A Novel Method Based on Fuzzy Inference Systems, Multi-agent Systems and Genetic Programming for the Forecast of Financial Markets

Amaury Hernandez-Aguila, Mario Garcia-Valdez

Index Terms

Economic forecasting, multi-agent systems, fuzzy systems, genetic programming, decision support systems.

Abstract

A novel method is proposed for the forecast of financial markets. The method can forecast the direction and strength a market is going to follow, and can also give a recommendation of what market the user should invest on. As a consequent, the method can act as a regression technique and as a decision support system. This method involves the use of a multi-agent system where the agents' functions are generated by a genetic programming algorithm. The only two operators used in this algorithm are the sine and the sum operators, and thus, the membership functions are generated by a sum of sines. These agents' functions are converted to membership functions of fuzzy inference systems that act as the decision makers for the agents. Several experiments were performed in order to support the validity and efficacy of the system, and the results demonstrate that the proposed method is a viable technique to forecast financial markets.

Amaury Hernandez-Aguila is with Instituto Tecnológico de Tijuana

Mario Garcia-Valdez is with Instituto Tecnológico de Tijuana

I. Introduction

Since many decades ago, financial traders have implemented statistical models to forecast the time-series, and a field called Technical Analysis was created [1]. This type of market analysis is characterized by using past financial data to describe certain aspects of a market, such as its volatility, trend, and momentum [2]. These indicators receive the name of Technical Indicators (TI). Some examples of TI are those created by Welles Wilder, such as Relative Strength Index (RSI) and Average Directional Index [3], and an example of an oscillator (a type of TI which usually tells if a market is overbought or oversold) is the Stochastic Oscillator [4], created by George Lane. This TI are mentioned in particular as they are used as input to the Proposed Method in this work. In contrast to Technical Analysis, Fundamental Analysis relies on the examination of the underlying forces that affect a particular financial market (or several of them), instead of just relying on the price movements in a time-series.

A more modern approach to perform these analyses is to use Machine Learning (ML) techniques. Using ML algorithms, researchers create regression models using Technical or Fundamental Indicators as trainig datasets [5]. Examples of regression techniques are autoregression [6], symbolic regression [7], and linear regression [8]. Other more elaborated techniques exist in ML for regression or curve-fitting tasks, such as the use of Artificial Neural Networks [9] and Support Vector Regression [10]. But a problem that often arises with these models is that they can't take into account emergent phenomena: these models learn from past behaviors and can't predict what they haven't "seen." An approach that has shown excellent results in simulating these behaviors is Agent-based Modeling (ABM). Citing the work by Bonabeau [11], ABM is, "by its very nature, the canonical approach to modeling emergent phenomena," and for this reason, it is a suitable tool to model complex systems [12], such as financial markets. A concise definition of ABM can be found in the work by Gilbert [13]: "Agent-based modeling is a form of computational social science... One creates some kind of simplified representation of 'social reality' that serves to express as clearly as possible the way in which one believes that reality operates." In particular, ABM has been proved to be useful to analyze price stability in financial markets [14], scientometrics and domain visualization [15], and in social sciences in general [13], among others.

This work proposes a method based on ABM to construct a Decision Support System (DSS) that aids traders on the process of trading financial markets. DSS is not a well defined term, as for some researchers is just an interactive system for managers, and for others the focus is on understanding and improving the decision process [16]. Since its inception, DSS have been created to aid managers in business-related areas [17] [18], and one can see specialized cases where DSS are used to support the decision process when forecasting financial markets, such as in the work by Tsang, Yung and Li [19]. The proposed method is aimed at improving the decision process of trading financial markets: the trader receives a recommendation of what financial market to trade, as in the work by Brown, Pelosi, and Dirska [20]. But the created algorithm goes deeper than just giving recommendations; in the end, a novel method for performing regression was created, and the authors believe that it can be extended to perform classification tasks. The method uses a combination of Fuzzy Inference Systems (FIS), Genetic Programming (GP) [21] [22] and Multi-agent Systems (MAS) [23] to produce models that can explain the behavior of complex systems such as that of a financial market.

The differences between ABM and MAS are subtle, but have to be noted before proceeding. According to Niazi, and Hussain [15], MAS is a sub-domain of ABM; the principal objective of ABM is to provide an explanation of a phenomenon through the interaction of agents, while MAS provides a specific application of ABM in order to solve a practical problem. The MAS developed in this work belongs to a class of Coalitional Games called Constant-sum game [23], and, in particular, it is a zero-sum game. Agents in the proposed method are selfish agents that don't intervene with other agents, but there's a higher order process that is dictating how they must organize. In this case, the collective work of every agent must give a "zero-sum": the sum of their forces must be equal to the observed price, or in other words, they must adjust to the real prices. To achieve this, each agent uses a FIS to act intelligently, and its Membershif Functions (MF) are found using GP. Specifically, the predicates and consequents of the FIS are generated using sum of sines that are found using a modified algorithm based on GP. The modified algorithm implements communities of agents that perform different operations among their agents: crossover, migration, replace and mutation. In the end, the developed algorithm gives as result the community of agents that obtained the absolute value of the lowest Mean-Squared Error (MSE) plus the sum of its agents scores (explained in Section III). At first, the authors of this work had the hypothesis that the developed system would enable the user to examine the MF of each agents' FIS to develop new economic theories, but, even though the system achieved remarkable results, the generated MF are too chaotic to be interpreted with ease. The novelty of the method lies in the use of GP to automatically create Membership Functions for the given inputs. Although the currently generated MF are too chaotic, this behavior can be changed by modifying the GP operators and how these generated functions are transformed into MF. Another useful feature that the Proposed Method will enable, is that the user of the system can easily inject features as inputs to the FIS of the agents, and compare their behaviour to other communities that don't have these features. This could help the user create new economic theories or find better trading strategies.

The structure of this work is as follows: in Section II an extensive background of related works is given. Special attention was given to this section in order to provide the necessary foundations for the Proposed Method, as it is a new algorithm based on many different technologies and ideas. In Section III the theory of the Proposed Method is given, explaining every aspect of its specification. A number of extensive experiments are presented in Section IV, along with their results. The experiments were carefully designed in order to provide enough information to support the efficacy of the Proposed Method. Conclusions about the presented work are given in Section V.

II. Related Work

The layout of this section starts, in Subsection II-A with the mention of a series of works about Financial Forecasting and some of the different techniques that have been used to effectively describe and predict financial markets. Several and different approaches of the design of MAS are presented in a series of works in Subsection II-B. Methods related to the Proposed Method that used Evolutionary Algorithms in order to optimize or find better architectures in the development of forecasting models are presented in Subsections II-C and II-D. It is important to pay special attention to the works presented in Subsection II-D, as the Proposed Method specifically uses Genetic Programming as part of its algorithm. The next Subsection II-E presents works where Fuzzy Logic is used in the forecast of financial markets. Finally, a brief mention of some works about Decision Support Systems is presented in Subsection II-F.

A. Financial Forecasting

In general, this work involves the use of Machine Learning to forecast financial markets. In several efforts, researchers create regression models using Technical or Fundamental Indicators as trainig datasets. Examples of regression techniques are autoregression [6], symbolic regression [7], and linear regression [8].

The work by Brown, Pelosi and Dirska [20] uses a Niche Genetic Algorithm called Dynamic-radius Species-conserving Genetic Algorithm (DSGA) to select stocks to purchase from the Dow Jones Index. It is important to mention this work because, in the end, the DSS that is presented in the Proposed Method does the same kind of recommendation as in their work. More importantly, Brown, et al., uploaded the dataset that the authors of the present work used to perform the different experiments. In Section IV a comparison to their work is provided, along with many other experiments.

The work by Lu, Lee, and Chiu [24] point out the complexity of financial time-series. They note its noisy nature and propose a technique to reduce this noise based in a two-stage modeling approach using Independent Component Analysis (ICA) and Support Vector Regression (SVR). Their approach first uses ICA for generating independent components to identify and remove those containing the noise, then the remaining components are used to reconstruct the forecasting variables which now contain less noise and are the input of the SVR forecasting model. Their work was important for the development of the Proposed Method, as we believe that the ABM approach can then be used to diminish the noise in the market, by using a separate class of agents dedicated to model it.

Lastly, it is imperative to mention the use of Neural Networks in regression tasks, as it is a technique that has been proved to be very effective for this kind of problems. O'Connor and Madden [5] obtained some remarkable results where they obtained an annual 23.5% of Rate of Investment on Dow Jones data used for training and testing. Another example is given by Castillo and Melin [25], where they compare different hybrid architectures that combine Neural Networks and Fuzzy Logic for the prediction of financial time-series.

B. Multi-agent Systems

The core algorithm of the Proposed Method is, at its highest level, a Multi-agent System. It is therefore paramount to mention some works which use MAS for the forecast or understanding of financial markets.

Klingert and Meyer [26] implement a MAS to analyze the effect of two market mechanisms: the continuous double auction and logarithmic market scoring rule. The purpose of the agent-based simulation model is to see the effect on the number of trades, the accuracy of prediction markets and the standard deviation of the prices in order to prove three hypothesis that they propose. In the end, due to a higher amount of trades and lower standard deviation of the price, their results indicate that the logarithmic market scoring rule seems to have an advantage over the other mechanism.

Sherstov and Stone [27] present three automated stock-trading agents which follow different strategies to predict financial markets, and are compared. The first agent uses Reinforcement Learning, the second a Trend-following strategy, and the last one Market-making. These agents are part of a MAS where the better performing agent is chosen for the testing phase. It is noteworthy to mention that their strategy was used in a live competition and won.

