Received April 13, 2021, accepted May 2, 2021, date of publication May 6, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077910

Using Fuzzy Inference Systems for the Creation of Forex Market Predictive Models

AMAURY HERNANDEZ-AGUILA[1], MARIO GARCÍA-VALDEZ[1], JUAN-JULIÁN MERELO-GUERVÓS[2], MANUEL CASTAÑÓN-PUGA[3], AND OSCAR CASTILLO LÓPEZ[1], (Senior Member, IEEE)

[1]Department of Graduate Studies, National Technological Institute of Mexico, Tijuana 22414, Mexico
[2]Department of Computer Architecture and Technology, University of Granada, 18071 Granada, Spain
[3]Chemistry and Engineering School, Autonomous University of Baja California, Tijuana 22390, Mexico

Corresponding author: Mario García-Valdez (mario@tectijuana.edu.mx)

This work was supported in part by the Project DeepBio under Grant TIN2017-85727-C4-2-P.

ABSTRACT This paper presents a method for creating Forex market predictive models using multi-agent and fuzzy systems, which have the objective of simulating the interactions that provoke changes in the price. Agents in the system represent traders performing buy and sell orders in a market, and fuzzy systems are used to model the rules followed by traders performing trades in a live market and intuitionistic fuzzy logic to model their decisions' indeterminacy. We use functions to restrict the agents' decisions, which make the agents become specialized at particular market conditions. These "specialization" functions use the grades of membership obtained from an agent's fuzzy system and thresholds obtained from training data sets, to determine if that agent is specialized enough to handle a market's current conditions. We have performed experiments and compared against the state of the art. Results demonstrate that our method obtains predictive errors (using mean absolute error) that are in the same order of magnitude than those errors obtained by models generated using deep learning and models generated by random forest, AdaBoost, XGBoost, and support-vector machines. Furthermore, we performed experiments that show that identifying specialized agents yields better results.

INDEX TERMS Economic forecasting, fuzzy systems, multi-agent system, forex market.

I. INTRODUCTION AND STATE OF THE ART

The prices of a financial market can be forecasted using different techniques, such as the analysis of raw price data or news involving the financial market of interest [1]. These approaches have the disadvantage of generating predictions entirely dependent on the trader's skills and knowledge about the market being predicted. A more robust approach is to preprocess the raw price data using technical indicators—series of data points, obtained by applying a formula to price data [2]—to distill different aspects of it [3], such as a market's volatility or general direction. In the case of using news to draw predictions for a market, a more robust approach would be the use of sentiment analysis [4], [5], which can draw conclusions about the general sentiment of a financial market in order to know if prices will go down or up. An inherent disadvantage of these methods is that we are not obtaining explanations about why certain behaviors occurred; we are not generating a model that could help us do an abstract simulation of a market. Predictive models can be created using a variety of techniques, such as ARIMA [6] and hidden Markov models [7], or machine learning techniques, such as support-vector machines (SVM) [8] and neural networks [9]. As we mentioned before, a drawback of some of these methods is that they generate *black box* models, i.e. it is difficult for a user to understand how the model is processing its inputs. A technique that alleviates this problem is fuzzy logic, which can also be used to create predictive models [10]–[12].

The method presented in this paper uses a hybridization of fuzzy logic and multi-agent systems (MAS). The use of fuzzy logic enables the method to generate easily interpretable models to the user, especially if we use Mamdani fuzzy sets [13] to create the membership functions of a fuzzy system, as in the work of Abdulgader and Kaur [14]. In particular, we propose the use of intuitionistic fuzzy logic (IFL) [15], [16], as IFL adds another layer of interpretability for the fuzzy systems through the concept of indeterminacy.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

Although there are alternatives to IFL, such as type-2 fuzzy systems, the defuzzification of IFL systems can be faster than the aforementioned systems [17], while also providing comparable interpretability [18], [19].

Arguably, the most prominent use of fuzzy systems has occurred in the field of control of nonlinear systems [20], [21], but fuzzy systems have proved to be useful in many other fields. In the case of financial market prediction, fuzzy systems have been used successfully, such as in the works by Tsai *et al.* [22] and Zeng *et al.* [23], where fuzzy time-series are used to perform price predictions; as in the works by Rajab and Sharma [24] and Vlasenko *et al.* [25], where a neuro-fuzzy approach is taken; or as in the works by Yue *et al.* [26], Witayakiattilerd [27] and Mansour *et al.* [28], where fuzzy logic is used to perform a portfolio selection of financial markets.

MAS provide mechanisms and an architecture that enable agents to behave comparably to how traders behave in the real world: traders have beliefs that make them interpret market data in different ways, as well as rule systems that define their decision processes. Furthermore, MAS coordinate the inputs given to each agent, as well as how we can use the agents' outputs to generate a simulation of market prices. Furthermore, MAS architectures can be executed in a distributed manner [29], [30], enabling faster performance due to the rules of different agents being evaluated in parallel. The current implementation of our method does not follow such architecture, but adapting our method to a distributed evaluation of the agents' rules is considered as a future work, as is discussed in Section VII.

The present work uses fuzzy systems in combination with MAS to create Forex market predictive models. A predictive model is a relationship between inputs and outputs based on a data model, which can be used to perform regression analysis [31]. In the case of Forex market predictive models, the inputs can represent financial indicators, such as market prices, financial statements' data, or the public's sentiment towards the asset, while the outputs usually represent data that help a trader take a decision regarding the direction—buy or sell—and the quantity of an asset to trade [32].

The agents in the MAS represent traders, and the trades performed by these traders are used to simulate the real prices of a market. MAS have been demonstrated to be an effective approach to simulate very complex systems, such as those represented by financial markets [33]–[37]. Additionally, the agents in the MAS can be examined to understand how individual traders are interacting in a financial market [24], [38], which is a feature that we leverage in our method, by using fuzzy systems to construct the agents' rules, resulting in interpretable inference systems.

We highlight the main contributions as follows:

- First, our method describes a novel architecture that employs MAS and a Mamdani intuitionistic fuzzy inference systems for modeling agents' rules, and beliefs. This approach enables the creation of MAS predictive models, that include certain elements not found in the literature of financial market prediction as indeterminacy, and doubt.
- Second, we propose a process for agent specialization that are part of each agent's beliefs, in the case of financial market forecasting, this models the decision of agent between making a trade or not. We found, that adding this concepts helps to decrease the error between predicted and real market prices.
- Lastly, we prove through the results of experiments, that the models generated by our method are competitive against those generated by state of the art methods: deep learning, random forest, AdaBoost, XGBoost and SVM.

The reader will find an in-depth explanation of our method in Section III, and explanations for the concepts required to understand the method can be found in Section II. In order to evaluate the performance of our method, experiments were performed and are described in Section IV. The results of the experiments are presented in Section V, and a discussion of these results can be found in Section VI. Finally, Section VII discusses some directions that our presented method can take in the future to better demonstrate its capabilities.

II. PRELIMINARIES

This Section describes the concepts that the reader needs to be familiar with in order to better understand the proposed method in Section III.

A. FUZZY SETS

A traditional set is a collection of items that share a common characteristic. This characteristic serves as a membership, because all the items in a universe either have that characteristic—and then the item is part of the set—or it does not have it—and then the item is not part of the set. Traditional sets can be extended to fuzzy sets, as explained by Zadeh [39]. Fuzzy sets are then a generalization of traditional sets, i.e. any traditional set can be represented as a fuzzy set. The difference between these two type of sets lies in the concept of membership: memberships are not only used to represent binary outcomes, i.e. *true* or *false*, but now a possibly infinite number of outcomes. An item can now be partially a member of a set, and the only way an item is not part of such set is if its membership is totally *false*. In order to represent this grade of membership one can use real numbers. Thus, one can say, for example, that an item is *0.7 green*, *0.5 blue* and *0.0 red*. These values can represent an adverb and an adjective, such as "very green," "somewhat blue" and "not red at all." This is especially useful when designing fuzzy systems (see Subsection II-B).

B. FUZZY SYSTEMS

In traditional logic one can generate logical inferences, such as *if it's raining, then there are clouds in the sky*. In a similar fashion, we can use fuzzy sets to represent the antecedents and consequents in a logical inference process [40].

For example, one can extend the previous example to: *if it's raining a lot, then there are many clouds in the sky*.

