A Genetic Fuzzy System Based On Improved Fuzzy Functions

Asli Celikyilmaz
Department of EECS, University of California, Berkeley, CA 94720-1776, United States
Email: asli@eecs.berkeley.edu

I. Burhan Turksen
TOBB Economy and Technology University, Department of Industrial Engineering Ankara, Turkey
University of Toronto, Department of Industrial Engineering, Toronto, Canada
Email: bturksen@etu.edu.tr, turksen@mie.utoronto.ca

***Abstract*—Fuzzy inference systems based on fuzzy rule bases (FRBs) have been successfully used to model real problems. Some of the limitations exhibited by these traditional fuzzy inference systems are that there is an abundance of fuzzy operations and operators that an expert should identify. In this paper we present an alternate learning and reasoning schema, which use fuzzy functions instead of *if…then* rule base structures. The new fuzzy function approach optimized with genetic algorithms is proposed to replace the fuzzy operators and operations of FRBs and improve accuracy of the fuzzy models. The structure identification of the new approach is based on a supervised hybrid fuzzy clustering, entitled Improved Fuzzy Clustering (IFC) method, which yields improved membership values. The merit of the proposed fuzzy functions method is that the uncertain information on natural grouping of data samples, i.e., membership values, is utilized as additional predictors while structuring fuzzy functions and optimized with evolutionary methods. The comparative experiments using real manufacturing and financial datasets demonstrate that the proposed method is comparable or better in modeling systems of regression problem domains.**

***Index Terms*—fuzzy functions, genetic algorithms, fuzzy clustering**

I. INTRODUCTION

Fuzzy system modeling has been studied to deal with complex, not clearly explained and uncertain systems, in which conventional mathematical models may fail to reveal satisfactory results. In most fuzzy systems, fuzzy rule bases are used together with membership functions of fuzzy linguistic terms for input-output variables. In these models, the input and output relationships are represented with *if…then* rules. Although their capacity in approximating real systems is promising, in recent times the lack of learning capabilities have been the starting point of many researches, who introduced hybridization in the framework of soft computing e.g., neuro-fuzzy models [20] based on neural network algorithms [16], or evolutionary fuzzy systems based on genetic algorithms [28], [36]. These approaches are based on fuzzy rule bases, which should not be the only structure to build fuzzy systems. Thus, in this paper an alternative approach to fuzzy rule bases, entitled, "*Fuzzy Functions*" are used for fuzzy modeling.

One of the common property of the well-known fuzzy system models is that they are based on fuzzy rule base (FRB) structures, which require a series of fuzzy operations, e.g., aggregation of antecedents, implication, aggregation of consequents, and selection of the type of connection operators, e.g. t-norm, t-conorms, etc. Pointing out the abundance of these operators in [4],[5],[33],[34], a new concise fuzzy system design with improved fuzzy functions [4], is proposed which does not require most of the common FRB system operators and operations and simplify the structure identification and inference modules. In general terms the Fuzzy Functions approach is a fuzzy granular modeling, where one function is approximated for each cluster identified by Improved Fuzzy Clustering (IFC) algorithm. In this paper, we extend this approach.

While there has been a large number of fuzzy system modeling approaches in the past years, in this paper, we only focus on the fuzzy systems that implement fuzzy clustering methods, e.g., [4],[5][20],[21], [34], etc., to identify the hidden patterns in a given domain and then identify the local input-output relationships for each of these structures (patterns). In these fuzzy systems, the membership values of the fuzzy sets may represent different attributes such as the degree of belongingness, the degree of fire, the degree of compatibility, the cluster loadings, the weight or strength of local functions, or individual objects. For instance in Fuzzy c-Regression methods [15], the membership values are used as weights for each local function identified by the system model or as in [30] they are used as loadings that can be approximated using regression functions of input variables. Nonetheless, in the structure identification of the fuzzy functions, the membership values are used as additional dimensions in approximating hyper-surfaces of

Corresponding author : Asli Celikyilmaz, asli@eecs.berkeley.edu.

each cluster identified by any type of fuzzy clustering algorithm. In this paper we use the novel improved fuzzy clustering (IFC) method [4] to obtain enhanced the membership values to predict the relationship between independent and dependent variables in local structures. We hypothesize that the membership values obtained from improved clustering algorithms can increase the predictive power of the fuzzy models of each cluster, when used together with the original input variables to explain the behavior of the input and output variables in local models. In this sense, the resulting fuzzy functions are referred to as "***Improved Fuzzy Functions***".

Recent publications [3],[19],[35] have shown that evolutionary algorithms to optimize the rule base parameters such as the shape of fuzzy sets, the number of fuzzy rules and the type of the rule base operators mentioned above, may have major positive effect on the modeling performance. Such potential performance improvement due to hybridization method can be used in fuzzy function systems as well. Although it has been demonstrated in various publications [4],[5],[34] that the fuzzy functions approaches are concise approaches that can improve the modeling performance and can reduce the number of required fuzzy operations and operators as opposed to traditional FRB models, one of the glitches of the fuzzy function models is that the system parameters of both fuzzy clustering and fuzzy functions should be known prior to the model execution. Hence we propose a new evolutionary fuzzy system with improved fuzzy functions (EIFF) approach to optimize the system parameters with genetic algorithms. Preliminary results have shown improvements compared to the traditional fuzzy inference systems [9] on manufacturing domains. In this paper, we elaborate on the details of the proposed approach and how it differs from the previous fuzzy inference systems and its applications on different domains by using a new performance measure suitable for financial problem domains.