Kendall and Su [28] use a MAS to simulate stock markets within which stock traders are modeled as heterogeneous adaptive artificial agents. On average, 80% of the artificial stock traders were able to trade using successful trading strategies which brings the investors higher returns compared to a simple buy-and-hold strategy.

The authors of this work gained useful knowledge about MAS from two theses. The first one is the work from Grothmann [29], "Multi-agent Market Modeling based on Neural Networks." This work served as inspiration for the architecture of the Proposed Method. The second thesis is Boer-Sorbán's "Agent-Based Simulation of Financial Markets," which gave an overview of approaches to describe and understand financial market's dynamics, and motivated the authors of this work to use the approach of Agent-based Computation to perform financial forecast.

As a final mention, Samanidou, et. al. [30], provides the reader a very comprehensive overview of Agent-based Modeling, where different techniques to perform this kind of models are discussed.

C. Genetic Algorithms

In the Proposed Method, Genetic Programming is used to generate the Membership Functions (MF) of the Fuzzy Inference Systems that act as the agents' functions. The use of Evolutionary Algorithms to generate MF has been proposed before in several works. What follows is the mention of two works which use Genetic Algorithms to perform such a task, and in the next Subsection, one can find more specialized works where Genetic Programming is used.

Thrift [31] explores a nowadays widely used technique which involves the use of a Genetic Algorithm (GA) to discover the parameters of the Membership Functions (MF) in a Fuzzy Inference System to obtain a better performance. Homaifar and McCormick [32] go further and use GA to simultaneously design the MF and the rule sets for fuzzy logic controllers.

D. Genetic Programming

The previous Subsection served as a little introduction to this Subsection, where Genetic Programming (GP) is used to optimize architectures or perform regression tasks in financial forecast applications. GP has been previously used in many fields of economics. Commonly, as mentioned before, GP is used to create regression models or as a mean to find a better architecture in trading strategies. In the Proposed Method, GP is used to generate the Membership Functions of the Fuzzy Inference Systems that serve as the agents' functions in the MAS, an uncommon technique, as far as the authors of this work know. What follows is a set of works of some common uses of GP in financial applications.

Li and Tsang [33] developed a system that generates decision trees of Technical Indicators using GP. Preliminary results showed that it outperforms commonly used, non-adaptive, individual technical rules with respect to prediction accuracy and average annualized rate of return over two different out-of-sample test periods (three and a half year in each period).

Garcia-Almanza and Tsang [34] used GP as a regression tool and a technique called Repository Method to model "rare instances" or emergent phenomena. The Repository method is a technique that analyses decision trees produced by GP to discover classification rules. It lets model the rare instances in different ways, increasing the possibility of identifying similar cases in the future in the time-series. The work of Garcia-Almanza was useful for the authors of the Proposed Method, as it noted the importance of determining this emergent phenomena in financial markets, and stressed the usefulness of ABM to model it.

An artificial market that models technical, fundamental, and noise traders was developed by Martinez-Jaramillo and Tsang [35]. This work resulted interesting as GP is used to generate the agent functions of the technical traders, and Technical Indicators were used as the operators for the GP algorithm.

Chen and Yeh [36] propose an architecture based on GP and ABM that takes trader's (agents) search behavior densities, and by using Simulated Annealing, connect these behaviors to psychological factors, such as peer pressure or economic factors such as the standard of living. Their work is very interesting, as their results, in the end, support the Efficient Market Hypothesis.

As a final mention, Bastian [37] uses GP to identify the input variables, the rule base, and the involved membership functions of fuzzy models.

E. Fuzzy Prediction

As has been previously mentioned, in this work each agent uses a Fuzzy Inference System as its agent's function. This agent's function has the objective to determine how much "trading force" an agent has to provide, so the sum of all the agent forces give a zero-sum system. Fuzzy logic has successfully been used to predict financial markets in the past. This Subsection provides a series of works which use fuzzy logic and other related techniques to the Proposed Method, as a mean to forecast financial markets.

Ijegwa, et al. [38], crated a Fuzzy Inference System that uses four Technical Indicators as input and the output is a recommendation to buy, sell or hold in a financial market. This way, their system aids the trader in the decision making process. They decided to use the Moving Average Convergence/Divergence (MACD), Relative Strength Index (RSI), Stochastic Oscillator (SO) and On-Balance Volume (OBV) as the Technical Indicators that serve as input to the FIS.

Huang, Pasquier, and Quek [39] describe the application of a hierarchical coevolutionary fuzzy system called HiCEFS for predicting financial time-series. Their system is based on the Technical Indicator Price Percentage Oscillator. An interesting part of this work is their use of Irregular Shaped Membership Functions (ISMF). The employment of ISMF allows their system to construct an accurate predictive model, and it outperfrms the simple buy-and-hold strategy.

As a final mention, a robust system that combines MAS, Fuzzy Logic, Genetic Algorithms, and Moving Averages as input to the Fuzzy Controllers was implemented by Gamil, et al. [40]. The MAS helps in the gathering of stock information from different information sources, and aids the processing of the system, as the necessary processing power needed to predict the buy/sell decisions need more power to do the job efficiently, so using different agents for decision support was useful.

F. Decision Support Systems

This final Subsection is small, as the Proposed Method doesn't extensively rely on Decision Support Systems (DSS). Here are presented two works related to DSS and financial forecast, and the design and implementation of DSS in general.

The first work is by Keen [16], which helped the authors of the present work to understand DSS. His work presents valuable definitions related to DSS, as well as design guidelines and a series of case studies that illustrate the process of creation of DSS.

Secondly, Tsang, et al. [41] [42] [19] [43], present a DSS, over a series of works, for financial forecasting called EDDIE. The system serves to improve the odds of a trader to perform successful trades, and is designed as an interactive decision tool, not as a replacement of expert knowledge. The performance of the system depends on the quality of user's input and the efficiency of its GP search engine.

III. Proposed Method

A. Decision Support System

The final outcome of the Proposed Method is a recommendation to buy, sell or hold in a financial market. The system tells the user the direction which the market is going to take (downtrend or uptrend) and its strength, through the use of the Directional Strength indicator (see Subsection IV-A).

Nevertheless, the underlying method can be used to perform regression tasks, and it could be extended to perform classification tasks in a future work. What follows is the explanation of this underlying method.

B. Communities of Agents

As has been noted before, the Proposed Method is based on Agent-based Modeling (ABM). To implement a system based on ABM, the method developed the concept of a Community of agents. A Community is a collection of agents, who are the final actuators in the environment (the simulated financial market). The action of each of the agents in the community involves in providing certain "strength," which is a value between -100 and 100. The sum of all of the forces of the agents in a Community must be equal to the observed or real Directional Strength indicator for a particular record in the dataset. A Community receives its inputs, which are the Technical Indicators ADX(t), SO(t), and RSI(t), and give as output the DS($t+1$) (the Directional Strength of a future record).

The method creates several Communities, which exchange genetic material of its agents among them, in order to produce better performing Communities, and their fitness is determined using a sum of the Mean-Squared Error and the absolute value of the sum of the strengths of its agents. The use of this summation was determined to be a good fitness indicator by trial and error, but other fitness functions could be used.

C. Fuzzy Inference Systems

Each of the agents in the Comunnities has a Fuzzy Inference System (FIS) that acts as its agent function. The inputs of this FIS, as has been mentioned before, are the ADX, SO, and RSI Technical Indicators. The output is a real number between -100 and 100 that represents the Directional Strength (DS). To arrive to this output number, one can choose a number of Membership Functions (MF) that act as inputs and a number of MF that act as outputs. By trial and error, the authors found that a number of 5 outputs gave really good results in most cases. As these MF are generated by a GP algorithm, they do not really have an human understandable interpretation. The FIS just defuzzifies the aggregation of the MF at their activation level determined by its inputs, and this will represent a DS for that particular agent.

The rule set is depicted below, and is comprised of 15 rules: 5 outputs for each of the 3 Technical Indicators.