There is a number of ways in which one can construct a fuzzy inference system, where one or more inputs or antecedents can be used to generate one or more outputs or consequents. Arguably, the two most popular types of fuzzy inference systems are the ones proposed by Mamdani and Assilian [13], and Takagi and Sugeno [41]. These systems use a series of fuzzy sets to represent the relationship between an input and its grade of membership to a set. These sets usually represent adjectives that describe the inputs, and are also considered to be the antecedents in the fuzzy inference system. For example, an input of 0.8 can represent a "very high" value. After obtaining these grades of membership, one can use these values to "fire" or "activate" the consequents. In the case of a Mamdani system, the consequents are represented as fuzzy sets, just like the antecedents. In contrast, in a Sugeno system, consequents are represented by mathematical functions. A set of rules is used to determine the relationship between the antecedents and the consequents, for example: *if food quality is high then tip is high*. The aforementioned rule is creating a relationship between the fuzzy set that represents "high food quality" in the antecedents, and the fuzzy set that represents "high tip" in the consequents. Further continuing with the example, if "food quality" is represented by a value of 0.8, the rule that creates the relationship between "food quality" and "tip" could determine a "tip" of 0.8 too, depending on what membership function and what parameters are decided to be used to represent each.

We have explained how a relationship between antecedents and consequents can be constructed in a fuzzy inference system. Nevertheless, the most interesting problem arises when a problem involves several fuzzy sets to represent different adjectives for single antecedents or consequents. In these cases, depending on the fuzzy rules, a number of consequents can be fired according to the inputs to the system. As seen in Figure 1, the input—represented by the dotted vertical black line—is associated with three fuzzy triangular sets or antecedents, where it "activates" two of them. According to a set of fuzzy rules, the inputs then fire a set of triangular fuzzy sets that represent the consequents, as seen in Figure 2.

The fuzzy sets that represent the consequents are cut, and new shapes are obtained using those cuts, as represented by the green shapes in Figure 2. These shapes are aggregated and result in the output of the fuzzy inference system, and this result can then be defuzzified using different methods,

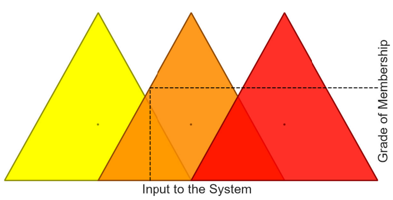


FIGURE 1. **Example of antecedents in a Mamdani fuzzy system.**

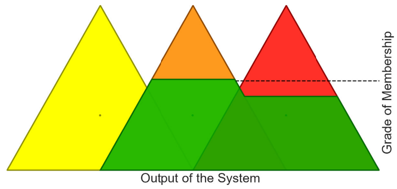


FIGURE 2. **Example of consequents in a Mamdani fuzzy system.**

such as obtaining the centroid of the shape. In this example, a Mamdani fuzzy inference system is considered; in the case of a Sugeno system, for example, the antecedents would be represented by arbitrary mathematical functions, instead of membership functions representing shapes such as the triangles in the example presented above.

C. INTUITIONISTIC FUZZY SETS

In contrast to the traditional fuzzy sets discussed in Subsection II-A, intuitionistic fuzzy sets consider a grade of non-membership in addition to a grade of membership associated to an element in the fuzzy set [15], as expressed by (1).

$$A^* = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in E\} \tag{1}$$

For every one of the elements contained in an intuitionistic fuzzy set, equation (2) must hold true.

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \tag{2}$$

Intuitionistic fuzzy sets are an extension to traditional fuzzy sets, as any traditional fuzzy set can be expressed as a particular case of an intuitionistic fuzzy set, as in (3).

$$\{\langle x, \mu_A(x), 1 - \mu_A(x)\rangle | x \in E\} \tag{3}$$

If the sum of the membership $\mu_A(x)$ and non-membership $\nu_A(x)$ of an element is less than 1, the concept of indeterminacy or hesitancy arises [15], which is described by (4). Indeterminacy is used to represent doubt in the grade of membership of an element in an intuitionistic fuzzy set and is described by (4).

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x) \tag{4}$$

Traditional fuzzy sets can be extended to increase their capabilities of representing uncertainty by introducing the concept of footprint of uncertainty [42], [43]. A footprint of uncertainty is achieved by extending the membership function, where each value transforms from a crisp value into a fuzzy set. Indeterminacy serves a different purpose than that of footprint of uncertainty. Instead of extending the uncertainty provided by traditional fuzzy sets, indeterminacy helps to model doubt. For example, if traditional fuzzy sets can model the following sentence: "the object is very hot", indeterminacy can model "it is unsure that the object is very hot".

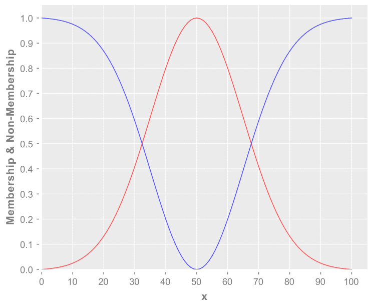


FIGURE 3. Traditional fuzzy set represented as an intuitionistic fuzzy set.

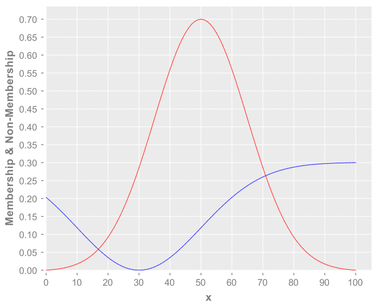


FIGURE 4. Intuitionistic fuzzy set with membership and non-membership functions with different means and standard deviations.

D. INTUITIONISTIC FUZZY SYSTEMS

Intuitionistic fuzzy sets, like traditional fuzzy sets, can be used to create inference systems. Antecedents and consequents in the system can be handled by intuitionistic fuzzy sets, as in a Mamdani system [13]; alternatively, they can be used solely for the antecedents, with mathematical functions used for the consequents as in a Sugeno system [41].

Various approaches have been taken in the past by different authors on how to build intuitionistic fuzzy systems. The authors of the present paper have worked in a certain way to achieve this type of system, and this method is described in [18] and [17]. The method described in the aforementioned works is presented in this Subsection for the reader as a reference implementation.

As it is explained in [18], in an IFIS, in order for an antecedent to fire a consequent according to a set of fuzzy rules, the final grade of membership of an element has to be expressed in terms of its grade of membership and its grade of non-membership. The resulting grade of membership of an element belonging to A is represented by $i\mu_A(x)$, and is defined in (5).

$$i\mu_A(x) = (\nu_A(x) + \mu_A(x))\mu_A(x) \tag{5}$$

To perform an alpha-cut in a consequent, one has to separate it in two stages: i) first, perform a traditional alpha-cut in the membership function following equation (6), and then ii) perform an alpha-cut in the non-membership function

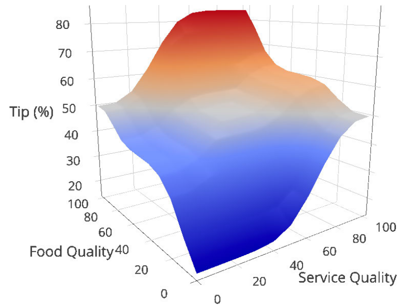


FIGURE 5. Output surface for the tipping problem using a traditional fuzzy system.

following equation (7).

$$\alpha(\mu(x), \mu_\alpha) = \begin{cases} \mu(x), & \text{if } \mu(x) \leq \mu_{alpha} \\ \mu_\alpha, & \text{otherwise} \end{cases} \tag{6}$$

$$\alpha_{NMF}(\nu(x), \mu_\alpha) = \begin{cases} \nu(x), & \text{if } \nu(x) \geq \nu(\mu_{alpha}) \\ \nu(\mu_{alpha}), & \text{otherwise} \end{cases} \tag{7}$$

The aggregation of the fired consequents is performed by applying (8) on the alpha-cuts.

$$A \cup B = \{\langle x, max(\mu_A(x), \mu_B(x)), \times min(\nu_A(x), \nu_B(x))\rangle | x \in E\} \tag{8}$$

The final modification to the traditional inference process in a FIS is made to the center of area procedure. The equation to calculate the center of area of a traditional fuzzy set is (9). In order to implement a center of area for an intuitionistic fuzzy set, one has to incorporate the concept of $i\mu(x)$, giving as a result (10), and its simplification form (11).

$$A_{CoA} = \frac{\sum_{i=1}^{N} \mu(x_i)x_i}{\sum_{i=1}^{N} \mu(x_i)} \tag{9}$$

$$A_{iCoA} = \frac{\sum_{i=1}^{N}(\mu(x_i) + \nu(x_i))\mu(x_i)x_i}{\sum_{i=1}^{N}(\mu(x_i) + \nu(x_i))\mu(x_i)} \tag{10}$$

$$A_{iCoA} = \frac{\sum_{i=1}^{N} i\mu_A(x)x_i}{\sum_{i=1}^{N} i\mu_A(x)} \tag{11}$$

E. MULTI-AGENT SYSTEMS AND AGENT-BASED MODELS

In our method, fuzzy systems are used to model a trader's knowledge and how they take trades according to a market's current conditions, and MAS are used to model the collective traders of a market and their respective influence on the price of it. MAS create agent-based models, which represent a problem that people can analyze and infer new knowledge from it [34].