In what follows we shall explain briefly the differences between FRB and Fuzzy Functions structures, and give the architectural design of the proposed evolutionary system – EIFF –. Next, the results from the application of the proposed approach on real case studies will be presented in comparison to other hybrid fuzzy rule bases and a non-fuzzy soft computing method. Finally, conclusions will be drawn.

II. Fuzzy Functions

In this section, we present a brief comparison and a summary of the traditional fuzzy rule base systems [26],[32],[37] and the Fuzzy Function systems [4],[33],[34].

A. Fuzzy Rule Base (FRB) Systems

Traditional fuzzy inference system structure is based on the fuzzy *if-then* rules:

$$R_i\text{: } IF\ antecedent_i\ THEN\ consequent_i. \qquad (1)$$

In (1) each R_i, $i=1\ldots c$, represents one fuzzy rule. Based on the representation of the consequents structure, fuzzy inference system (FIS) gets the name; *Linguistic* FIS when the consequents are represented with fuzzy sets as in Zadeh [37], *Mizumoto* FIS [26] when the consequents are represented with a scalar values, *Takagi-Sugeno FRB* [32] when the consequents are represented with linear or non-linear equations of input variables. A fuzzy set is identified for each input variable, assuming these input variables are independent, viz. non-interactive. Fuzzy connectives are used to combine antecedent fuzzy sets to calculate the degree of fire of each rule, and to combine the output fuzzy sets of each rule to obtain aggregated output fuzzy set.

Among some of these challenges of these fuzzy rule base structures are the identification of the types of the antecedent and consequent membership functions, and their varying parameters, the optimum combination operators (t-norm, t-conorm, etc.), conjunction operators during aggregation of antecedents, and consequents, the type of the implication operator to capture uncertainty associated with the linguistic "AND", "OR", "IMP" for the representation of the rules, and reasoning with them, and the type of defuzzification method. Even though the performance of the fuzzy inference systems is slightly affected by the change in the type of the t-norm operators, one still needs to decide on the type of t-norm and t-conorm operators. Over the course of many years these challenges have been investigated to reduce the fuzzy operations [1] and expert intervention by building hybrid fuzzy systems using other soft computing methods such as genetic algorithms or neural networks, e.g., [3],[13],[20],[21],[35].

Next subsection briefly reviews such systems, which forms the basis of the proposed genetic fuzzy functions of this paper.

B. Fuzzy Functions and Fuzzy Functions Systems

"Fuzzy Functions" has been used by researchers to define different things. One of the many uses of the term refers to membership functions. It is impossible to classify the researchers who use "Fuzzy Functions" to denote membership functions since most of the fuzzy theory researchers use them interchangeably. There are several definitions of the fuzzy functions that are in need for comment.

The building blocks of the fuzzy set theory is proposed by Prof. Lotfi A. Zadeh [37]-[38], especially the fuzzy extensions of the classical basic notations such as logical connectives, quantifiers, inference rules, relations, arithmetic operations, etc. Hence, these constitute the initial definitions of fuzzy functions. Marinos [25] introduced the concept of the well-known conventional switching theory techniques into the design of fuzzy logic systems based on fuzzy set theory and fuzzy operations of Zadeh [37]. Marinos developed algebra for fuzzy sets, where the membership functions are interpreted as fuzzy numbers. Generally speaking, the processes with fuzzy attributes are represented with fuzzy logic functions. Hence, Marinos's paper is one of the examples of the application of fuzzy inference mechanisms onto real world engineering problems based on multi-valued fuzzy functions. Later Sasaki [29], Siy and Chen [31] and

Demirci [12] has explored and presented many different arithmetic operators on complex fuzzy functions. These research on fuzzy functions are the conceptual origin of the fuzzy functions as proposed in this paper. The following is the basic definition of fuzzy function with respect to fuzzy sets [25].

Let X and Y be two fuzzy sets and x and y be the membership grades of an "object" with respect to sets X and Y, respectively.

Definition 1: Two fuzzy sets X and Y are equal ($X=Y$) if, and only if, for every object i one has that its membership grade x_i in X, *i.e.*, $\mu(x_i)$, is equal to its membership grade y_i in Y, *i.e.*, $\mu(y_i)$.

Definition 2: A fuzzy set is a complement of another fuzzy set X and is denoted by X' if, and only if, for every object i one has that its membership grade x_i' in X' *is equal to* ($1-x_i$), where x_i is the membership grade of the object i in X.

Definition 3: A fuzzy set X is contained in another fuzzy set Y if, and only if, for every object i one has the $x_i \leq y_i$.

Definition 4: Two fuzzy sets X and Y form a union, denoted by $z=X+Y$, if, and only if, for every object i one has that $z=\max(x_i,y_i)$.

Definition 5: Two fuzzy sets X and Y form an intersection, denoted with $Z=X\cdot Y$, if, and only if, for every object i one has the $z=\min(x_i,y_i)$.

Based on the above definitions, algebra for fuzzy sets is developed analogous to the Boolean algebra of two valued logic. In the sequel, the term membership grade will be replaced by more appropriate term "fuzzy variable" of a sample fuzzy function, which can be defined as follows:

$$f(x,y)=x\cdot y' + x'\cdot y. \quad (2)$$

which implies that:

$$f(x,y)=\max[\min(x,1-y), \min(1-x,y)]. \quad (3)$$

In other words, (3) is a function defined between two fuzzy sets identified with fuzzy numbers, i.e., membership values, and the membership values of this function is determined by the (·) and (+) operators among each membership values of its objects.