1) if ADX is GPpred1 then DS is GPcons1
2) if ADX is GPpred2 then DS is GPcons2
3) if ADX is GPpred3 then DS is GPcons3
4) if ADX is GPpred4 then DS is GPcons4
5) if ADX is GPpred5 then DS is GPcons5
6) if RSI is GPpred6 then DS is GPcons6
7) if RSI is GPpred7 then DS is GPcons7
8) if RSI is GPpred8 then DS is GPcons8
9) if RSI is GPpred9 then DS is GPcons9
10) if RSI is GPpred10 then DS is GPcons10

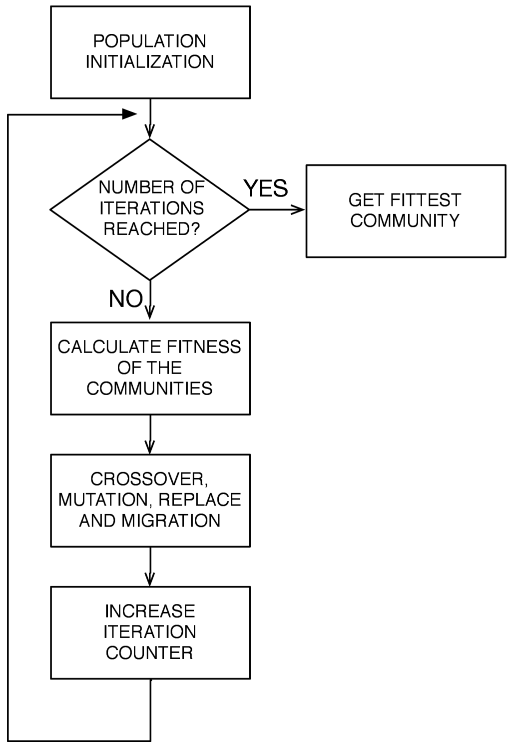
11) if SO is GPpred11 then DS is GPcons11
12) if SO is GPpred12 then DS is GPcons12
13) if SO is GPpred13 then DS is GPcons13
14) if SO is GPpred14 then DS is GPcons14
15) if SO is GPpred15 then DS is GPcons15

1) Genetic Programming: In order to create the MF of agents' FIS, a GP algorithm performs a number of operations among the initially generated Communities. These operations are the classical crossover and mutation operations, along with two new operations: migration and replace. The crossover operation simply chooses a subtree of the GP function of the *i*th agent from a Community A, and passes it to the GP function of the *i*th agent from a Community B. A subtree from the Community B is passed to the GP function of the *i*th agent. As a result, two new agents are generated and replace the old agents. To determine what Communities are to be crossed over, a tournament is held where two random Communities are chosen from the population, and the one with the lowest MSE plus the absolute value of the sum of forces (as explained is Subsection III-B is chosen; this will obtain the first Community. This process is repeated to obtain a second Community. As for the mutation operation, a subtree from a Community A is selected and randomly changed to produce a new GP function. The Community to be mutated is randomly selected from the population.

The migration operation is a rather simple one. According to a migration chance, an agent from a Community A will get interchanged with another agent from a Community B. In order to determine if this interchange is to be performed, the algorithm sees if the sum of the forces of Community A is of different sign than the sum of forces of Community B. This ensures that the migration will produce a Community closer to a zero-sum (the Community will be closer to the observed or real prices). As for the replace operation, a whole new Community is generated and replaces the Community with the highest absolute value of the MSE plus sum of forces. All the four operations have a pre-established chance of occurring (except for the replace chance, which is currently being dynamically adapted by using a FIS. See Subsection III-D). One can see a flowchart of the general process in Figure 1.

In addition to this set of operations, the GP algorithm uses the parameters of population size (in this case, the number of Communities), the number of agents per Community, the number of generations, how many inputs the agents will process (in this case, the Technical Indicators),

Fig. 1: Flowchart of the Proposed Method with Constant Replace Chance



and how many outputs (which, in the end, as has been noted previously, get defuzzified into a Directional Strength value).

The generation of the Membership Functions (MF) is a novel or, at least uncommon, method (the authors of this work could not find any works that use this specific technique). The GP algorithm uses a sum of sines to produce functions that are converted to the MF of the FIS of the agents. An example of the functions which represent these MF are depicted in Figure 2, and a graphical representation of an input MF and an output MF are depicted in Figure 3 and Figure 4, respectively. This means that the GP algorithm uses only two operators: the sum and the sine operators. The sum of sines is represented by the Formula 1.

$$y = \sum_{i=1}^{n} a_i sin(b_i x + c_i) \tag{1}$$

Taking into consideration Formula 1, the GP algorithm uses the following literals set: *a*, *b*, and *c* are random real numbers, where *a* ranges from 0.0 to 0.3, *b* ranges from 0.0 to 10.0, and

Fig. 2: Sum of Sines Functions Representing Membership Functions

```
((((SINN *X 0.23037868996521715 2.7761673700763634 0.4135250087684561)
   (+ (SINN *X 0.262973502540366 6.346144278151511 0.6758001742364756)
      (SINN *X 0.13362033925303674 5.394568163138159 0.06676290904721371)))
  ((SINN *X 0.17010129502753712 1.5965960066197438 0.15040655344928777)
   (SINN *X 0.2741113583452902 3.7814252392188386 0.297874314733259))
  ((SINN *X 0.09544464642924137 6.509390929630825 0.22047088693064598)
   (+ (SINN *X 0.05993492488920584 9.778734305957935 0.5379728556793228)
      (SINN *X 0.13947563057905668 1.4024726972811652 0.9319373467802954)))
  ((SINN *X 0.13463449814838419 2.0257133650136083 0.8845112434389377)
   (SINN *X 0.07162331531657691 1.3861046521468712 0.30080402237879444))
  ((+ (SINN *X 0.25539039003504754 6.119689070466617 0.46720486666099736)
      (SINN *X 0.2525326122562175 1.3867222217564579 0.46918620294630586))
   (SINN *X 0.1299292519802109 8.394003893868975 0.030392542277453582)))
 (((SINN *X 0.01597927164635844 4.573736052014943 0.4737114947536405)
   (SINN *X 0.025589142355023742 8.978166840436355 0.4273956237001104))
  ((SINN *X 0.29129910919922647 2.4819335987965685 0.6088701897179578)
   (+ (SINN *X 0.16152735824636513 8.041031447694639 0.4340122970525053)
      (+ 0.5369457787470397 -0.12995320475047967)))
  ((SINN *X 0.1448525007968042 2.5715218608979673 0.006222140652903052)
   (+ (+ -0.228132728294707 -0.2178175442339354)
      (+ 0.21022838294039525 -0.3630571221528671)))
  ((+ (SINN *X 0.27936428731652 0.9398876244442289 0.6901226629283603)
      (SINN *X 0.13765108730795408 9.199216192862472 0.4700384136176663))
   (+ (SINN *X 0.1933054916231759 5.637394569979072 0.8165322348700774)
      (SINN *X 0.16116533100419977 3.421276680611176 0.8881960443471546)))
  ((SINN *X 0.21362617110948692 4.918543124060637 0.3742906739235934)
   (+ (+ -0.06048168948220911 -0.5846347385464021)
      (SINN *X 0.19038984093691538 3.5009857004473277 0.8913758658870001))))
 (((SINN *X 0.08155621212063024 5.614458425451561 0.1678937350055687)
   (SINN *X 0.06117271239381787 8.26537712686279 0.6398335017442704))
  ((SINN *X 0.12057451843939361 7.447231630178406 0.7211996663308153)
   (SINN *X 0.13016373863339306 2.0896639795396355 0.17999302829656472))
  ((+ (+ -0.04678145818652735 -0.6405007156251812)
      (SINN *X 0.0981792675306891 4.369447904506789 0.6833371220318276))
   (SINN *X 0.03145639584250661 7.674708757092983 0.6714127365207057))
  ((SINN *X 0.22199938514131742 9.835615700542702 0.6054584276656114)
   (SINN *X 0.02099896123469711 7.429261523185158 0.7204574125307763))
  ((SINN *X 0.09600277922604698 5.00596372759574 0.8357216310128355)
   (+ (+ 0.2899202005115409 0.12953967147775258)
      (+ -0.46010185187165553 -0.20582971585427012)))))
```

c ranges from 0.0 to 1.0. These ranges were obtained by trial and error, and other ranges could perform better. An additional real number literal is added to the set, which ranges from -1 to 1, along with the variable x, which takes the value of the current input (ADX, RSI or SO).

D. Dynamic Adaptation of the Replace Chance

The replace chance parameter, instead of being constant, is dynamically adapted using a Fuzzy Inference System (FIS) while the GP process is running. This idea was borrowed from the series of works by Castillo, et al. An example of this dynamic adaptation of parameters can be found in [44], where the authors apply a FIS that controls the value of certain parameters in an Ant Colony Optimization (ACO) algorithm, proposing different architectures. Then these architectures are compared in performance. In the end, the authors prove that the dynamic adaptation of parameters