Agents can be seen themselves as programs that interact with their environment, which may include other agents. MAS are composed of different, autonomous entities with "beliefs" and "rules". Beliefs are used by agents to arrive to an interpretation of their environment, and rules are used to arrive to actions to be performed by the agent towards

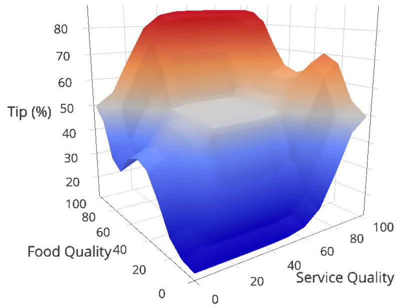


FIGURE 6. **Output surface for the tipping problem using an intuitionistic fuzzy system.**

their environment. Agents in the system are constantly assessing their environment to determine what actions to take according to their beliefs and rules. MAS have the objective of solving a practical problem, unlike agent-based models which are more focused on simulating it. Both use the same tools, and only intent is different.

The proposed method involves the use of a MAS which acts in a decentralized fashion. However, some of its mechanisms are centralized: the outputs of the agents are averaged, and agents receive their inputs from the same source.

III. PROPOSED METHOD

The Overview, Design Concepts, and Details (ODD) protocol for describing Agent-Based Models (ABMs) are now broadly accepted and used to document individual and multi-agent models [44]. In the following section, we present a detailed description of the proposed method's structure and dynamics approaching the ODD protocol.

A. OVERVIEW

1) PURPOSE AND PATTERNS

We present a method for creating forex market predictive models using multi-agent and fuzzy systems, which have the objective of simulating the interactions that provoke changes in the price. Agents in the system represent traders performing buy and sell orders in a market, and fuzzy systems are used to model the rules followed by traders performing trades in a live market and intuitionistic fuzzy logic to model their decisions' indeterminacy.

Fuzzy systems are used to model a trader's knowledge and how they take trades according to a market's current conditions, and MAS are used to model the collective traders of a market and their respective influence on the price of it.

2) ENTITIES, STATE VARIABLES AND SCALES

Multi-agent systems are used to model the collective traders of a market and their respective influence on the price of it.

B. DESIGN CONCEPTS

1) BASIC PRINCIPLES

Agent rules are defined by Mamdani intuitionistic fuzzy inference systems. Fuzzy systems are convenient, as they can be interpreted, as opposed to, for example, neural networks, where the weights associated to the neurons and the connections among themselves become obscure to interpretation. Furthermore, the use of intuitionistic fuzzy systems provides an additional layer for interpretation: indeterminacy.

Indeterminacy arises as a consequent of the inclusion of non-membership to a fuzzy system (see Subsection II-C). This concept allows the designer of a fuzzy system to model doubt or hesitancy in a data set. In the proposed method, indeterminacy is obtained in a heuristical manner, as part of the optimization algorithm that searches for a combination of agents that are used to create a predictive model.

2) INDIVIDUAL DECISION-MAKING

The membership functions in the fuzzy systems are always defined as Gaussian functions, although in future experiments this design choice can change, as other types of membership functions could provide benefits over Gaussian membership functions, such as better interpretability for particular problems being modelled, and improvements in computational cost. Gaussian membership functions were chosen because of their ability to model knowledge in a smoother way than their alternatives, such as triangular or trapezoidal membership functions. Although other membership functions can be better suited for certain situations, the proposed method is currently designed for the creation of predictive models for arbitrary data sets, where solutions are found using iterated local search (ILS).

In the proposed method, the mean of each Gaussian membership function is equal to a random data point from the training data set, while the spread of the Gaussian membership functions that represent a fuzzy rule will be equal to the standard deviation of the aforementioned randomly chosen data points. At least two Gaussian membership functions are used to describe each agents' rule antecedents and consequents for, as obtaining the standard deviation of only one data point would be equal to 0, which would generate Gaussian membership functions with spreads equal to 0. In the case of the antecedents, the means are equal to data points from the training data set that represent inputs, while outputs in the training data set are used as the means for the membership functions that form the consequents. This is expressed by (12), where $\bar{x}$ represents the mean of sample inputs from the training data set, s represents their standard deviation and $\mu(x)$ represents a Gaussian membership function.

$$\mu(x) = e^{-\dfrac{(x-\bar{x})^2}{2s}} \tag{12}$$

We decided to use this method for choosing the means of the Gaussian membership functions in order to guarantee that these functions perfectly describe a relationship between a set of inputs and a set of outputs from the training dataset. A heuristic process still takes place in the optimization stage of our method, as the effectiveness of an agent will depend on what input sets get chosen to be used to describe their rules.

The input grades of memberships obtained by an agent will affect membership functions obtained from different input data sets, so the ILS is indirectly looking for input and output data sets that work well together.

Data points are used as the mean of each membership function to guarantee an agent's ability to react to at least those input values, and thus every agent will respond to at least one data point, which is a common practice in certain methods, such as rule-based classifiers [45], [46]. On the other hand, using the standard deviation of those chosen data points as the spread of the Gaussian membership functions guarantees that uncertainties associated to each membership function will affect—in terms of fuzzy inferences—at least one of the other membership functions. If none of the membership functions were affecting, in any way, the rest of the membership functions, the use of a fuzzy system to represent the agent rules would be meaningless.

The domain of each membership function is not fixed as it is determined by the chosen data points and their standard deviation. The domain of a membership function is defined by the set given by (13), where $\bar{x}$ represents the mean of sample inputs from the training data set and s represents their standard deviation. As a consequent of the previous definition, the domain of either the antecedents or the consequents in a fuzzy system is defined by the set given by (14), where $\bar{x}_{min}$ and $\bar{x}_{max}$ represent the minimum and maximum means—from the set of means obtained from the training data set—used to define the membership functions, respectively, and s_{min} and s_{max} represent the minimum and maximum standard deviations that are associated to $\bar{x}_{min}$ and $\bar{x}_{max}$, respectively.

$$\{ x \mid \bar{x} - s \leq x \leq \bar{x} + s \} \tag{13}$$

$$\{ x \mid \bar{x}_{min} - s_{min} \leq x \leq \bar{x}_{max} + s_{max} \} \tag{14}$$

The number of inputs in the fuzzy systems of each agent can vary. As an agent can potentially have dozens of inputs, associating linguistic variables to these inputs becomes inconvenient.

As the agents' rules are represented by intuitionistic fuzzy systems, the core of the membership functions is not necessarily equal to 1—or it does not exist, if one considers the core of a membership function to be restricted to a value of 1—, as it is the case in traditional fuzzy systems [47]. The value of the greatest grade of membership in a membership function is determined heuristically using an optimization algorithm that is explained in Subsection III-B5.

Indeterminacy is used to "fuzzify" specialization thresholds associated to inputs to the agent, which determine if the agent should respond to that input or not. Two agents with the same fuzzy rules, membership functions and specialization thresholds will respond differently to the same inputs if they have different non-membership functions. This way, the system uses uncertainty—coming from membership functions—and indeterminacy—coming from non-membership functions—to model the membership of an input and a specialization function, respectively. Considering the agent system architecture proposed by Shoham [48], in the proposed method an agent's fuzzy system represents an agent's rules, while its specialization functions represent an agent's beliefs.

Inputs to an agent's fuzzy system are associated to grades of membership—as is usual in fuzzy inference systems—and this grade of membership can either belong or not to a set Λ, which is defined by (15), where $\mu(x)$ represents the grade of membership associated to an input x and λ represents a specialization threshold. An example of a set Λ can be seen in Figure 7. If a grade of membership can be found in a set Λ, then the agent will consider that input to be used to fire a consequent in its agent rules. It is noteworthy that different specialization thresholds λ and non-membership functions can be associated to each of the membership functions present in the antecedents of an agent's fuzzy system.

$$\boldsymbol{\Lambda} = \{ \mu(x) \mid \mu(x) \geq \lambda \} \tag{15}$$

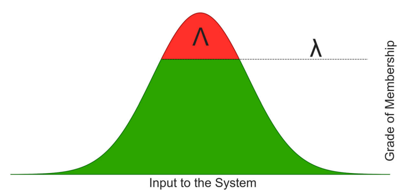


FIGURE 7. Depiction of a specialization threshold λ and a set Λ in a membership function.