As the number of variables and the number of operations increase, the arithmetic operations on latter types of fuzzy functions become more complex that an optimization method is necessary to find the best combinations. Kandel [22] has worked on the minimization of the fuzzy functions to build simplified reasoning algorithms with fuzzy logic functions. Later Ziwei [39] examines the properties of the fuzzy switching functions of n variables. It should be pointed that, these fuzzy functions are represented only via membership values. These types of fuzzy functions are used in the improved fuzzy clustering algorithm used in this paper and denoted as "*Interim Fuzzy Functions*".

The "Fuzzy Functions" have also been used to refer to the fuzzy rules in fuzzy rule base systems, particularly to refer to the Takagi-Sugeno fuzzy inference systems [32], where the consequents are the linear or non-linear combinations, or functions between the input and the output variables. In these systems, each local model, i.e., fuzzy rule, is identified with a separate function. Different approximators, i.e., fuzzy function approximators, are used to identify fuzzy functions, such as linear regression functions [32], multi-layered neural networks [20],[21] or genetic algorithms [3],[19]. The application of these types of "Fuzzy Functions" is the closest to the "Fuzzy Functions" strategies that are used in this paper.

The "Fuzzy Functions" systems [8],[33] of this paper are multi-variable crisp valued functions. The prominent feature of these functions, $f(X,\mu)$, are that they use the multi-dimensional degree of membership, μ_{ik}, of any multi-dimensional object $x_k \in X$ to the specified fuzzy cluster (i.e., partition, pattern set, etc.) *i, i=1,…max-number of fuzzy sets,* as an additional attribute. In a sense, the membership values become the predictors. This type of "Fuzzy Functions" emerged from the idea of representing each unique fuzzy rule in terms of functions and using them as additional input variables to explain the output in local models. One of the aims of formulizing this type of "Fuzzy Functions" is that they would not require most of the fuzzy operators and only require the knowledge of how fuzzy sets of the given system and the "Fuzzy Functions" are structured. Any function approximation method, such as least squares or neural networks tool can be used to find the parameters of these "Fuzzy Functions". Needless to say, empirical applications [34] of the system modeling with "Fuzzy Functions" using simple linear regression methods has shown promising results as compared to the conventional fuzzy rule base systems. Later, these functions are extended using machine learning algorithms, e.g., support vector machines [4],[5], for approximation of non-linear local models.

The structure identification of the concise Fuzzy Functions Systems is based on the Improved Fuzzy Clustering (IFC) algorithm. The learning algorithm as sketched in Figure 1 can be summarized as follows:

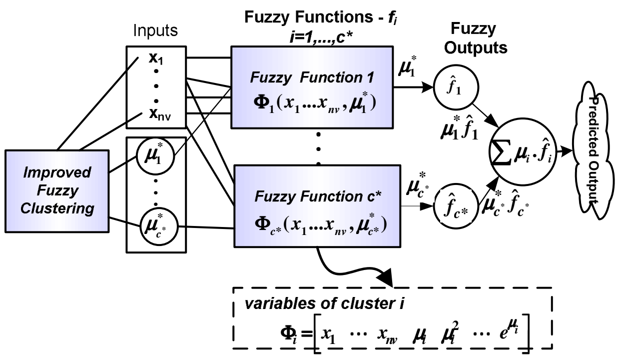


Figure 1. Structure Identification of Fuzzy Functions Approach; μ_i represents the membership values of the data points to a cluster i , $e^{\mu i}$ is the exponential power of μ_i.

- ➢ *Initialize the clustering and function approximation parameters,*
- ➢ *Cluster the given training data with IFC algorithm to*

obtain improved membership values,
- *Approximate fuzzy functions for each cluster,*
- *Obtain output values of each instance from each fuzzy function of each cluster,*
- *Apply output weighing method to obtain crisp outputs for each instance.*

The major goal in this system is to eliminate most of the aforementioned fuzzy operations of traditional fuzzy rule bases. In a simplified view such fuzzy systems work as follows:
- The domain $X \subseteq \Re^{nv}$ with nv dimensional input space is partitioned into c overlapping clusters using IFC, and each cluster is represented with cluster centers, V_i, i=1,..,c, and membership value matrix, U_i.
- To each of these regions a local fuzzy model f_i: $V_i \rightarrow \Re$ is assigned by using membership values as additional predictors to given input vector, $x \in X$. The system then identifies one fuzzy output from each fuzzy model and then weights these outputs based on the membership values of the given input vector to each cluster.

Let $(\boldsymbol{x}_k, y_k)$ denote each training data point, where $\boldsymbol{x}_k=\{x_{1,k}\ldots x_{nv,k}\}$, is the kth input vector of nv dimensions, y_k, is their output value, $\mu_{ik} \in [0,1]$ represent the membership value of kth vector to cluster $i=1\ldots c$, c be the total number of clusters, m, be the level of fuzziness parameter. The learning algorithm of type-1 Improved Fuzzy Functions approach [4] is as follows:

Step 1: IFC is a dual-structure clustering method combining FCM [2] and fuzzy c-regression algorithms [15] within one clustering schema and has the following objective function:

$$\min J_m^{IFC} = \sum_{i=1}^{c} \sum_{k=1}^{n} \mu_{ik}^m d_{ik}^2 + \sum_{i=1}^{c} \sum_{k=1}^{n} \mu_{ik}^m E_{ik} \quad . (4)$$