Fig. 3: Graphical Representation of an Input Membership Function

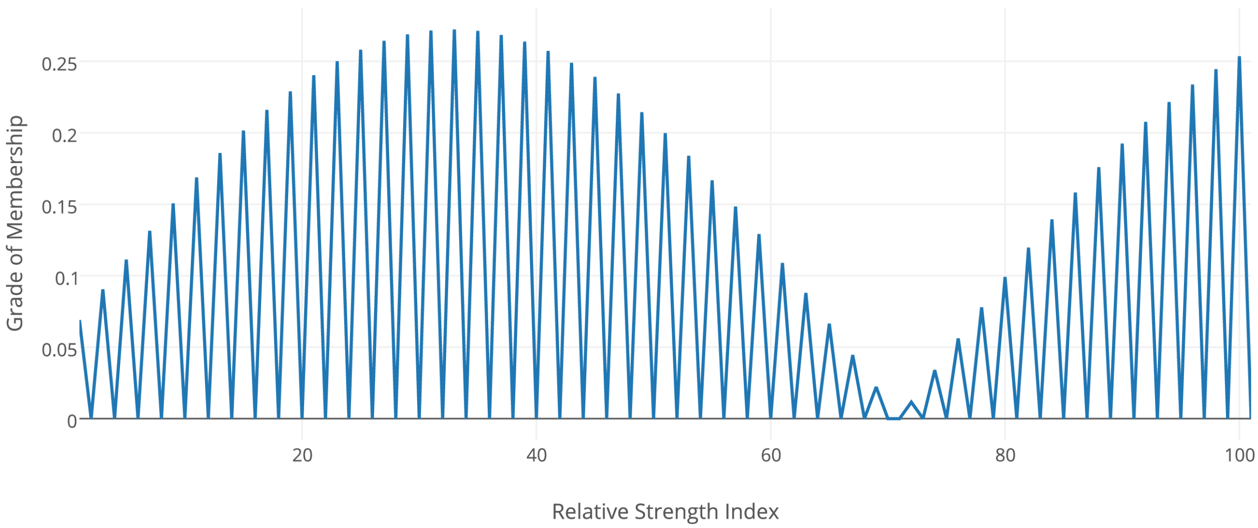
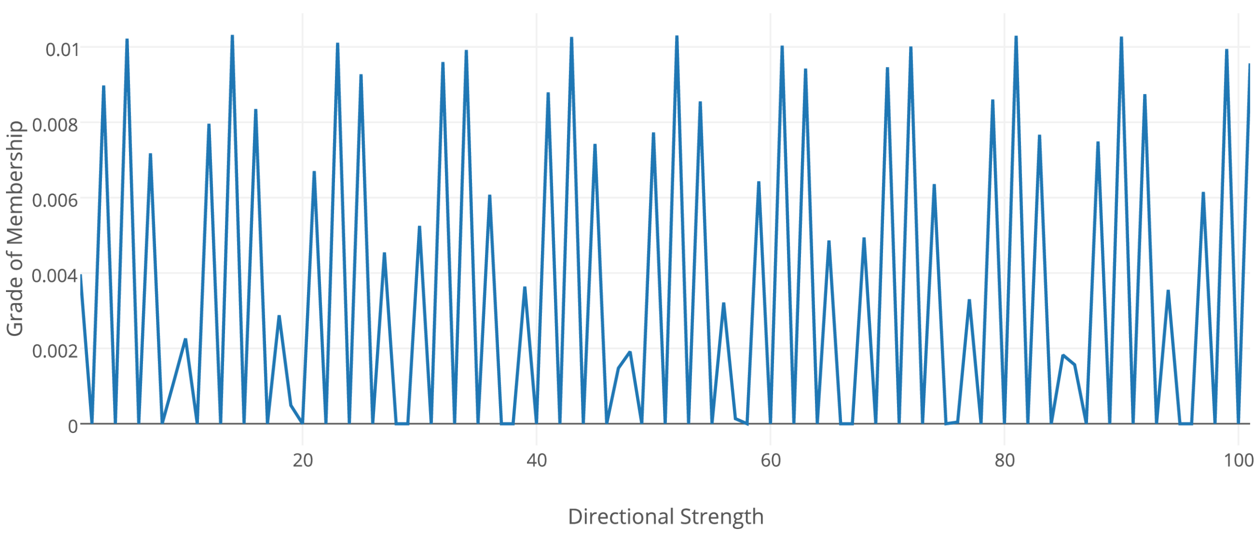


Fig. 4: Graphical Representation of an Output Membership Function



through the use of FIS can greatly improve the performance of ACO. Additionally, one can find a survey of the application of this type of technique in the work by Valdez, Melin and Castillo [45].

In the Proposed Method, a similar approach is taken, where dynamic adaptation of parameters through FIS is used to avoid the stagnation of the GP algorithm. In this case, a stagnated GP algorithm would mean that it has been producing the same fitness over a number of generations, i.e., the absolute value of MSE plus sum of forces is not being lowered in the last N generations. To calculate this stagnation level, the authores of this work propose the Stagnation Index, which is described by the Formula 2, where S(P) means the Stagnation Index of P, P is the set of fitnesses over certain number of generations, and C(P) is the count of repeated fitnesses over the last N generations. For example, given the set P = 5, 5, 5, 3, 2, 1, C(P) = 3.

$$S(P) = 1 - \sqrt{\frac{1}{C(P)}} \tag{2}$$

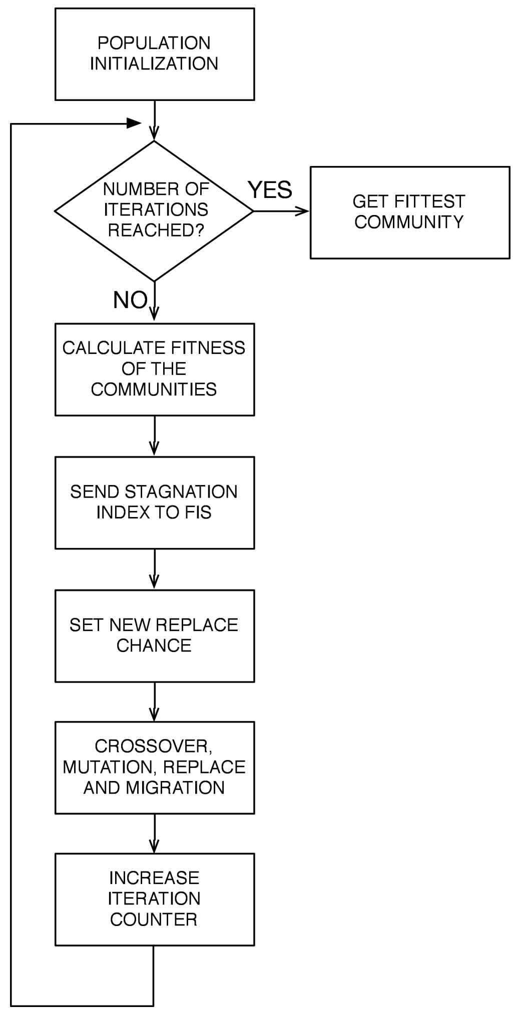
The Stagnation Index is given as input to a FIS, which gives as output a new replace chance for the replace operation. The objective of this dynamic adaptation of parameters is to obtain lower errors in a smaller amount of generations. This is proven in Section IV. Also, it is worth mentioning that this technique could be applied to the crossover, migration, and mutation chances, but due to a limitation of time, the authors of this work could only apply the technique to one parameter.

A new flowchart depicting the modified GP algorithm is shown in Figure 5.

IV. Experiments and Results

This Section begins explaining the dataset used for the training and testing processes. After this explanation, experiments and results obtained from implementing the dynamic adaptation of the replace chance through the use of a Fuzzy Inference System are presented. Then, the Proposed Method is compared against the method proposed by Brown, Pelosi, and Dirska [20], using their proposed dataset. The authors of this work considered that the design of the experiment in [20] were insufficient to prove the efficacy of a novel method such as the one proposed in this work, and the experiments in this Section were expanded in order to better support the efficacy of the system. The experiment by Brown, et al., was expanded to 30 trials, instead of just 8. Furthermore, a series of experiments (where a winning stock has to be chosen, just like in the work by Brown, et al.) with an increasing number of generations were performed, and, again, 30 trials are ran instead of just 8, with 30 different number of generations. Another set of experiments similar to the experiments mentioned before were performed, where different

Fig. 5: Flowchart of the Proposed Method with Dynamic Adaptation of Replace Chance



configurations of the Proposed Method are used. Each of the different configurations is ran for 30 trials, and the average Rate of Return is collected.

After the previous set of experiments were performed, the authors decided to perform experiments where the forecast of individual stocks take place, instead of choosing a stock to buy for each of the weeks in the dataset. This set of experiments is more a regression type of experiments than a recommendation type. Five stocks were chosen where the Proposed Method was applied with different configurations. The effectiveness of the experiments is determined by how much average Rate of Return is collected.

As a final experiment, the Proposed Method performs a pure regression task, where it tries to predict the Directional Strength of the next week's record in the dataset.

A. Dataset

The dataset used in this work was first used by Brown, Pelosi, and Dirska [20], and can be found at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/) under the name of "Dow Jones Index Data Set." In this work we only used the Open, High, Low, and Close prices, and we obtained the Average Directional Index (ADX), Stochastic Oscillator (SO), and Relative Strength Index (RSI) for the dataset records by using an online tool called FreeStockCharts located at the URL http://www.freestockcharts.com/. Basically, we added the Technical Indicators to the stocks provided in the dataset by Brown et al. using FreeStockCharts, and downloaded the full historical data from the platform. Then, a filtering was performed where only the relevant records were extracted (the same records that are present in the benchmarking dataset).

As a final step, another attribute was added to the dataset, which the authors of this work call "Directional Strength." Basically, this attribute tells how strong a movement was at a particular time in a time-series, and what its direction was (downtrend or uptrend). An absolute value of 100 of the Directional Strength means that it was a very strong movement, where all the traders in that financial market agree to buy or sell, where a value of 0 means strong indecision (and, for example, a Candlestick Pattern called Doji could arise). This Directional Strength indicator could have already been mentioned in the literature, but the authors of this work couldn't find a reference to this particular formula. The formula to calculate the Directional Strength is as follows:

$$DS(t) = 100\frac{C(t) - O(t)}{H(t) - L(t)} \tag{3}$$

Where t represents a particular point of time in the time-series, O represents the Open price, H the High, L the Low, and C the Close price.

B. Dynamic Adaptation of the Replace Chance

To compare the efficacy of the Proposed Method with, and without dynamic adaptation of the replace chance, two experiments were performed and then compared using a hypothesis test.