An agent's actions can be greatly limited by its specialization functions, as having a single input associated to a grade of membership which does not belong to the specialization interval Λ is sufficient for an agent to take no action. This behavior allows to precisely control the magnitude of specialization of each agent in a predictive model, as the designer of the model—such as an optimization algorithm—can assign specialization thresholds and non-membership functions that restrict an agent to be activated to only a handful of inputs from a training data set. Furthermore, specialization functions work as a coordination mechanism for the agents in a predictive model, as they prevent certain agents from taking action in situations where they would perform sub-optimally and others would perform optimally.

The design of each agent's fuzzy system follows the general architecture shown in Figure 8. Each agent can have an arbitrary number of rules, and there can be an arbitrary number of agents in a predictive model, with rules defined heuristically.

3) PREDICTION

Predictive models that follow the proposed method are formed by a set of one or more agents constructed using the architecture described in Subsection III-B2, resulting in a MAS.

Agents must work together in order to simulate a financial market. One way of achieving this is to obtain the output of every agent in the predictive model and to use an aggregation

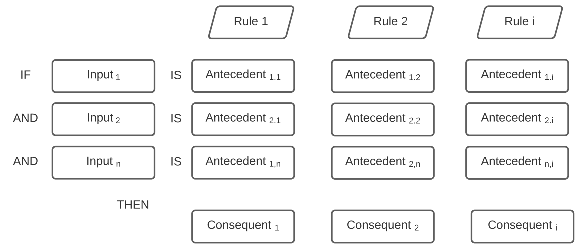


FIGURE 8. **Architecture of an agent's fuzzy system.**

process to unify them into one single output. Instead, the proposed method uses specialization functions to restrict what agent outputs are used to respond to a set of inputs. This is similar to what happens in a real market: the aggregation of all the trades, from every trader, result in the current price of that market. Furthermore, although traders could decide how to trade a market at any given point, traders sometimes restrict themselves from trading because they consider a market's current condition to not be ideal.

The restriction imposed by the specialization functions ensures that every agent is specialized at different subsets of the training data set. During the creation of the agent, the agent is tested with each of the input data points from the training data set to compile a list of specialization levels—i.e., the grades of membership associated to each of the inputs, considering the membership functions in the antecedents of the agent's rules. Once the list of specialization levels is compiled, the list is sorted by using the sum of the specialization levels.

4) INTERACTION

After compiling the sorted list of specialization levels, the proposed method chooses one of them, from highest to lowest specialization, according to a *depth* parameter. This process is depicted in Algorithm 1.

Algorithm 1 Selection of Specialization Threshold

procedure activation-threshold(A, I);
  $ant_A \leftarrow extract\_antecedents(A)$
  $c \leftarrow 0$
  for each $doinp \in \mathcal{I}$
    $acts \leftarrow specialization(ant_A, inp)$
    $sum[c] \leftarrow \sum_{i=1}^{n} acts_i$
    $c \leftarrow c + 1$
  end for
  return $sum(DEPTH)$
end procedure

The chosen specialization level serves as a specialization threshold, as any set of inputs that do not activate all the membership functions according to the specialization level will fail to activate the agent to take an action. As a consequent—during the training stage of the method—any agent will be activated to a number of inputs equal to the value of the *depth* parameter shown in Algorithm 1.

An implication of the aforementioned process is that specialization functions also create a restriction for the actions of the agents in a predictive model. This restriction works as a coordination mechanism to ensure agent participation in the prediction process never reaches one hundred percent, as certain inputs will not trigger a response from any of the agents. In other words, the predictive model only outputs a response if, and only if, the agents have learned a pattern with a strong resemblance to certain inputs.

Although the specialization functions cause the agents to be specialized at responding to a number of inputs, multiple agents could respond to the same set of inputs. In this case, the outputs of the agents—positive or negative numbers, which represent buy or sell orders, respectively—are averaged. We have the hypothesis that specialization functions improve the performance of the models for the forecasting of forex markets, as agents only respond to those inputs where grades of membership in the fuzzy system's antecedents are the highest or, in other words, agents are restricted to respond to those inputs that are similar to those used to create the agents' membership functions.

The process of agent specialization using specialization functions is well suited for the creation of predictive models where it is not desired to obtain a response for any arbitrary set of inputs. In the case of financial market forecasting, this is translated to a recommendation of not trading a particular market, i.e. an unknown pattern is arising and the trader following the recommendations from the predictive model should be wary.

However, it must be noted that the proposed method can be extended to the creation of predictive models that always yield a response. This can be achieved by selecting the agent that is closer to being activated.

Finally, we believe that the specialization functions found with the proposed method could be used for other methods, particularly neural network-based architectures.

5) STOCHASTICITY

The sets of inputs and outputs that are used as the means of the membership functions in the agent rules are determined by using random data points from the dataset—as we explained in Subsection III-B2. In order to obtain a predictive model, ILS is used to find combinations of agents that generate a suitable simulation of a financial market.

The proposed method uses a basic search algorithm that adds and/or removes agents from a list of agents, where these agents work together to create a simulation of the market. The list of agents begins at iteration 0 with a single, random agent, and in the following iterations it is randomly decided to either add new agents or remove them from the list. The modifications are committed only if the addition or removal operation improves the performance of the predictive model, and the algorithm finishes after a number of iterations has

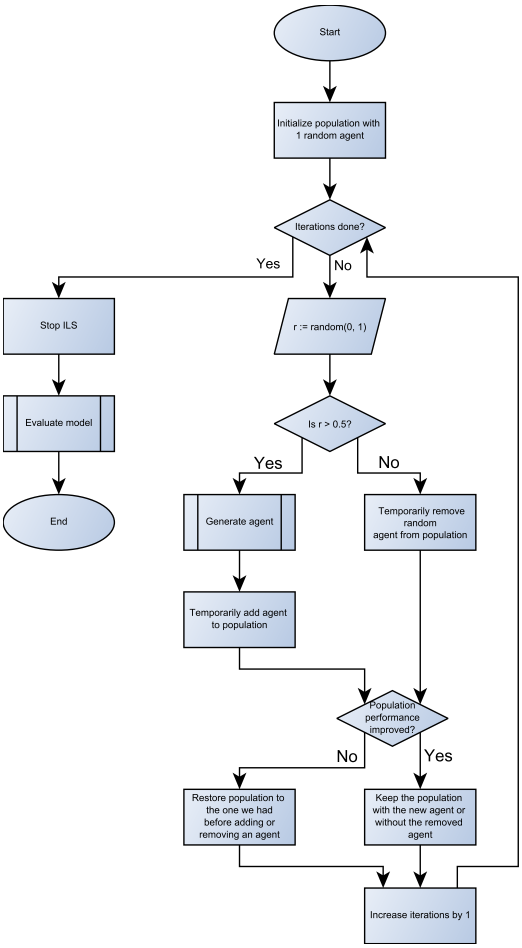


FIGURE 9. **Flowchart for the iterated local search.**

been reached. This process is described by the flowchart in Figure 9.

The performance improvement is measured with a loss function, which is used to determine if a predictive model is better than its predecessor.

As is mentioned in Subsection II-E, the resulting predictive model does not necessarily respond to every set of inputs. This behavior is accentuated when the model is tested against a data set that is different than the one used for the training stage of the method—i.e., a testing data set—, as inputs from this data set could not resemble at all any of those present in the training data set. We have the hypothesis that this behavior helps the generated models achieve better performances.

The agents in the prediction models are represented as objects with the following properties: agent's rules, which stores an intuitionistic fuzzy system; specializations, which stores the antecedents of the fuzzy system; and specialization thresholds, which stores the levels that need to be surpassed by a set of inputs in order to activate the agent.

Indeterminacy in an agent's fuzzy system is determined randomly and optimal values are found with the ILS method described in Figure 9. In order to obtain intuitionistic fuzzy sets where (2) holds true, our implementation generates membership functions where the greatest possible grade of membership is M and non-membership functions where the greatest possible grade of non-membership is $1 - M$. The mean and spread of the Gaussian membership functions are equal to the mean and spread of the Gaussian non-membership functions in all cases.

The list below summarizes the design concepts of the predictive models generated by our proposed method.