In (4), $d_{ik}=\|\boldsymbol{x}_k - v_i\|$, represents the distance of each $\boldsymbol{x}_k$ to each cluster center, v_i. The error $E_{ik}=(y_k - g_i(\tau_{ik}))^2$ is the squared deviation between of the approximated fuzzy models, namely the *interim* fuzzy functions, $g_i(\boldsymbol{\tau}_i)$ of cluster i and the actual output. The novelty of each interim fuzzy function $g_i(\boldsymbol{\tau}_i)$ is that the corresponding membership values and their possible transformations are the only predictors of the interim fuzzy functions, while excluding original variables. The aim is to calculate the membership values that can be candidate input variables when used to estimate the local models. An example interim fuzzy function can be formed using:

$$g_i(\boldsymbol{\tau}_i, \hat{w}_i) = \hat{w}_{0i} + \hat{w}_{1i}\mu_i + \hat{w}_{2i}(1+\exp(-\mu_{ik}^m)) \ . (5)$$

In (5), $\hat{w}_i$ represents the regression coefficients. The second term of the objective function can be minimized if optimum functions can be found. Thus, the algorithm searches for the best interim fuzzy functions, $g_i(\tau_i)$. Note that the interim input matrix τ_i is composed of improved membership values and their potential transformations as shown below.

$$\tau_i = \begin{bmatrix} \mu_{i,1} & (\mu_{i,1})^{p\neq 0} & \cdots & e^{(\mu_{i,1})^{p\neq 0}} \\ \mu_{i,2} & (\mu_{i,2})^{p\neq 0} & \cdots & e^{(\mu_{i,2})^{p\neq 0}} \\ \vdots & & \cdots & \\ \mu_{i,n} & (\mu_{i,n})^{p\neq 0} & \cdots & e^{(\mu_{i,n})^{p\neq 0}} \end{bmatrix} . \quad (6)$$

τ_i is obtained from the clustering algorithm IFC, which contains both the classical FCM clustering objective function term and the added "Interim Fuzzy Function", $h_i(\tau_i, \hat{w}_i)$, error estimation.

From the Lagrange transformation of the objective function in (4) the membership value calculation equation is formulated as follows:

$$\mu_{ik} = \left(\sum_{j=1}^{c} \left[\left(d_{ik}^2 + E_{ik}^2 \right) / \left(d_{jk}^2 + E_{jk}^2 \right) \right]^{1/(m-1)} \right)^{-1} ; \sum_{i=1}^{c} \mu_{ik} = 1 \cdot \quad (7)$$

, $i=1\ldots c$, $k=1\ldots n$. Punishing the objective function with an additional error, forces to capture the membership values that would help to improve the local models, but at the same time identify the clusters. Thus, the new membership function yields "*improved*" membership values, $\mu_{ik}^* \in U^* \subset \Re^{n \times c}$.

Step 2: One fuzzy function is approximated for each cluster to identify the input-output relations as a local model. The dataset of each cluster is comprised of the original input variables, $\boldsymbol{x}$, improved membership values, μ_{ik}^*, of particular cluster i obtained from IFC, and their user defined transformations, e.g., $((\mu_{ik}^*)^p$ $(p>1)$, e^{μ^*}, etc, see Fig 1(Bottom). This is same as mapping the nv dimensional input space, $\Re^{nv}$, of each individual cluster i onto a higher dimensional feature space $\Re^{nv+nm}$, i.e., $\boldsymbol{x} \rightarrow \Phi_i(\boldsymbol{x}, \mu_i^*)$, where nm is the total number of membership value transformations used to structure the *principle fuzzy functions,* $\hat{f}_i(\Phi_i)$, to determine the local relations of each cluster in $(nv+nm)$ space.

The interim fuzzy functions, $g_i(\boldsymbol{\tau}_i)$ are different from principle fuzzy functions $\hat{f}_i(\Phi_i)$, since $g_i(\boldsymbol{\tau}_i)$ is used to shape the membership values during IFC and only uses the membership values and their transformations as input variables.

Step 3: After the clustering algorithm, a local linear or non-linear model is approximated for each cluster identified. Any regression approximation method can be employed to identify the parameters of local functions, e.g. LSE or soft computing approaches such as neural networks or support vector machines (SVM) [14]. For instance, when LSE is used to identify the local models of a cluster i, a sample principle fuzzy function can be optimized as follows:

$$\hat{y}_i = \hat{f}_i(x, \Phi_i) = \beta_{0,i} + \beta_{1,i}\mu_i^* + \beta_{2,i}x \, . \quad (8)$$

Step 4: One crisp output is obtained by taking the average weight of the outputs from each principle function i, using the corresponding membership values as follows:

$$\hat{y} = \sum_{i=1}^{c^*} \mu_i^* \hat{f}_i(\Phi_i). \quad (9)$$

Although the fuzzy functions techniques have successful implementations which prove their modeling performance compared to FRB methods, there are several important points relating to their structure identification method. Mainly, there is an uncertainty in determining the type of membership value transformations that are used during IFC algorithm (τ_i) and for approximating the system fuzzy functions, $\hat{f}_i(\Phi_i)$, $i=1...c$. Hence, in this paper, genetic algorithms (GA) will be employed to find the optimum membership value transformations to construct τ_i and Φ_i. Since τ_i and Φ_i are two different datasets constructed during two different steps of the modeling approach, clustering and function approximating, but they include the same list of membership value transformations, we will represent them with a single parameter, Ω. Since we implement IFC method to find the membership values, one should determine the optimum degree of fuzziness, m, and number of clusters, c as well. In this paper we build evolutionary improved fuzzy functions (EIFF), which would enable to obtain the optimum values for these parameters. Next section presents the design architecture of the proposed system modeling approach.