TABLE I: Averages of the 1000 Generations for each of the 30 Experiments

Without Dynamic Adaptation		With Dynamic Adaptation	
1311.36858	1046.104578	1216.510814	1143.389484
1229.470821	1355.991986	1241.307708	1102.711613
1295.573386	1091.08208	1120.805705	1249.783318
1030.289883	1327.723752	1172.208267	1111.148651
1336.959564	1091.20479	972.4929395	1215.476516
1384.057403	1265.682395	1170.073256	1169.588612
1303.139391	1246.964085	1278.311636	1040.093053
1391.94702	1158.730397	1154.370559	762.2223289
1125.607147	1373.704464	1063.93828	1207.359942
1336.603628	1130.282865	1227.164144	1168.572853
1308.989252	1135.924961	1067.38689	1162.467748
1253.008383	1384.15463	1273.537452	1248.3565
1265.489682	1487.437018	960.6685944	1140.389075
1247.161813	1215.320663	1173.268224	1309.74542
1025.353754	1173.167377	1169.425716	978.8573602

30 experiments were run with certain configuration, for 1000 generations each, with a constant value of replace chance. Then, 30 experiments were run with the same configuration, for 1000 generations each, but now with a FIS controlling the replace chance according to the Stagnation Index explained in Subsection III-D. The error is recorded for each of the 1000 generations, and after one of the experiments is finished, an average of the error for the 1000 generations is calculated. This process is repeated for each of the 60 experiments. Table I shows the averages of these experiments.

The design of the Fuzzy Inference System is rather simple. The input of the FIS is the Stagnation Index, and the output is the new replace chance for the GP system in that particular iteration or generation. The input Membership Functions are depicted in Figure 6 and the output Membership Functions are depicted in Figure 7. The design of these Membership Functions were obtained according to the subjective opinion of the authors, and by trial and error. The rule set is presented below:

1) if SI is Low then RC is Low
2) if SI is Medium then RC is Medium

3) if SI is High then RC is High

In Figure 8, one can see how the dynamic adaptation of the replace chance helps avoid stagnation of the Proposed Method. Nevertheless, a hypothesis test was performed to formally prove this hypothesis. The hypothesis test was performed in the statistics software Stata, and the results are shown in Figure 9. One can see that the hypothesis test obtained a t-score of -3.3529, where the critical t value for a 95% confidence interval is -2.0018. As a conclusion, the null hypothesis is rejected (which is that the average of the error of the 30 averages of the experiments with dynamic adaptation of the replace chance is greater than or equal than the averages without the dynamic adaptation), and this means that the Proposed Method with dynamic adaptation of the replace chance outperforms the Proposed Method with a constant value of replace chance.

Fig. 6: Input for the Dynamic Adaptation of the Replace Chance

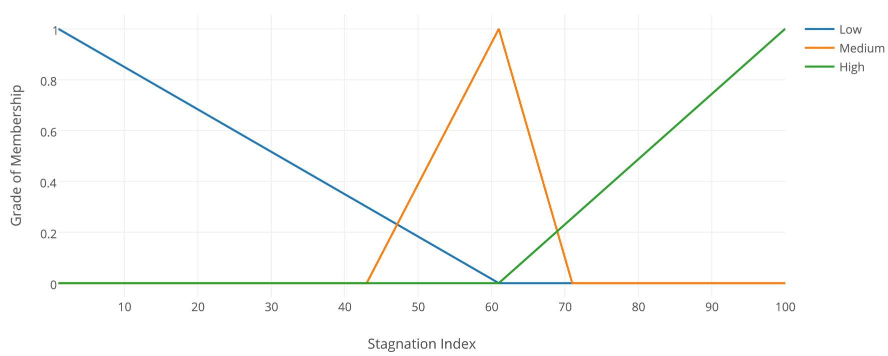


Fig. 7: Output for the Dynamic Adaptation of the Replace Chance

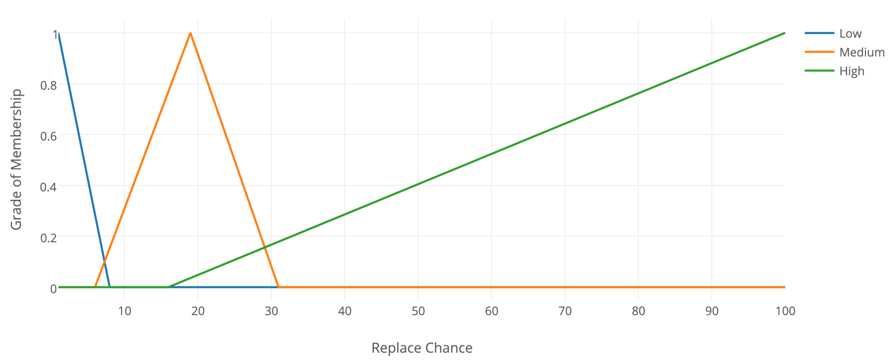


Fig. 8: Dynamic Adaptation of Replace Chance vs Constant Replace Chance

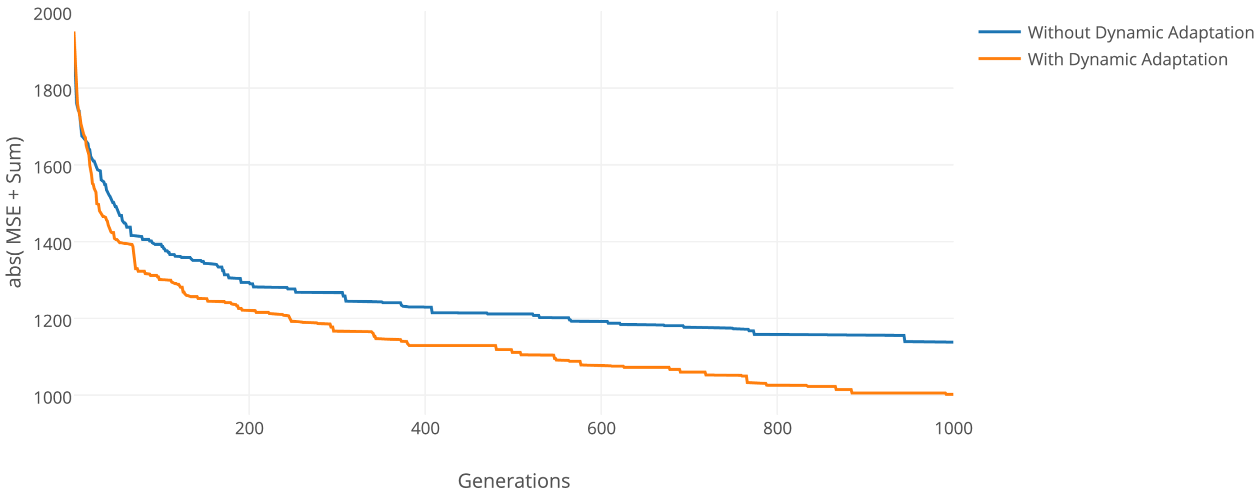


Fig. 9: Hypothesis Test for the Dynamic Adaptation of the Replace Chance

Two-sample t test with equal variances

Variable	Obs	Mean	Std. Err.	Std. Dev.	[99% Conf.	Interval]
with_fis	**30**	**1142.388**	**20.95996**	**114.8024**	**1084.614**	**1200.162**
withou~s	**30**	**1244.283**	**22.00501**	**120.5264**	**1183.629**	**1304.937**
combined	**60**	**1193.335**	**16.46105**	**127.5067**	**1149.52**	**1237.151**
diff		**-101.8954**	**30.38981**		**-182.8322**	**-20.95866**

```
    diff = mean(with_fis) - mean(without_fis)                     t =  -3.3529
Ho: diff = 0                                     degrees of freedom =       58

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0007         Pr(|T| > |t|) = 0.0014          Pr(T > t) = 0.9993
```

TABLE II: Stocks selection for trial 4

Week	Stock	Return	Week	Stock	Return
1	MMM	0.58265	8	MMM	1.27858
2	HD	1.92256	9	IBM	-2.01259
3	AA	3.72861	10	INTC	-1.92661
4	CSCO	3.48494	11	PFE	0.896414
5	CSCO	0.285551	12	AA	3.81731
6	HD	0.216626	13	PFE	3.23383
7	HD	0.981194	Total		16.489065

C. Stock Selection

In all of the following experiments, the dataset provided in [20] is used, and the first Quarter (12 records) of each stock is used for training, and the second Quarter (13 records) is used for testing.

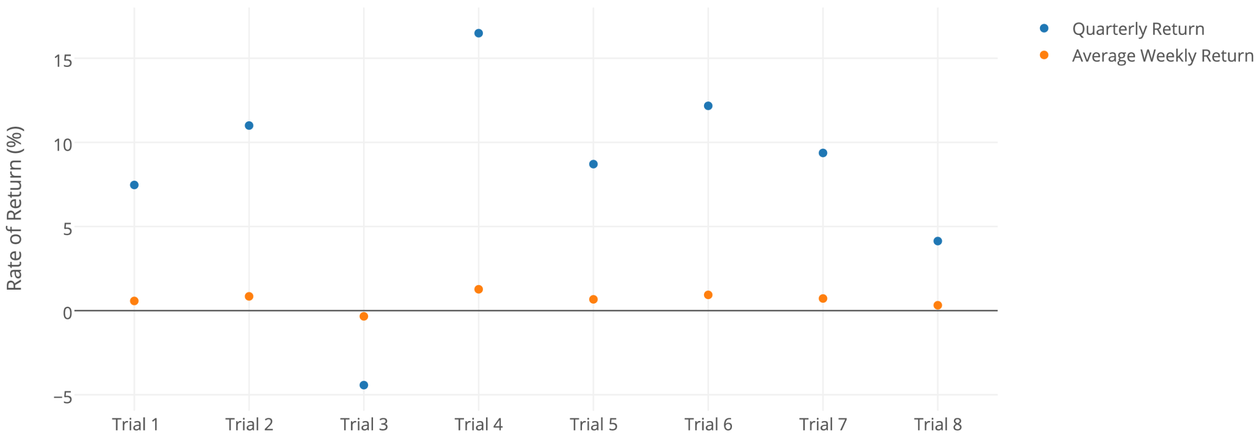
1) Comparison against Brown, et al. results: Brown, et al. [20] used their method to run a set of 8 trials, where the purpose of the experiment is to choose what the algorithm considers will be the better performing stock to buy for the next week. In the end, a single trial sums the Rate of Return from buying all the stocks that were chosen by the algorithm. In their work, they present a table showing what stocks were chosen and what was the total Rate of Return of their best trial (which in their case was the 4th trial). Then, they illustrate the results of all the trials in a similar chart to Figure 10, and they provide an average of the 8 trials.