- Emergence
  - Models find patterns that yield the lowest predictive error through specialization functions.
- Adaptation
  - Agents evolve to specialize in similar patterns (same market trend/direction, and similar input and output magnitudes)
- Fitness
  - Minimization of RMSE
- Prediction
  - The mean of the agents' outputs represents an estimation of the next price of a market being modelled
- Sensing
  - Raw market prices (open, high, low, close) and timestamps when these prices occurred are sensed (fed as inputs) by the agents in the predictive models
  - Different beliefs are associated to each agent in a predictive model, which make the agent preprocess raw market prices differently from other agents in the model
- Interaction
  - There is no interaction among the agents when it comes to taking decisions or processing data coming from their environment
  - Agents' outputs are averaged in the end to create a final price prediction for a market
- Stochasticity
  - Agents' rules are constructed using random data points taken from a training data set
- Observation
  - The predictive models generate a time series which represents a series of predictions for future market prices

C. DETAILS

1) INITIALIZATION

The agents that are randomly generated for the ILS need to pass a test before they can be considered a candidate to be part of a predictive model. Considering a training data set, an agent's specialization functions are evaluated against the inputs set obtained from that training data set in order to generate a set of specializations, where each specialization is the equivalent of calculating the grade of membership associated to a set of inputs.

After performing this step, all the specializations are summed to obtain a score S that represents the intensity to which that agent reacts to that set of inputs. This score is defined by (16), where N represents the number of inputs, and $\mu(x_i)$ represents each of the grades of membership associated to each input. The scores associated to each set of inputs are ordered in a descending manner, so the first elements are the scores that represent those sets of inputs that would fire an agent's consequents the most. The ordered scores list is also used to obtain the outputs that are associated to these ordered scores, so we can know what are the outputs that the agent *should* evaluate to given those particular sets of inputs. The resulting set of outputs are used to determine if the agent is a suitable candidate; if most of the outputs associated to the highest scores have the same direction (negative or positive), then it is a suitable candidate. This mechanism ensures that the chosen agents are specialized in similar inputs that yield similar outputs. For our implementation, the highest scores have to be associated to outputs that share the same direction at least 66% of the time. This value was chosen empirically after performing some preliminary tests, where different values greater than 50%—as we want the majority of the outputs to share the same direction—were used. However, it is not statistically proven that this value will yield better results than other values.

$$S = \sum_{i=1}^{N} \mu(x_i) \tag{16}$$

The optimization process for the predictive models loads configuration files as its first step, depending on the market that we want to use to obtain training and testing data sets. These configuration files set parameters for the proposed method, such as the train-test ratio and number of inputs, outputs and number of rules for the agents' fuzzy systems.

For our implementation of the proposed method we chose Common Lisp as our language, so we could use a software library for the creation of intuitionistic fuzzy systems that we implemented in the past [17], [18]. All the populations are compressed and stored in a PostgreSQL database, which enables us to resume the optimization of a model at any time. Storing the populations in a database also enables us to use populations of agents to be tested in other data sets, as well as to extract agents from certain populations to be used in other prediction models. The source code of our implementation can be found in the git repositories at this link: `https://bitbucket.org/overmind-group/workspace/projects/OT`.

The evaluation of the agents proved to be a computationally expensive task, as the system needs to evaluate hundreds of fuzzy systems per iteration in the optimization process. For this reason, we implemented a caching system where the output of an agent is stored in memory using a technique called *memoization* [49] in a functional programming context. After *memoizing* an agent, if that agent is required to be evaluated with exactly the same inputs as the ones used during the *memoization* process, then we can assume that the output will be the same, and thus we can simply query for the output stored in memory. The caching system prevents our implementation from re-evaluating hundreds of fuzzy systems in the optimization process.

TABLE 1. Multi-agent systems parameters.

Market	# of Inputs	# of Rules	Inputs
EUR/USD	9	3	HH_3, LH_3, CH_3
USD/JPY	9	6	HH_3, LH_3, CH_3
USD/CHF	12	2	HH_4, LH_4, CH_4
GBP/USD	9	3	HH_3, LH_3, CH_3
USD/CAD	12	2	HH_4, LH_4, CH_4
AUD/USD	9	3	HH_3, LH_3, CH_3
# of iterations		100	
Loss function		RMSE	

IV. EXPERIMENTS

In order to determine the efficacy of the proposed method, we designed experiments that involved data sets and performance metrics used in the work by Munkhdalai *et al.* [50]. In [50], daily prices for the following Forex markets are used: EUR/USD, USD/JPY, USD/CHF, GBP/USD, USD/CAD and AUD/USD. The last year of prices are used as their data set, although they do not provide the exact starting and ending dates. Their datasets are partitioned into three parts: training, validation and test sets, where the training set corresponds to 80% of the data set, the validation set corresponds to 10% of the data set, and the test set correspond to 10% of the data set. The authors used a 5-fold time series cross-validation method to obtain their performance metrics.

For our experiments, we decided to use random samples of 63 trading days—which corresponds to a quarter of the trading days in a year—for our data sets, which can be extracted from the last 5000 trading days (from August 28th 2004 to May 18th 2020), for the same Forex markets used in [50]. We did not use exactly the same data sets as Munkhdalai *et al.*, because they do not provide the starting and ending dates for their data sets, and because a bigger data set—around 15 years of data, instead of 1 year of data—provides a better challenge for avoiding accidental ''cherry picking'' [51] when choosing optimal hyperparameters for the method.

Each of the data points in the datasets includes open, high, low, close prices, as well as a timestamp and the volume of the asset exchanged associated to each day (data point) in the dataset. The data was obtained from Tiingo,[1] an online financial market data provider. Tiingo states that their market

[1] https://api.tiingo.com

TABLE 2. Comparison between our results and the ones obtained by Munkhdalai *et al.*, using MAE as the loss function.

Model	Activation function	EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
RNN	Sigmoid	0.0060	5.4000	0.0081	0.0126	0.0060	0.0079
	Swish	0.0044	0.4925	0.0042	0.0074	0.0068	0.0043
	Munkhdalai et al.	0.0043	0.4479	0.0037	0.0083	0.0047	0.0034
GRU	Cosine	0.0098	2.0191	0.0130	0.0396	0.0106	0.0162
	Linear	0.0044	0.4475	0.0041	0.0083	0.0052	0.0041
	Munkhdalai et al.	0.0044	0.5327	0.0039	0.0062	0.0046	0.0071
LSTM	ReLU	0.0053	1.4920	0.0054	0.0101	0.0058	0.0046
	Swish	0.0045	0.5243	0.0049	0.0065	0.0063	0.0054
	Munkhdalai et al.	0.0046	0.5200	0.0060	0.0069	0.0061	0.0044
MLP	ReLU	0.0049	0.7499	0.0058	0.0150	0.0052	0.0038
	Swish	0.0043	0.7351	0.0039	0.0073	0.0055	0.0042
	Munkhdalai et al.	0.0047	0.6114	0.0059	0.0066	0.0049	0.0041
Random forest		0.0053	0.5209	0.0061	0.0156	0.0059	0.0044
AdaBoost		0.0059	0.6440	0.0103	0.0158	0.0063	0.0066
XGBoost		0.0059	0.4958	0.0048	0.0156	0.0064	0.0045
SVM		0.0304	0.4718	0.0376	0.0294	0.0099	0.0176
Ours without SF		0.0050	0.4903	0.0041	0.0073	0.0050	0.0048
Ours with SF		**0.0033**	**0.1810**	**0.0019**	**0.0038**	**0.0033**	**0.0027**

data has been cleaned from erroneous and missing prices, so we are assuming this to be true and using their data verbatim.

These data sets are split into two parts, a training data set which corresponds to 70% of the data set, and a test data set which corresponds to 30% of the data set. As a consequence, a validation step was not involved in our experiments. Regarding the performance metrics, we provide results using MAE and RMSE, in order to compare against the results presented in [50]. A total of 30 experiments were performed for each forex market, and we provide the means and standard deviations for each of our performance metrics in Section V.

In [50], the results of several predictive models are provided. In order to obtain the hyper-parameters of the different algorithms, random searching was used, with the exception of deep learning neural networks (recurrent neural networks (RNN), long short-term memory (LSTM) neural networks, and gated recurrent unit (GRU) neural networks), where the learning rate was set at 0.001 using the Adam optimizer [52], batch size at 64 instances for each iteration, MSE as the loss function, and maximum number of epochs at 3000 for the first fold, and then they used 300 epochs with the first pre-trained model for the remaining folds. Regarding multi-layer perceptrons (MLP), the authors used an input layer of 5 neurons (for the prices of the last 5 days), a hidden layer of 16 neurons and an output layer of 1 neuron (for the prediction of the next day's price). In addition to neural network-based algorithms, [52] also provides results for models obtained by random forest, AdaBoost, XGBoost and SVM architectures.

