: Comparison of Heuristic criteria for fuzzy rule selection in classification problems. Fuzzy Optimization and Decision making, 3, 2, 119-139, (2004)
7. Ishibuchi, H.: Effect of rule weights in fuzzy rule-based classification system. IEEE Transactions on Fuzzy Systems, 1, 59-64 , (2000)
8. Ishibuchi, H.: Rule Weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems. 13, 4, 428-435, (2005)
9. Ishibuchi, H.: Three-objective genetic-based machine learning for linguistic rule extraction. Information Sciences. 136, 1-4, 109-133, (2001)
10. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multy-objective genetic local search algorithms and rule evaluation measures in data mining. Information Sciences. 141, 1, 59-88, (2004).
11. Ishibuchi, H., Yamamoto, T., Nakashima, T.: An approach to Fuzzy Default Reasoning for function approximation. Soft Computing. 10, 9, 850-864, (2006).
12. Shigeo, A., Lan, M.: A method for fuzzy rules extraction directly from numerical data and its application in pattern classification. IEEE transactions on fuzzy systems, 3, 1, 18-28, (1995).
13. Ahmad, S., Jahormi M.: Construction accurate fuzzy classification systems: A new approach using weighted fuzzy rules. Computer Graphics, Imaging and Visualisation, 408-413, (2007).
14. Wang, L.: The WM Method Completed: A flexible fuzzy system approach to data mining. IEEE Transactions on fuzzy systems, 11, 6, 768-782, (2003).
15. Mansoori, E., ZolGhadri, M., Katebi, D.: Using distribution of data to enhance performance of fuzzy classification system. Iranian Journal of Fuzzy Systems, 4, 1, (2006).
16. Fernandez, A., Jose del Jesus, M., Herrera, F.: Improving the performance of fuzzy rule based classification systems for highly imbalanced data-sets using an evolutionary adaptive inference system. Bio-Inspired Systems: Computational and Ambient Intelligence. 5517, 294-301, (2009)
17. Tsang, E., Markose, S., Er, H.: Chance discovery in stock index option and future arbitrage, New Mathematics and Natural Computation. World Scientific, 1, 3, 435-447, (2005).
18. Garcia-Almanza, A., "New classification methods for gathering patterns in the context of genetic programming", PhD Thesis, Department of Computing and Electronic Systems, University of Essex, (2008).
19. Chi, Z., Yan, H., Pham, T.: Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. World Scientific Pub Co Inc, ISBN / ASIN: 9810226977, (1996).
20. Olson, D., Delen, D.: Advanced Data Mining Techniques. Springer, 1st edition, 138-139, ISBN 3540769161, (2008).
21. Garcia-Almanza, A.L. & E.P.K. Tsang, Evolutionary Applications for Financial Prediction: Classification Methods to Gather Patterns Using Genetic Programming, VDM Verlag, 2011