The authors of this work don't consider this experiment to be enough to prove the efficacy of the Proposed Method. Nevertheless, we provide an experiment similar to theirs in order to compare results. Table II shows what stocks our Proposed Method chose, along with the total Rate of Return. In this case, it outperformed the results obtained by Brown, et al.

As mentioned before, Figure 10 shows a graphical representation of the results obtained from the 8 trials, where the Quarterly Return and the Average Weekly Return are shown. In this case, the average Rate of Return of the 8 trials was 8.1140%, which again outperforms the average obtained by Brown, et al., which was 7.075%.

2) Extended Experiments - Constant Configuration, Increasing Generations: The first extended experiment consists on providing the Proposed Method certain configuration (4 commu-

Fig. 10: Rate of Return of the 8 Trials



nities, 4 agents, 3 inputs, and 5 outputs) which was shown to give good results on a trial-and-error manner. This configuration was run 30 times for 30 different values of generations. The results are the average Rates of Return of the 30 trials, and are depicted in Figure 11. This graph shows that the Proposed Method tends to get under-trained with a low number of generations, begins to throw good results with a moderate number of generations, and starts to get over-trained with a higher number of generations. With a very high number of generations, the method starts acting chaotically. It is noteworthy to mention that the Proposed Method never achieved a negative average Rate of Return.

Fig. 11: Experiments of the Proposed Method with a Constant Configuration and Increasing Generations

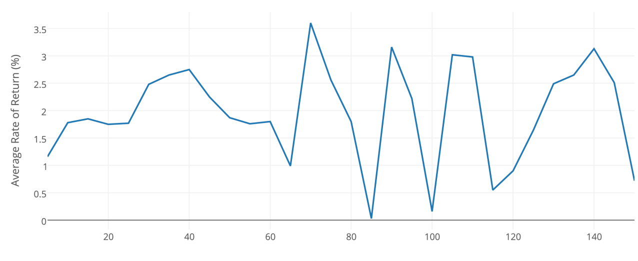


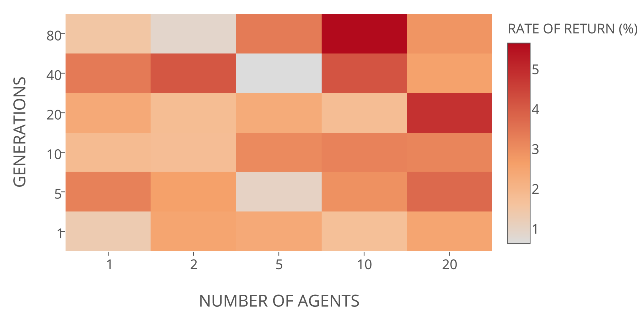
TABLE III: Proposed Method choosing what Stock to buy - Variable Configurations

	NUMBER OF AGENTS				
GENERATIONS	1	2	5	10	20
2	1.27%	2.46%	2.33%	1.68%	2.44%
5	3.22%	2.57%	0.97%	2.90%	3.70%
10	1.82%	1.75%	3.04%	3.19%	3.14%
20	2.32%	1.77%	2.28%	1.76%	4.82%
40	3.35%	4.07%	0.58%	4.13%	2.53%
80	1.48%	0.85%	3.34%	5.61%	2.81%

3) Extended Experiments - Variable Configurations: This experiment is similar to that presented by Brown, et al., with the difference that 30 trials are run instead of 8, and different configurations of the Proposed Method are used. The results are shown in Table III.

A heatmap is provided in Figure 12 for the reader in order to better appreciate the results obtained.

Fig. 12: Stock Recommendation Rate of Return Heatmap



4) Extended Experiments - Variable Configurations, Predicion of Single Stocks: In this series of experiments, instead of choosing what stock to buy, a single stock is chosen, and the algorithm had to determine if the trader had to buy or sell that particular stock. The following tables show the Rate of Return achieved, along with the number of correct predictions (if the trader should sell or buy) enclosed in parentheses. The results are shown for each of the configurations in the following Tables (IV V VI VII VIII), along with corresponding Heatmaps (13 14 15 16 17) that help the reader see what configurations worked better.

TABLE IV: American Airlines Rates of Return

	TRAINING SET SIZE			
GENERATIONS	2	4	6	8
5	-5.77% (6)	6.50% (8)	9.57% (9)	12.30% (8)
10	-5.77% (6)	6.50% (8)	7.68% (8)	12.30% (8)
15	-5.77% (6)	6.50% (8)	7.68% (8)	12.30% (8)
30	-5.77% (6)	14.95% (9)	7.68% (8)	9.13% (8)

TABLE V: CISCO Rates of Return

	TRAINING SET SIZE			
GENERATIONS	2	4	6	8
5	15.38% (8)	26.60% (10)	14.42% (10)	12.62% (10)
10	15.38% (8)	26.60% (10)	14.42% (10)	19.62% (10)
15	15.38% (8)	26.60% (10)	13.46% (9)	15.15% (9)
30	13.02% (7)	26.60% (10)	13.46% (9)	19.62% (10)

TABLE VI: IBM Rates of Return

	TRAINING SET SIZE			
GENERATIONS	2	4	6	8
5	-10.43% (5)	-12.17% (5)	1.65% (9)	-4.16% (6)
10	-5.76% (6)	-4.18% (7)	-1.85% (8)	-2.29% (6)
15	-9.98% (6)	-7.21% (5)	-6.69% (7)	-2.65% (5)
30	-4.58% (6)	-9.61% (6)	2.99% (8)	-1.10% (7)

TABLE VII: The Coca-Cola Co. Rates of Return

	TRAINING SET SIZE			
GENERATIONS	2	4	6	8
5	-8.02% (5)	1.86% (6)	2.19% (7)	-1.93% (7)
10	-9.59% (4)	-4.76% (5)	-3.51% (7)	0.58% (9)
15	-7.87% (6)	-0.81% (6)	-1.43% (7)	4.69% (8)
30	-8.02% (5)	-8.08% (5)	-4.43% (6)	-7.03% (6)

TABLE VIII: Microsoft Rates of Return

	TRAINING SET SIZE			
GENERATIONS	2	4	6	8
5	8.35% (10)	12.26% (11)	6.19% (9)	-1.52% (8)
10	11.82% (10)	12.26% (11)	1.48% (7)	6.97% (9)
15	8.35% (10)	12.26% (11)	11.74% (10)	6.97% (9)
30	8.35% (10)	12.26% (11)	11.74% (10)	2.42% (9)