Finding optimal values for the hyper-parameters—what input variables to consider for the agents—of our method is a challenge due to the high number of combinations that can lead to favorable results. We decided to find arguments through trial and error for these parameters that allowed our method to generate models that yielded results in the same order of magnitude as the results presented by its competing methods. In the end, we arrived to the arguments presented in Table 1. In this table, *HH* means "high height", and represents the price difference between the high and the greater price between close or open prices of a trading day; *LH* means "low height", and represents the price difference between the low and the lesser price between close or open prices of a trading day; *CH* means "candle height" and represents the absolute price difference between the open and close prices of a trading day; and the subscript following each of the aforementioned abbreviations represents the number of past trading days that were considered.

The output variable for the agents represents the price difference [53] between a current session's close price's and the next future close price. These outputs—like an agent's inputs—are obtained from the training dataset. This price difference can either be positive—which represents a "buy prediction"—or negative—which represents a "sell prediction".

V. RESULTS

Tables 2 and 3 show subsets of the results presented by Munkhdalai *et al.* [50], comparing them with the results of the method presented in this paper. In the case of neural networks (RNN, GRU, LSTM and MLP), the table shows the results obtained by Munkhdalai *et al.*, as well as the worst and best results that are not obtained by using the activation function proposed by Munkhdalai *et al.* In addition to the neural network-based results, results for the predictive models based on random forest, AdaBoost, XGBoost and SVM are also provided. The purpose of this table is to provide a comparison between the predictive models generated by

TABLE 3. Comparison between our results and the ones obtained by Munkhdalai *et al.*, using RMSE as the loss function.

Model	Activation function	EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
RNN	Sigmoid	0.0079	6.0677	0.0097	0.0188	0.0075	0.0090
	Swish	0.0059	0.6435	0.0062	0.0099	0.0085	0.0054
	Munkhdalai et al.	**0.0057**	0.5993	0.0059	0.0098	0.0062	**0.0045**
GRU	Cosine	0.0129	2.6072	0.0168	0.0548	0.0133	0.0187
	Linear	0.0058	0.6054	0.0061	0.0104	0.0066	0.0052
	Munkhdalai et al.	0.0059	0.6871	0.0060	0.0083	0.0060	0.0082
LSTM	ReLU	0.0069	1.6890	0.0079	0.0155	0.0074	0.0058
	Swish	0.0061	0.6890	0.0072	0.0088	0.0081	0.0069
	Munkhdalai et al.	0.0062	0.6757	0.0087	0.0092	0.0078	0.0055
MLP	ReLU	0.0064	0.9711	0.0081	0.0217	0.0066	0.0048
	Swish	0.0058	0.9360	0.0061	0.0097	0.0070	0.0054
	Munkhdalai et al.	0.0061	0.7722	0.0082	0.0088	0.0064	0.0053
Random forest		0.0068	0.6781	0.0081	0.0244	0.0075	0.0056
AdaBoost		0.0076	0.8198	0.0136	0.0238	0.0080	0.0082
XGBoost		0.0076	0.6554	0.0071	0.0241	0.0085	0.0058
SVM		0.0332	0.6387	0.0394	0.0329	0.0125	0.0200
Ours without AF		**0.0057**	**0.5391**	**0.0047**	0.0079	**0.0057**	0.0052
Ours with AF		0.0071	0.5630	0.0051	**0.0078**	**0.0057**	0.0058

TABLE 4. Results of our method with and without specialization functions to restrict their actions.

Metric		EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
	n	60	60	60	60	60	60
	n_{SF}	50	51	56	51	50	44
MAE	Mean	5.03×10^{-03}	4.90×10^{-01}	4.13×10^{-03}	7.34×10^{-03}	5.05×10^{-03}	4.83×10^{-03}
	Std dev	2.79×10^{-03}	2.20×10^{-01}	1.94×10^{-03}	4.01×10^{-03}	2.24×10^{-03}	2.12×10^{-03}
MAE_{SF}	Mean	3.27×10^{-03}	1.81×10^{-01}	1.95×10^{-03}	3.80×10^{-03}	3.32×10^{-03}	2.74×10^{-03}
	Std dev	3.81×10^{-03}	1.31×10^{-01}	1.53×10^{-03}	3.12×10^{-03}	1.94×10^{-03}	2.25×10^{-03}
	$t-value$	-2.71555	-9.13914	-6.74278	-5.2259	-4.3400	-4.7952
	Conclusion	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0	Reject H_0
RMSE	Mean	5.69×10^{-03}	5.39×10^{-01}	4.69×10^{-03}	7.93×10^{-03}	5.70×10^{-03}	5.18×10^{-03}
	Std dev	3.18×10^{-03}	2.51×10^{-01}	2.09×10^{-03}	4.83×10^{-03}	2.64×10^{-03}	2.34×10^{-03}
$RMSE_{AF}$	Mean	7.12×10^{-03}	5.63×10^{-01}	5.11×10^{-03}	7.79×10^{-03}	5.70×10^{-03}	5.81×10^{-03}
	Std dev	5.15×10^{-03}	2.98×10^{-01}	3.21×10^{-03}	4.05×10^{-03}	2.36×10^{-03}	3.72×10^{-03}
	$t-value$	1.7104	0.4543	0.8288	-0.1661	0.0000	0.9890
	Conclusion	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0

our proposed method and the predictive models generated by other methods.

Our results can be found in the last rows of the table, and the best result for each market is shown in bold. It must be noted that no statistical testing was performed when comparing our method against the ones provided by Munkhdalai *et al.*; the results in the table have the purpose of showing the competitiveness of our method to some extent, in terms of error, in order to justify further research to improve our current method. Two results are presented for our method: one where specialization functions (SF) are used to restrict agents from certain trades, and another for agents not using specialization functions to restrict their decisions. It is noteworthy that these results cannot provide conclusive proof that one method is better than another, as the testing datasets and methods are not the same. Table 4 provides again the means of our results, in addition to their standard deviations, sample sizes and statistical tests.

TABLE 5. Parameters used for the hypothesis tests.

Parameter	Value
Confidence interval	95%
H_a	$\mu_1 < \mu_2$
H_0	$\mu_1 \geq \mu_2$
Critical t	-1.9958

Finally, Table 6 shows a comparison of our method against a buy and hold strategy. For the buy and hold, we subtract the closing price of May 18th 2020 to the closing price of August 28th 2004, which represents a buy order that was opened on the former date and that was closed on the later date.

TABLE 6. Comparison of our method in terms of revenue, with and without SF, against buy and hold strategy.

Strategy	EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
Buy and hold	-0.1194	-2.4735	-0.3109	-0.5800	0.0897	-0.0571
Ours without SF	-0.0068	-0.6736	0.0095	0.0439	-0.0388	0.0349
Ours with SF	0.0137	2.7859	0.0056	0.0344	0.0100	0.0238

VI. CONCLUSION

Financial markets are examples of complex systems, which is why they can be better represented using agent-based models. The advantage of these models, which are capable of simulating the simultaneous operations and interactions of multiple agents, is that they can recreate and predict the behavior resulting from the complex phenomena of emergence. These models have the additional advantage of being interpretable and can give us additional knowledge about the dynamics of the market.

In this work, we propose a MAS that models agents' knowledge as intuitionistic fuzzy inference systems, along with specialization functions that allow agents to become specialized for trading particular market conditions. The MAS was adjusted to obtain a forex market predictive model by using our proposed ILS to optimize the models' parameters.

The elements of our method enable the generation of predictive models that present a unique set of features. Agents in the MAS represent the fundamental mechanic behind financial markets, which is the traders. The agents use intuitionistic fuzzy systems to represent their trading rules, which allows the method to model a trader's uncertainty and indeterminacy. Specialization functions work as a mechanism for coordination among the agents: we can choose what agents are more appropriate for particular market conditions. Lastly, fuzzy systems and MAS provide the foundations for more interpretable predictive models, compared to other methods such as neural networks or SVM.

To evaluate our model's predictive performance, we compare it against models generated by RNN, LSTM, GRU, MLP, random forest, AdaBoost, XGBoost, and SVM. The results lead us to conclude that our method proves to be competitive for predicting forex markets, as shown in Table 2, where our method performed the best for every forex market, if our models use specialization functions. This demonstrates that using specialization functions to create agents that are specialized at particular market conditions help to increase the generated models' efficacy.