III. Evolutionary Design of Improved Fuzzy Functions (EIFF)

EIFF is an iterative hybrid system, in which, the structure is build and parameters are constructed and tuned by genetic learning algorithm. The learning algorithm determines the size and the structure of the information granules, which are two fundamental phases of system identification. Proposed fuzzy model as depicted in Figure 2, is comprised of two fundamental phases based on cross validation:

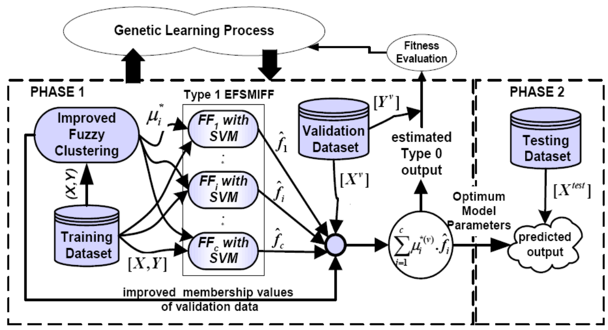


Figure 2. Evolutionary Improved Fuzzy Function (FF) Architecture. *i=1…c, v: validation data, test: testing data*

- ➢ ***Phase 1:*** Determination of the optimum parameters using genetic learning process: learning from training and fitness evaluation with validation data,
- ➢ ***Phase 2:*** Inference with testing data using the optimum model parameters.

A. Genetic Learning Process

This section presents, in sequence, basic mechanism of the GA, e.g., coding, initial population creation, fitness function, genetic operators and stopping criterion, adopted in the present work.

The structure of each chromosome encodes proposed EIFF model, which are parameters of IFC algorithm and fuzzy function structures. We used linear least squares estimation (LSE) or support vector regression (SVM) [5], to approximate fuzzy function parameters.

We utilized the hierarchical heterogeneous chromosome formulation of GA[36], where the genes of chromosomes are classified into two different types and structures : parameter genes that are real numbers, and control genes that are binary codes.

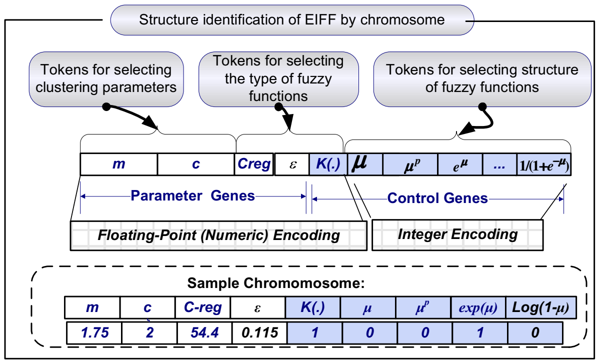


Figure 3. Hierarchical structure chromosome formulation (Bottom) A Sample chromosome structure when SVM is used.

Parameter genes are composed of in sequence: two of IFC parameters, $m\in[1.1, 3.5]$ and $c\in Z^+$ $[2,n*(1/10)]$. Depending on the type of the function approximation method, e.g., linear regression, SVM or neural networks, the next tokens after the clustering tokens denote a few additional function parameters. For instance, when SVM is used, then three of the SVM parameters, $Creg\in[2^{-3},2^7]$, $epsilon\in[0.01,0.5]$ and kernel type $K(.)$ are introduced to the chromosome. Creg is the regulation parameter to balance SVM objective function, the weight vector and the error margin, and epsilon (ε) is the error margin to flatten the decision surface. The control genes are composed of kernel type, $K(\cdot)$, and additional variables for the fuzzy function structures, Ω, i.e., interim and principle fuzzy functions structures. We used two separate kernel types for SVM formulation: linear $K(x_k,x_j)= x_k^T x_j$, and, non-linear Gaussian radial basis kernel (RBF), $K(x_k,x_j)=\exp(-\delta\|x_k-x_j\|)$, $\delta>0$. ($\therefore K(.)=0$ means linear SVM and $K(.)=1$ means RBF SVM.)

To activate the control genes, an integer 1 is assigned to represent ignition, whereas 0 is for turning it off. Hence 0 in control gene represents that the corresponding membership value transformation will no be used in the model if that chromosome is used to set the parameters. In this paper, since we used the same fuzzy function structures and parameters for each cluster, static length chromosome represent each cluster structure. The length of the fuzzy function structures, viz. the number of membership value transformations used in regression functions, is determined prior to chromosome formation.

Initial population is randomly generated. Parameter genes are formed by random numbers that can assume values between suggested intervals depicted in the previous paragraph. Fitness function is defined based on performance of the evolutionary improved fuzzy function (EIFF) using validation dataset. The performance measures that we used in this paper are the root mean square error:

$$RMSE_p = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(y_k - \hat{y}_{k,p}\right)} \qquad (10)$$

, where *p=1…population-size* and mean absolute percentage error (MAPE):

$$MAPE_p = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{y_k - \hat{y}_{k,p}}{y_k}\right|.100 \qquad (11)$$

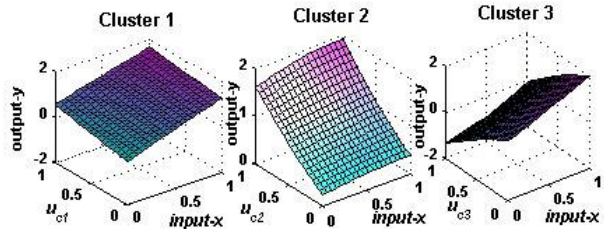
MAPE produces a measure of relative overall fit. *MAPE* is a normalized value between 0 and 1. MAPE=0 means that model output exactly matches with the observed output. In the experiments section, we also present a profitability measure for the financial dataset analysis.