Fig. 13: American Airlines Rates of Return Heatmap

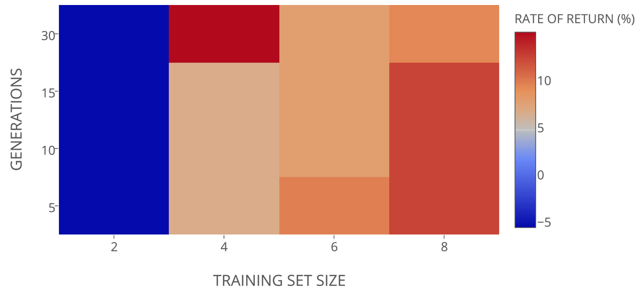


Fig. 14: CISCO Rates of Return Heatmap

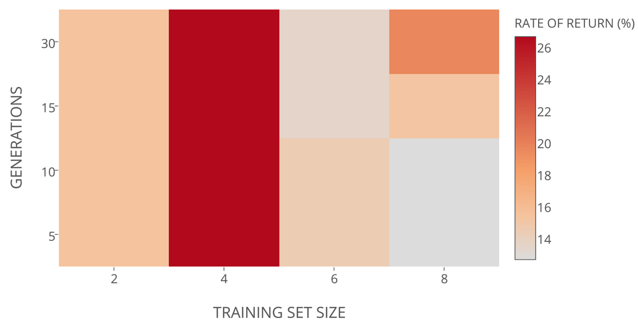


Fig. 15: IBM Rates of Return Heatmap

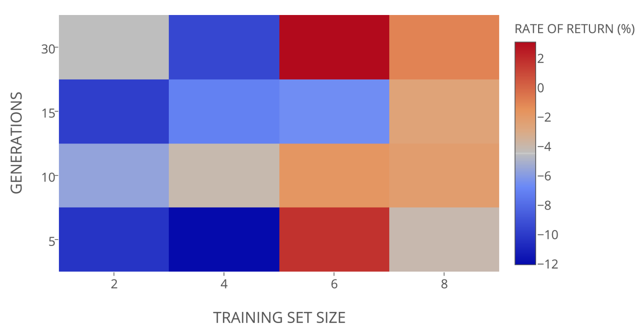


Fig. 16: The Coca-Cola Co. Rates of Return Heatmap

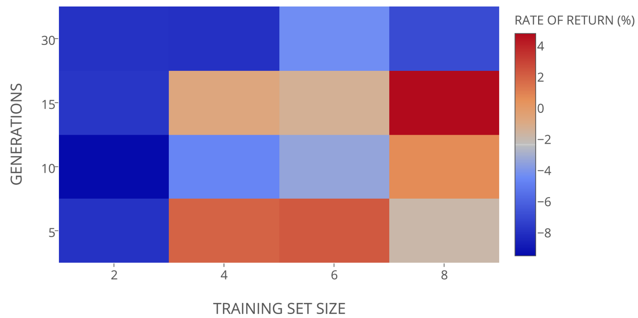


Fig. 17: Microsoft Rates of Return Heatmap

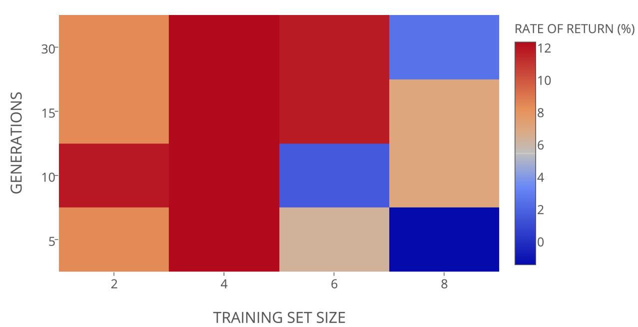


TABLE IX: Directional Strength Prediction

Generations	abs(MSE + sum of forces)
10	4430.62
100	2573.73
1000	2310.74

5) Extended Experiments - Directional Strength Prediction: The final experiment consisted on creating regression models to predict the Directional Strength of a stock. The chosen stock was American Airlines, and the configuration of the Proposed Method was 4 communities, 100 agents per community, 3 inputs (the 3 Technical Indicators), and 5 outputs. The high number of agents was chosen to demonstrate how the method behaves with such a higher number. Three experiments were performed: the first one was run for 10 generations, the second one for 100 generations, and the third one for 1000 generations. The objective of this experiment was to demonstrate that with higher generations, the method can decrease the error. The objective was achieved, as can be seen in Table IX. The curve-fitting charts can be found in Figure 18 (10 generations), Figure 19 (100 generations), and Figure 20 (1000 generations).

Fig. 18: Directional Strength Simulation - 100 Agents, 10 Generations

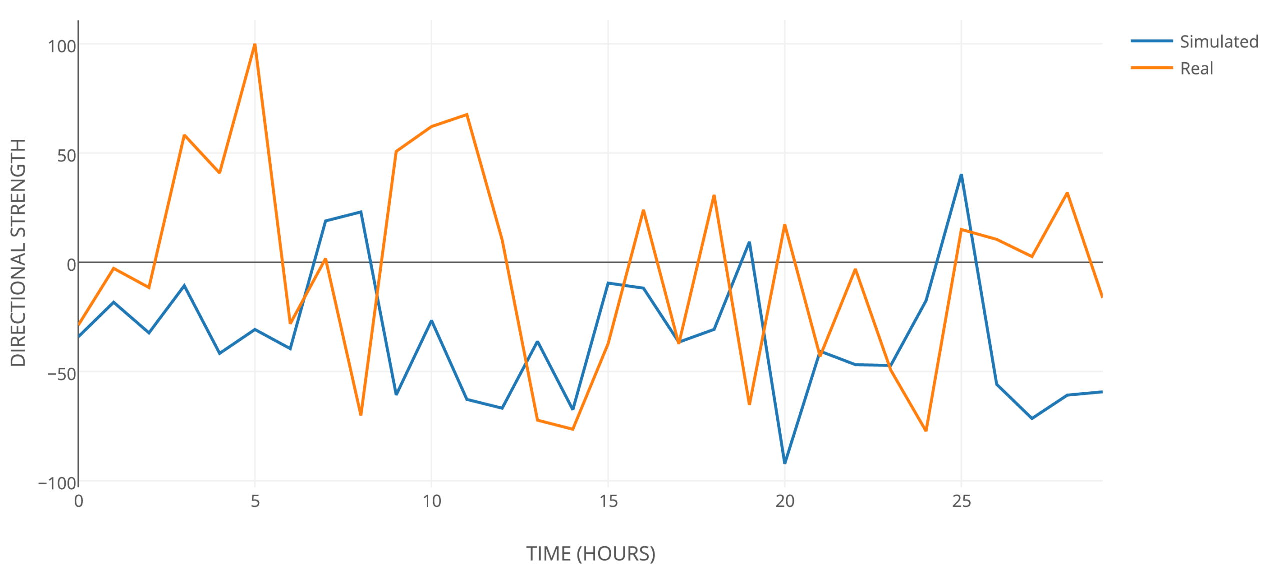


Fig. 19: Directional Strength Simulation - 100 Agents, 100 Generations

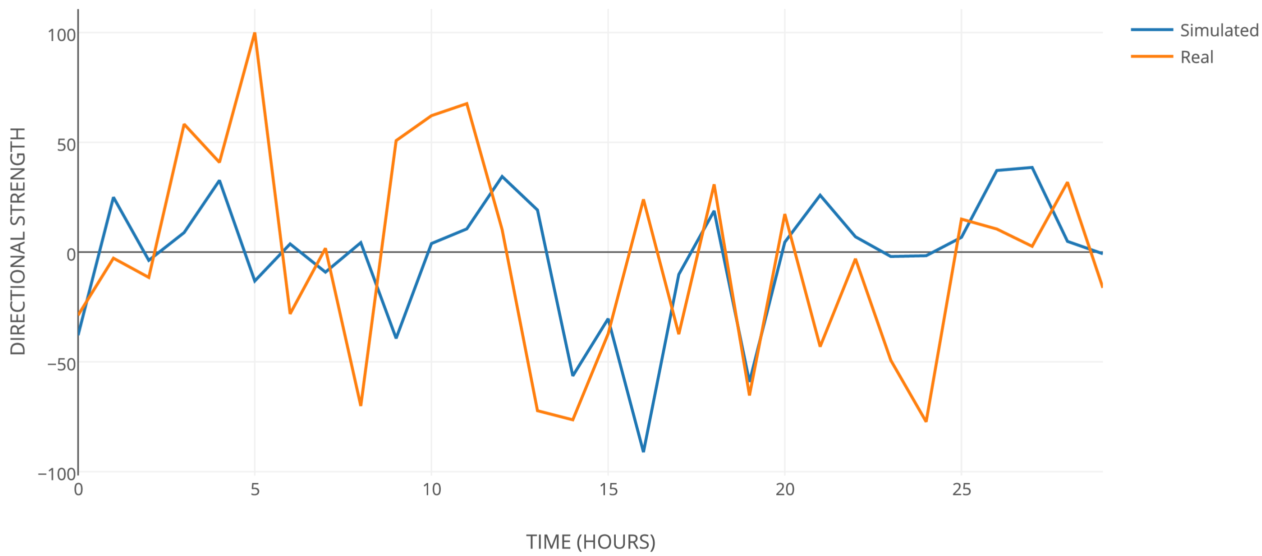
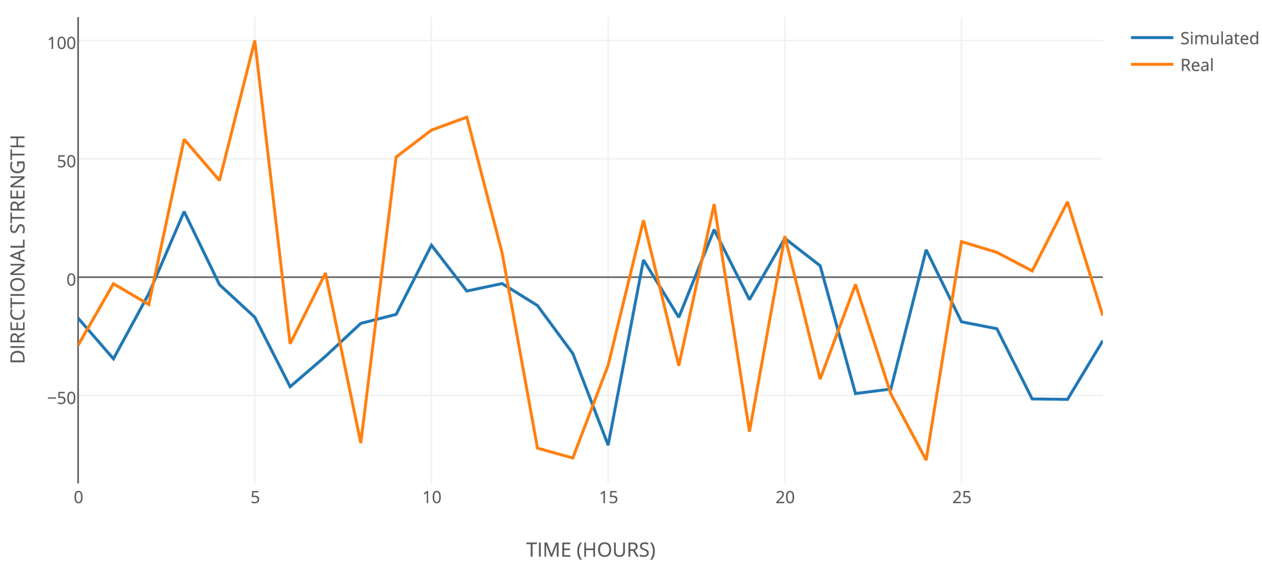


Fig. 20: Directional Strength Simulation - 100 Agents, 1000 Generations



V. CONCLUSIONS

The authors of this work performed exhaustive experiments in order to support the validity and efficacy of the Proposed Method. Nevertheless, the method should keep being validated and improved. Regarding the validation of the method, more experiments can be performed with other datasets and more comparisons with other works can be done. Regarding the improvement of the method, there are many areas of the algorithm that are known that can be improved, like applying dynamic adaptation of parameters to other parameters of the algorithm, such as the crossover, migration, and mutation chance. Also, the operators of the Genetic Programming algorithm can be modified to different sets in order to obtain better performing Membership Functions, and the ranges of the literals can be adjusted (or even dynamically adapted) to improve the overall performance of the algorithm.