We can also observe in Table 2 that our method achieves competitive predictive errors when RMSE is used as its performance metric. Specialization functions seem to increase the predictive error when RMSE is used, but this is misleading, as the error distribution in our method is not Gaussian—which is the case for our method, as agents not always decide to take trades—, and in these cases it has been found that MAE is a better metric [54]–[56].

As an additional test, we compared our method against a buy and hold strategy. Table 6 shows that our method performs better than buy and hold for all Forex markets—in terms of revenue—, and that the use of specialization functions improves the results of our method.

In addition to the low predictive errors obtained, it should be noted that our method has the advantage of being more interpretable than the method proposed by Munkhdalai *et al.* The agents in our predictive models can be displayed as membership functions being activated by inputs, along with their consequents being alpha-cut, aggregated and deffuzified to a final output value. The user can easily examine the rules that the agents are following and thus have a clear picture of how the agents react to the prices. For example, consider the following fuzzy rules:

1) if RSI(x) is LOW, then BUY is HIGH
2) if RSI(x) is MEDIUM, then BUY is MEDIUM
3) if RSI(x) is HIGH, then BUY is LOW

If an agent is performing very well during the current market conditions, we can then assume that there is a relationship between the technical indicator *RSI* and upward directional movements. After examining multiple agents, we can draw more complex conclusions about a market.

As can be concluded, choosing the correct technical indicators or any other functions that preprocess the agents' inputs is crucial for generating knowledge that can be interpreted by the user. Furthermore, the selection of these functions, along with every other value for the parameters of our method is equally as important. Currently, our implementation does not have a mechanism for the selection of optimal values for these parameters. A user of our method would need to search by trial and error for values that yield desirable results according to the nature of the problem they want to model.

VII. FUTURE WORK

The experiments presented in this work demonstrate that better methodologies need to be used in order to provide conclusive proof that the use of specialization functions in the agents' fuzzy systems is beneficial for the creation of predictive models. Additionally, we can test our specialization functions in other methods, such as neural networks. Furthermore, we could design experiments that demonstrate which version of our method helps real traders make better decisions.

The proposed method was tested using a subset of all the forex markets currently available. More experiments could be performed where additional forex markets are tested. Moreover, our method should be tested with financial markets of different natures, such as stocks, bonds, commodities and metals.

An advantage of fuzzy systems over other modelling techniques is that fuzzy systems are interpretable. Natural language processing techniques that use the fuzzy systems as inputs could be used to provide texts that describe the conditions of a market, as perceived by the agents in the MAS.

Our implementation should be adapted to perform in a distributed manner, where agents are evaluated in parallel in different CPU cores or even different physical machines. A distributed architecture will help achieve results faster, which will allow us to test different approaches for the creation of predictive models using our method.

Finally, the ILS method used for optimization in the proposed method should be benchmarked against other optimization techniques, such as genetic algorithms or particle swarm optimization. Finding better optimization algorithms for our method can help us achieve better results, and probably faster. It is also probable that lower errors could be achieved, as we do not know if our current optimization method is exploring a wide enough search space.

REFERENCES

[1] G. Liu and X. Wang, "A numerical-based attention method for stock market prediction with dual information," *IEEE Access*, vol. 7, pp. 7357–7367, 2019.

[2] P. Fernández-Blanco, D. J. Bodas-Sagi, F. J. Soltero, and J. I. Hidalgo, "Technical market indicators optimization using evolutionary algorithms," in *Proc. 10th Annu. Conf. Companion Genetic Evol. Comput. (GECCO)*, 2008, pp. 1851–1858.

[3] Y. Alsubaie, K. E. Hindi, and H. Alsalman, "Cost-sensitive prediction of stock price direction: Selection of technical indicators," *IEEE Access*, vol. 7, pp. 146876–146892, 2019.

[4] D. L. Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, and H. Moon, "Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network," *IEEE Access*, vol. 6, pp. 55392–55404, 2018.

[5] D. Cabrera, C. Cubillos, A. Cubillos, E. Urra, and R. Mellado, "Affective algorithm for controlling emotional fluctuation of artificial investors in stock markets," *IEEE Access*, vol. 6, pp. 7610–7624, 2018.

[6] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.

[7] W. Cao, W. Zhu, and Y. Demazeau, "Multi-layer coupled hidden Markov model for cross-market behavior analysis and trend forecasting," *IEEE Access*, vol. 7, pp. 158563–158574, 2019.

[8] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[9] Y. Chen, W. Lin, and J. Z. Wang, "A dual-attention-based stock price trend prediction model with dual features," *IEEE Access*, vol. 7, pp. 148047–148058, 2019.

[10] J.-A. Jiang, C.-H. Syue, C.-H. Wang, J.-C. Wang, and J.-S. Shieh, "An interval type-2 fuzzy logic system for stock index forecasting based on fuzzy time series and a fuzzy logical relationship map," *IEEE Access*, vol. 6, pp. 69107–69119, 2018.

[11] X. Sang, Y. Zhou, and X. Yu, "An uncertain possibility-probability information fusion method under interval type-2 fuzzy environment and its application in stock selection," *Inf. Sci.*, vol. 504, pp. 546–560, Dec. 2019, doi: 10.1016/j.ins.2019.07.032.

[12] K. Chourmouziadis and P. D. Chatzoglou, "Intelligent stock portfolio management using a long-term fuzzy system," *Appl. Artif. Intell.*, vol. 33, no. 9, pp. 775–795, Jul. 2019, doi: 10.1080/08839514.2019.1630124.

[13] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Machine Stud.*, vol. 7, no. 1, pp. 1–13, Jan. 1975.

[14] M. Abdulgader and D. Kaur, "Evolving Mamdani fuzzy rules using swarm algorithms for accurate data classification," *IEEE Access*, vol. 7, pp. 175907–175916, 2019.

[15] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets Syst.*, vol. 21, no. 1, pp. 87–96, Aug. 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011486800343

[16] K. T. Atanassov, "Intuitionistic fuzzy sets past, present and future," in *Proc. EUSFLAT Conf.*, vol. 1, Sep. 2003, pp. 145–149. [Online]. Available: http://ifigenia.org/mediawiki/images/d/d3/EUSFLAT-2003-012-019.pdf

[17] A. Hernandez-Aguila, M. Garcia-Valdez, O. Castillo, and J. J. Merelo-Guervós, "An open source implementation of an intuitionistic fuzzy inference system in Clojure," *IEEE Int. Conf. Fuzzy Syst.*, Jul. 2017, pp. 1–6.

[18] A. Hernandez-Aguila, M. Garcia-Valdez, and O. Castillo, "A proposal for an intuitionistic fuzzy inference system," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2016, pp. 1294–1300. [Online]. Available: http://ieeexplore.ieee.org/document/7737838/

[19] O. Castillo and K. Atanassov, "Comments on fuzzy sets, interval type-2 fuzzy sets, general type-2 fuzzy sets and intuitionistic fuzzy sets," in *Recent Advances in Intuitionistic Fuzzy Logic Systems*. Cham, Switzerland: Springer, 2019, pp. 35–43.

[20] K. Sun, Q. Jianbin, H. R. Karimi, and Y. Fu, "Event-triggered robust fuzzy adaptive finite-time control of nonlinear systems with prescribed performance," *IEEE Trans. Fuzzy Syst.*, early access, Mar. 6, 2020.

[21] K. Sun, L. Liu, J. Qiu, and G. Feng, "Fuzzy adaptive finite-time fault-tolerant control for strict-feedback nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 786–796, Apr. 2021.

[22] M.-C. Tsai, C.-H. Cheng, and M.-I. Tsai, "A multifactor fuzzy time-series fitting model for forecasting the stock index," *Symmetry*, vol. 11, no. 12, p. 1474, Dec. 2019.

[23] S. Zeng, S.-M. Chen, and M. O. Teng, "Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm," *Inf. Sci.*, vol. 484, pp. 350–366, May 2019, doi: 10.1016/j.ins.2019.01.071.

[24] S. Rajab and V. Sharma, "An interpretable neuro-fuzzy approach to stock price forecasting," *Soft Comput.*, vol. 23, no. 3, pp. 921–936, Sep. 2017.

[25] A. Vlasenko, N. Vlasenko, O. Vynokurova, Y. Bodyanskiy, and D. Peleshko, "A Novel ensemble neuro-fuzzy model for financial time series forecasting," *Econometrics Theory Pract.*, vol. 4, no. 3, pp. 439–453, 2019, Art. no. 126.

[26] W. Yue, Y. Wang, and H. Xuan, "Fuzzy multi-objective portfolio model based on semi-variance–semi-absolute deviation risk measures," *Soft Comput.*, vol. 23, no. 17, pp. 8159–8179, Sep. 2019, doi: 10.1007/s00500-018-3452-y.