Different genetic operators are utilized for parameter and control genes since they are real and binary numbers respectively. For parameter genes, we used arithmetic and simple crossover and non-uniform and uniform mutation operators. For control genes, simple crossover and shift mutation operators are utilized. Tournament selection based on elitist strategy is employed.

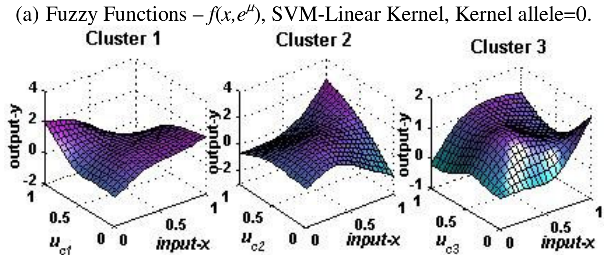
The purpose of this paper is to find the number (c) of the hidden patterns, the structure of fuzzy functions (Ω), the parameters and structure of membership functions (m, $Creg$, ε, $K(\cdot)$) so that the optimum model has accurate output in comparison to the identified system. The learning process of the proposed EIFF, as shown in Figure 2 Phase 1, is as follows:

GA initializes chromosomes to form initial population$^{g=0}$.
For each g=1… max-number-iterations,
{
chr_p : pth chromosome in the population with parameters m_p, c_p, $Creg_p$, ε_p, Kernel-type$_p$ {$K(\cdot)$}, and Ω_p .
if chr_p has not been used in the past iterations
{
– *Compute IFC with parameters from the chr_p using training data.*
– *Approximate fuzzy functions $f_i(x,\Phi_i)$ of each cluster $i=1…c_p$ using chr_p parameters.*
– *Find improved membership values of validation data and infer their output values using each $f_i(x,\Phi_i)$,*
– *Measure fitness value using the validation data.*
}
GA generates next population^{g+1} by means of crossover and mutation operations
Next generation (g=g+1)
}
End

For each cluster identified a different fuzzy decision surface is approximated based on the parameter and control genes of the chromosome structure using the membership values of the corresponding cluster as additional inputs. For instance, Figure 4 depicts the decision surfaces of each cluster of a non-linear sinusoidal toy dataset when all the alleles are kept the same but the kernel type (allele #5), which determines the non-linearity of the fuzzy functions. The presented EIFF approach searches for the best fuzzy decision surfaces based on the parameters that are represented with each chromosome.



(a) Fuzzy Functions – $f(x,e^{u})$, SVM-Linear Kernel, Kernel allele=0.



(b) Fuzzy Functions – $f(x,e^{u})$, SVM-Gaussian Kernel, Kernel allele=1.

Figure 4. The change in decision surfaces of each cluster when only the structure of the fuzzy function is changed. $(m,c,Creg,\varepsilon)$={$1.75,3,54.5,0.115$} . u_{ci} represents membership values of corresponding cluster.

B. Inference Algorithm of the EIFF

The main objective of the proposed cross validation based EIFF method is to find a crisp output for any given new testing data point (instance). Therefore, as a result of the algorithm, the type of the output is reduced from type 1 to type 0 by using the EIFF model parameters.

Inference engine is used to score the validation data during GA optimization as well as the testing dataset to evaluate the optimum model performance.

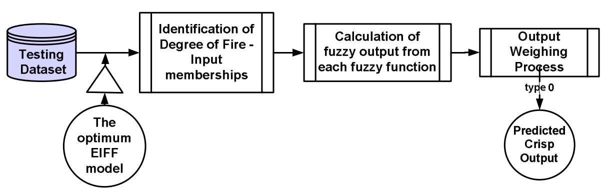


Figure 5. Inference Mechanism of the EIFF – Phase 2.

The second phase of the EIFF as shown in Figure 2, is the inference algorithm using the testing data. In the previous section, the selection of the optimum model parameters is presented by employing genetic learning with training data and fitness evaluation with verification data. Testing data, which has not been used during learning process, is used to evaluate the overall model performance. Figure 5 depicts the Inference framework.

After the optimum parameters of the EIFF model is obtained form the genetic learning process, i.e., $\langle c^*, m^*, \Omega^*, \{\hat{w}^*_{i=1,\ldots,c*}\}\rangle$, c^*:optimum number of clusters, m^*:optimum degree of fuzziness, Ω^*:optimum fuzzy function structures, and $\hat{w}^*_{i=1,\ldots,c}$:optimum interim fuzzy function parameters, firstly improved membership values of the testing data samples are captured using IFC membership function in (7). To use the membership calculation equation in (7) we need to have prior information about the output values of new data points, which is usually not known. Therefore, the following approach is processed:

We implement κ-*nearest* algorithm to calculate the membership values of a testing vector. In order use the membership value calculation equation in (7), we need to know the output values of the testing vectors, and their initial membership values that makes up the τ_i, $i=1\ldots c$, matrix. Hence we approximate the error of interim fuzzy functions such as in (5) using the optimum IFC model parameters captured from the learning phase, $\langle c^*, m^*, \Omega^*, \{\hat{w}^*_{i=1,\ldots,c*}\}\rangle$. The membership value of any lth testing vector in each cluster i is calculated as follows:

In the first step, κ training data samples that are nearest to l^{th} testing data sample are identified based on Euclidean distance measure. Using fuzzy function parameters, $\langle \Omega^*, \{\hat{w}^*_{i=1,\ldots,c*}\}\rangle\}$, improved membership values of κ-nearest training data samples are calculated using (7). As a result, an input matrix of κ number of nearest vectors for each cluster, $\boldsymbol{\tau}_i^* = [\boldsymbol{\tau}_{i1} \ldots \boldsymbol{\tau}_{i\kappa}]^T$, are obtained. Thus, using interim fuzzy function parameters, $\hat{w}_i^*$, and the matrix structure, $\boldsymbol{\tau}_i^*$, $i=1\ldots c^*$, output values of each κ-nearest training sample is calculated using $g_i(\boldsymbol{\tau}_{iq}^*, \hat{w}_i^*)$, $q=1\ldots\kappa$.