In conclusion, one can see that the results are very satisfactory, and that the system can effectively be used as a Decision Support System, and as a technique to create regression models of financial time-series.

REFERENCES

[1] A. W. Lo, H. Mamaysky, and J. Wang, "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation," National bureau of economic research, Tech. Rep., 2000.

[2] S. B. Achelis, *Technical Analysis from A to Z*. McGraw Hill New York, 2001.

[3] J. W. Wilder, *New concepts in technical trading systems*. Trend Research, 1978.

[4] H. Schirding, "Stochastic oscillator," *Technical Analysis of Stocks and Commodities*, vol. 3, p. 97, 1984.

[5] N. O. Connor and M. G. Madden, "A Neural Network Approach to Predicting Stock Exchange Movements using External Factors," 2005.

[6] J. P. Burg, "A new analysis technique for time series data," *NATO advanced study institute on signal processing with emphasis on underwater acoustics*, vol. 1, 1968.

[7] L. Billard and E. Diday, "Symbolic regression analysis," in *Classification, Clustering, and Data Analysis*. Springer, 2002, pp. 281–288.

[8] M. H. Kutner, C. Nachtsheim, and J. Neter, *Applied linear regression models*. McGraw-Hill/Irwin, 2004.

[9] P. Melin, A. Mancilla, M. Lopez, and O. Mendoza, "A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting," *Applied Soft Computing*, vol. 7, no. 4, pp. 1217–1226, 2007.

[10] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

[11] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[12] N. R. Jennings, "An agent-based approach for building complex software systems," *Communications of the ACM*, vol. 44, no. 4, pp. 35–41, 2001.

[13] G. N. Gilbert, *Agent-based models*. Sage, 2008, no. 153.

[14] P. Pellizzari and A. D. Forno, "A comparison of different trading protocols in an agent-based market," *Journal of Economic Interaction and Coordination*, vol. 2, no. 1, pp. 27–43, 2007.

[15] M. Niazi and A. Hussain, "Agent-based computing from multi-agent systems to agent-based models: A visual survey," *Scientometrics*, vol. 89, no. 2, pp. 479–499, 2011.

[16] P. G. Keen, "Decision support systems: a research perspective," *Decision Support Systems: Issues and Challenges (New York: Pergamon Press, 1980)*, pp. 23–44, 1980.

[17] R. H. Sprague, "A Framework for the Development of Decision Support Systems," *MIS Quarterly*, vol. 4, no. 4, pp. 1–26, 1980. [Online]. Available: http://www.jstor.org/stable/248957

[18] J. D. Little, "Decision support systems for marketing managers," *The Journal of Marketing*, pp. 9–26, 1979.

[19] E. Tsang, P. Yung, and J. Li, "EDDIE-automation, a decision support tool for financial forecasting," *Decision Support Systems*, vol. 37, no. 4, pp. 559–565, 2004.

[20] M. S. Brown, M. J. Pelosi, and H. Dirska, "Dynamic-radius species-conserving genetic algorithm for the financial forecasting of dow jones index stocks," in *Machine Learning and Data Mining in Pattern Recognition*. Springer, 2013, pp. 27–41.

[21] R. Poli, W. B. Langdon, N. F. McPhee, and J. R. Koza, *A field guide to genetic programming*. Lulu. com, 2008.

[22] J. R. Koza, "The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems," *Dynamic, Genetic, and Chaotic Programming*, no. June, pp. 203–321, 1992. [Online]. Available: http://www.genetic-programming.com/jkpdf/soucek1992.pdf

[23] Y. Shoham and K. Leyton-brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations," *ReVision*, 2009.

[24] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2009.02.001

[25] O. Castillo and P. Melin, "Simulation and forecasting complex economic time series using neural networks and fuzzy logic," in *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, vol. 3. IEEE, 2001, pp. 1805–1810.

[26] F. M. A. Klingert and M. Meyer, "Comparing Prediction Market Mechanisms Using An Experiment-Based Multi-Agent Simulation," in *ECMS 2012 Proceedings edited by: K. G. Troitzsch M. Moehring, U. Lotzmann*. ECMS, may 2012. [Online]. Available: http://dx.doi.org/10.7148/2012-0654-0661

[27] a. Sherstov and P. Stone, "Three automated stock-trading agents: A comparative study," *Agent-Mediated Electronic Commerce VI. Theories* . . . , vol. 18, no. 2, pp. 433–41, 2005. [Online]. Available: http://www.springerlink.com/index/1n177848nj158772.pdf$\delimiter"026E30F$nhttp://www.ncbi.nlm.nih.gov/pubmed/8364599$\delimiter"026E30F$nhttp://ukpmc.ac.uk/abstract/MED/8364599

[28] G. Kendall and Y. Su, "The co-evolution of trading strategies in a multi-agent based simulated stock market through the integration of individual learning and social learning," *Proceedings of the 2003 Congress on Evolutionary Computation*, pp. 200–206, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.9288&rep=rep1&type=pdf

[29] R. Grothmann, “Multi-agent market modeling based on neural networks,” *Faculty of Economics, University of Bremen*, 2002. [Online]. Available: http://lsc.fie.umich.mx/~juan/Materias/Cursos/ANN/Papers/multi-agent-market-modeling.pdf

[30] E. Samanidou, E. Zschischang, D. Stauffer, and T. Lux, “Agent-based models of financial markets,” *Rep. Prog. Phys.*, vol. 70, no. 3, pp. 409–450, feb 2007. [Online]. Available: http://dx.doi.org/10.1088/0034-4885/70/3/r03

[31] P. Thrift, “Fuzzy Logic Synthesis with Genetic Algorithms,” pp. 509–513, 1991.

[32] a. Homaifar and E. McCormick, “Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms,” *Fuzzy Systems, IEEE Transactions . . .* , vol. 3, no. 2, pp. 129–139, 1995. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=388168

[33] J. Li and E. Tsang, “Improving technical analysis predictions: an application of genetic programming,” *Proceedings of The 12th International Florida AI Research Society Conference, Orlando, Florida*, pp. 108–112, 1999. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Improving+Technical+Analysis+Predictions+:+An+Application+of+Genetic+Programming#0

[34] A. L. Garcia-Almanza and E. P. K. Tsang, “Forecasting stock prices using Genetic Programming and Chance Discovery,” *12th International Conference On Computing In Economics And Finance*, p. number 489, 2006. [Online]. Available: http://ideas.repec.org/p/sce/scecfa/489.html

[35] S. Martinez-Jaramillo and E. P. K. Tsang, “An heterogeneous, endogenous and coevolutionary GP-based financial market,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 33–55, 2009.

[36] S. H. Chen and C. H. Yeh, “Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market,” *Journal of Economic Dynamics and Control*, vol. 25, no. 3-4, pp. 363–393, 2001.

[37] A. Bastian, “Identifying fuzzy models utilizing genetic programming,” *Fuzzy Sets and Systems*, vol. 113, no. 3, pp. 333–350, 2000.

[38] A. D. Ijegwa, V. O. Rebecca, F. Olusegun, and O. O. Isaac, “A Predictive Stock Market Technical Analysis Using Fuzzy Logic,” *Computer and Information Science*, vol. 7, no. 3, pp. 1–17, 2014.

[39] H. Huang, M. Pasquier, and C. Quek, “Financial market trading system with a hierarchical coevolutionary fuzzy predictive model,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 56–70, 2009.

[40] A. a. Gamil, R. S. El-fouly, and N. M. Darwish, “Stock Technical Analysis using Multi Agent and Fuzzy Logic,” *Proceedings of the World Congress on Engineering*, vol. I, p. 6, 2007.

[41] E. P. Tsang, J. Li, and J. M. Butler, “EDDIE beats the bookies,” *Softw., Pract. Exper.*, vol. 28, no. 10, pp. 1033–1043, 1998.

[42] E. P. Tsang, J. Li, S. Markose, H. Er, A. Salhi, and G. Iori, “EDDIE in financial decision making,” *Journal of Management and Economics*, vol. 4, no. 4, 2000.

[43] E. Tsang, “Forecasting – where computational intelligence meets the stock market The bad news : the efficient market hypothesis,” 2008.

[44] O. Castillo, H. Neyoy, J. Soria, P. Melin, and F. Valdez, “A new approach for dynamic fuzzy logic parameter tuning in Ant Colony Optimization and its application in fuzzy control of a mobile robot,” *Applied Soft Computing*, vol. 28, pp. 150–159, 2015.

[45] F. Valdez, P. Melin, and O. Castillo, “A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation,” *Expert Systems with Applications*, vol. 41, no. 14, pp. 6459–6466, 2014.