[27] W. Witayakiattilerd, "Fuzzy quantitative analysis method for stock selection into portfolio," *Chiang Mai J. Sci.*, vol. 46, no. 4, pp. 799–811, 2019.

[28] N. Mansour, M. S. Cherif, and W. Abdelfattah, "Multi-objective imprecise programming for financial portfolio selection with fuzzy returns," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112810, doi: 10.1016/j.eswa.2019.07.027.

[29] M. Gao, S. Yang, and L. Sheng, "Distributed fault estimation for time-varying multi-agent systems with sensor faults and partially decoupled disturbances," *IEEE Access*, vol. 7, pp. 147905–147913, 2019.

[30] C. Yu, X. Chen, C. Wang, H. Wu, J. Sun, Y. Li, and X. Zhang, "An improved platform for multi-agent based stock market simulation in distributed environment," *IEICE Trans. Inf. Syst.*, vol. E98.D, no. 10, pp. 1727–1735, 2015.

[31] M. M. Barbieri and J. O. Berger, "Optimal predictive model selection," *Ann. Statist.*, vol. 32, no. 3, pp. 870–897, Jun. 2004.

[32] P. Taveeapiradeecharoen and N. Aunsri, "Dynamic model averaging for daily forex prediction: A comparative study," in *Proc. Int. Conf. Digit. Arts, Media Technol. (ICDAMT)*, Feb. 2018, pp. 321–325.

[33] B. Lebaron, "A builder's guide to agent-based financial markets," *Quant. Finance*, vol. 1, pp. 254–261, Mar. 2001.

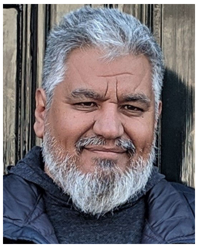
[34] R. Grothmann, "Multi-agent market modeling based on neural networks," Ph.D. dissertation, Fac. Econ., Univ. Brehemen, Bremen, Germany, Dec. 2002. [Online]. Available: http://lsc.fie.umich.mx/~juan/Materias/Cursos/ANN/Papers/multi-agent-market-modeling.pdf

[35] Y. Li, C. Wu, J. Liu, and P. Luo, "A combination prediction model of stock composite index based on artificial intelligent methods and multi-agent simulation," *Int. J. Comput. Intell. Syst.*, vol. 7, no. 5, pp. 853–864, Sep. 2014.

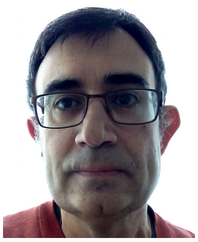
[36] X. Chen and S. Tokinaga, "Multi-agent-based modeling of artificial stock markets by using the co-evolutionary GP approach," *J. Oper. Res. Soc. Jpn.*, vol. 47, no. 3, pp. 163–181, 2004.

[37] J.-J. Kim, S.-H. Cha, K.-H. Cho, and M. Ryu, "Deep reinforcement learning based multi-agent collaborated network for distributed stock trading," *Int. J. Grid Distrib. Comput.*, vol. 11, no. 2, pp. 11–20, Feb. 2018.
[38] L. Wei, W. Zhang, X. Xiong, and Y. Zhao, "A multi-agent system for policy design of tick size in stock index futures markets," *Syst. Res. Behav. Sci.*, vol. 31, no. 4, pp. 512–526, Jul. 2014.
[39] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–352, Jun. 1965. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001999586590241X
[40] R. Kruse, J. E. Gebhardt, and F. Klowon, *Foundations of fuzzy systems*. Hoboken, NJ, USA: Wiley, 1994.
[41] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.
[42] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 117–127, Apr. 2002. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/995115/
[43] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 6, pp. 808–821, Dec. 2006. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4016089/
[44] V. Grimm, S. F. Railsback, C. E. Vincenot, U. Berger, C. Gallagher, D. L. DeAngelis, B. Edmonds, J. Ge, J. Giske, J. Groeneveld, A. S. A. Johnston, A. Milles, J. Nabe-Nielsen, J. G. Polhill, V. Radchuk, M.-S. Rohwäder, R. A. Stillman, J. C. Thiele, and D. Ayllón, "The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism," *J. Artif. Societies Social Simul.*, vol. 23, no. 2, pp. 1–20, 2020.
[45] S. Zhao, E. C. C. Tsang, D. Chen, and X. Wang, "Building a rule-based classifier—A fuzzy-rough set approach," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 624–638, May 2010.
[46] M. Almutairi, F. Stahl, and M. Bramer, "A rule-based classifier with accurate and fast rule term induction for continuous attributes," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 413–420.
[47] M. Wygralak, "An axiomatic approach to scalar cardinalities of fuzzy sets," *Fuzzy Sets Syst.*, vol. 110, no. 2, pp. 175–179, Mar. 2000.
[48] Y. Shoham, "Agent-oriented programming," *Artif. Intell.*, vol. 60, no. 1, pp. 51–92, 1993.
[49] M. Johnson, "Memoization of top down parsing," 1995, *arXiv:9504016*. [Online]. Available:https://arxiv.org/abs/cmp-lg/9504016
[50] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.
[51] J. M. Morse, "'Cherry picking': Writing from thin data," *Qualitative Health Res.*, vol. 20, no. 1, p. 3, 2010, doi: 10.1177/1049732309354285.
[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[53] C. W. J. Granger and R. Joyeux, "An introduction to long-memory time series models and fractional differencing," *J. Time Ser. Anal.*, vol. 1, no. 1, pp. 15–29, Jan. 1980.
[54] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, 2014.
[55] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, no. 1, pp. 79–82, 2005.
[56] C. J. Willmott, K. Matsuura, and S. M. Robeson, "Ambiguities inherent in sums-of-squares-based error statistics," *Atmos. Environ.*, vol. 43, no. 3, pp. 749–752, Jan. 2009.

AMAURY HERNANDEZ-AGUILA received the Ph.D. degree in computer science and the M.Sc. degree in computer science from the Tijuana Institute of Technology, Mexico, in 2014 and 2019, respectively. He is currently obtaining a postdoctoral training with the Tijuana Institute of Technology, researching how multi-agent systems and fuzzy logic can be used for the prediction of financial markets.



MARIO GARCÍA-VALDEZ is currently a full-time Research Professor with the Tijuana Institute of Technology. His research interests include the personalization of interactive systems, voluntary computing, parallel evolutionary computation, and interactive evolutionary computation.



JUAN-JULIÁN MERELO-GUERVÓS received the degree in theoretical physics from the University of Granada and the Ph.D. degree in physics from the University of Granada, in 1994. He is currently a Professor with the University of Granada. His main research interests include evolutionary algorithms, open source software, and complex systems.

MANUEL CASTAÑÓN-PUGA received the bachelor's degree in engineering and the master's degree in computer sciences from the Tijuana Institute of Technology, Tijuana, Mexico, and the Ph.D. degree in computer sciences from the Autonomous University of Baja California, Mexico. He is currently a Professor with the Autonomous University of Baja California. His research of modeling and simulation, agent-based simulation, hybrid-intelligent agents, and multi-agent systems explores the way in which software agents could be used to describe multidimensional environments, innovation, evolution, and adaptation in complex adaptive systems. He collaborates with multidisciplinary researchers and scientists to create multidimensional computer simulations of societies, political ideologies, trading economies, and urban landscapes. His research also intends to incorporate the ideas of complexity into the mainstream of engineering and in particular to its instruction at the undergraduate level.

OSCAR CASTILLO LÓPEZ (Senior Member, IEEE) received the Doctor of Science (Doctor Habilitatus) degree in computer science from the Polish Academy of Sciences. He is currently a Professor of computer science with the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. He is also serving as the Research Director of computer science and the Head of the Research Group on Hybrid Fuzzy Intelligent Systems. He received the Recognition as the Highly Cited Researcher by the Clarivate Analytics, in 2017, and the Web of Science, in 2018. He is also the President of Hispanic American Fuzzy Systems Association (HAFSA), the Past President of International Fuzzy Systems Association (IFSA), and the Chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). He also belongs to the Technical Committee on Fuzzy Systems of IEEE and to the Task Force on Extensions to Type-1 Fuzzy Systems. He is also an Associate Editor of the *Information Sciences* Journal, *Applied Soft Computing* Journal, Journal of *Engineering Applications of Artificial Intelligence*, *Granular Computing* Journal, and the *International Journal of Fuzzy Systems*.