The next step is to measure the values of these κ-nearest data points, i.e., $SE_{iq} = (y^*_q - g_i(\boldsymbol{\tau}_{iq}^*, \hat{w}_i^*))^2$, $i=1\ldots c$, to be used to approximate the average SE_i for the l^{th} test data sample.

Next, error values, SE_{iq}, are normalized with weight constants, η_{rq}, which are normalized distances of κ-training samples to testing sample l. Average approximate squared error of lth testing sample in ith cluster is calculated with weighted square error, $\overline{SE}_{il}$, which is used in the new membership function to calculate the improved membership values of lth testing sample. Then using these input membership values and optimum membership value transformations of the EIFF model, a fuzzy output value of each testing vector using each fuzzy function of each cluster is calculated. Single output value for each testing vector is obtained via weighted average defuzzification as in (8).

IV. Experiments

In this section we present the experimental analysis conducted on four different real datasets, i.e., one is from manufacturing domain, three are financial stock prices datasets, using the presented EIFF methodology. We used other fuzzy inference systems, i.e., adaptive network fuzzy inference system, ANFIS [20], dynamic evolving neural-fuzzy inference system, DENFIS [21], a genetic fuzzy system (GFS) [10], and a state-of the art non-fuzzy soft-computing method, i.e., the support vector machines (SVM) [14] as benchmark methods to justify the performance of the proposed methodology. Our aim is to demonstrate that the proposed method can have comparable or even better modeling performance in terms prediction accuracy. In Table 1, we list the parameters that are used in these models.

Genetic Fuzzy System (GFS) [10] is based on a traditional Takagi-Sugeno-Kang (TSK) [32] fuzzy model. In GFS, each chromosome encodes a complete fuzzy rule set. Parameters of triangular membership functions are optimized to build first-order TSK models. Tournament selection with elitist generation replacement strategy is used. The GA algorithm terminates when the change in error is below 10e-4.

TABLE I.
THE VALUES OF THE LIST OF PARAMETERS USED IN THE EXPERIMENTS

Method	Parameters
Support vector machines for regression – SVM [14]	*Creg*∈ $[2^{-3}, 2^7]$, *epsilon*∈ *(0.0,0.5]*, two different kernel functions, i.e., linear $K(x_k,x_j)= x_k^T x_j$, or, non-linear Gaussian radial basis kernel (RBF), $K(x_k,x_j)=\exp(-\delta\lVert x_k-x_j\rVert)$, $\delta>0$.
Adaptive Network Based Fuzzy Inference System - ANFIS [20]	Hybrid method to optimize inference parameters, Gaussian input membership function shapes, Linear rule base structure (TSK).
DENFIS [21]	Takagi-Sugeno online training
Genetic Fuzzy System – GFS [10]	GA with population size: 100, mutation and crossover probabilities are equal (=1) with 250 iterations. TSK FIS is used.
Proposed Evolutionary Improved Fuzzy Functions - EIFF	*LSE and SVM method are used separately.* *Creg_bounds*=$[2^{-3}, 2^7]$, *epsilon_bounds*= *(0.0,0.5]*, *m-bounds* =[1.01,3.5], *c*=[2,10] Genetic algorithm population size: 100, mutation and crossover probabilities are equal (=1) with 250 iterations. $\Omega=\{(\mu_i^p)\ p>0; \exp(\mu_i^p); \ln((1-(\mu_i)/(\mu_i))\}$

Method: The proposed ET2FF approach is used as a decision support tool to predict continuous output variables. All applications are implemented in Matlab. At each GA optimization, mutation and crossover is applied with a population size of 100. The algorithm is executed with 250 iterations. For parameter genes, we used arithmetic and simple crossover and non-uniform and uniform mutation operators. For control genes, simple crossover and shift mutation operators are utilized. Tournament selection based on elitist generation replacement strategy is implemented, in which all parents and off-springs of one generation are sorted according to their fitness and then t individuals with the best fitness are directly passed to the next generation.

A three-way cross validation method is employed. The entire dataset is randomly separated into three parts:

training, validation and testing. Training dataset is used to identify models, and validation dataset is used to tune parameters to capture the optimum model. Testing dataset, which has not been used to learn or optimize parameters, is used to evaluate performance of models (no tuning is done on the testing dataset). Each experiment is repeated k times with different random samples for train, validation and testing datasets.

A. Desulphurization Process of Steel Production

Desulphurization refers to the pre-treatment of the hot metal. When start-sulphur is not in the acceptable range of the targeted aim-sulphur level, two reagents, reagent1 and reagent2, are simultaneously added to control desulphurization. The aim is to build a desulphurization model that can determine the right amounts of reagents to avoid unacceptable end-products, which leads to re-desulphurization. The candidate input variables are: start-sulphur level, aim-sulphur level, temperature (of hot metal), weight (of the batch), fullness of the car (in kgs. of hot metal vessel), compounds (level of 5 different compounds) that are added to help desulphurization process. The two output variables are Reagent1 and Reagent2, which enable desulphurization. The two-output system is modeled as two Multi-Input-Single-Output (MISO) models.

After several statistical feature extraction methods and stepwise regression algorithms, 6 variables are selected as candidate input variables including start-sulphur and aim-sulphur levels, the weight of the batch, and three different compounds added to the hot batch. A three-way cross-validation is applied as follows: 150 observations for training, from the remaining data, 145 observations for validation dataset to optimize the model parameters, and then from the rest of the data, 50 of them are selected for the testing dataset. The experiments were repeated with 10 random subsets of above sizes. The results from the proposed EIFF models are compared to benchmark approaches as shown in Table 2, for each reagent output.

TABLE II
DESULPHURIZATION DATA - RMSE RESULTS OF PROPOSED AND OTHER WELL KNOWN MODELS. CALCULATED ERROR MARGINS OVER 10 REPETITIONS ARE ALSO DISPLAYED AFTER ± SIGN.

Modeling Method	Testing RMSE±SDV
OUTPUT: *Reagent1*∈ [1.07,7.27]*	
SVM-Regression	0.372±0.05
ANFIS	0.434±0.08
DENFIS	0.374±0.04
GFS	0.407±0.08
Proposed EIFF	**0.339**±0.04
OUTPUT: *Reagent2*∈ [0.2,1.43]	
SVM-Regression	0.067±0.001
ANFIS	0.065±0.001
DENFIS	0.070±0.001
GFS	0.072±0.04
Proposed EIFF	**0.063**±0.000

*the output variable is scaled with $\times 10^{-2}$.
The **bold** indicates the best model

The EIFF error margin is the minimum compared to the rest of methods and this generalization capacity of EIFF can be linked to the implemented control mechanism, such that in each step of the algorithm, the learning performance is controlled with a control dataset (in this case with a validation dataset), so that the estimated decision space can simulate almost the entire population. In addition although no prior knowledge is used (except the boundaries of the parameters), the parameter setting of EIFF are captured by an evolving method and it leads to a learning method that can capture the underlying dynamics. In order to further investigate the effects of the above facts on the outcome, we conducted additional experiments as discussed below.

B. Stock Price Estimation

In this paper we used the presented EIFF approach as a decision support tool to estimate the next day stock prices of three different stock prices, i.e., a financial bank, an insurance company and a food retail company. The three-way cross validation method applied to stock price estimation datasets is slightly different from the desulphurization dataset, especially in the construction of testing dataset. Initially, stock prices of the studied period are divided into two parts. Specific to stock price estimation models, the data is not randomly divided since these are time series data and the analysis requires continuity from one to the next data vector. The data indicating the first period is used for constructing five different training and validation datasets, i.e., each representing cross validation samples. The last period, 100 trading days of each stock price, is used for testing purposes, which has not been used for learning purposes.

A new performance measure, Robust Simulated Trading Benchmark (RSTB) [7], is used specifically for stock price prediction problems, which is in need for further explanation.

Robust Simulated Trading Benchmark (RSTB) [7], [8]

In many trading systems, the main goal is to improve the profitability. It is commonly observed from the results of many financial system modeling approaches based on machine learning tools that the stock price prediction model performances of different benchmarking methods are not always significantly different from one another. This makes it difficult to identify the best model for estimation of stock prices. Furthermore, in [11] it was shown that a neural network that correctly predicted the next-day direction 85% of the time, consistently lost money. Although the system correctly predicted market direction, the prediction accuracy was low. It is in this sense that the evaluation of the trading models should not just be based on predicted directions of stock price movements. Since the aim of stock trading models is to return profit, the profitability should be the performance measure. For these reasons, here we use RSTB criterion, based on profitability of models that are used to predict the stock prices. We also used well-known performance measure, MAPE, to compare the results with the RSTB performance measure. The RSTB combines three different properties to form one performance measure;

namely the market directions, prediction accuracy and robustness of models. RSTB is driven by a conservative trading approach. The higher the RSTB, the better the profitability of the model would be.

In order to investigate the relationship between accuracy and profit of a system, a simple strategy is studied first. This strategy simply suggests that traders buy whenever the predicted price climbs above the previous day's closing price, and sell when it drops below. The buying transaction occurs only if the investor has some cash and selling transaction occurs if the investor has stocks. The 'decision' to sell, buy or hold (do nothing) for calculation of the profit is made according to the closing date price and the model's estimated stock price.

If one analyses the error of different algorithms to estimate the value of the same stock, one may realize that not all the functions can estimate the stock price as close to the actual stock price. One should measure the error of each model to evaluate their performance. The error, $\hat{e}(t)$, is measured by simply taking the absolute difference between the estimated stock price, $\hat{y}(t)$ and the actual closing price of the previous day, $y(t\text{-}1)$, $\hat{e}(t) = |y(t-1) - \hat{y}(t)|$. The average daily price change of each stock is calculated by $\nabla y(t) = \frac{1}{n}\sum_{t}^{t+n} |y(t-1) - y(t)|$. We refer to $\nabla y(t)$ the ***confidence level*** for deciding the fitness of the prediction. If the measured error, $\hat{e}(t)$, is less than $\nabla y(t)$, confidence level, more than 50% of the estimation period, $\hat{e}(t) \leq \nabla y$, then this model is considered robust for this stock. If the error is above $\nabla y(t)$, 50% or more of the times, we might as well use random guessing (50% probability). If this happens, the new benchmark totally ignores the model and this model is not suggested (In the results we simply put N/A to indicate that this model is not better than random guessing, as shown in Table IV). The main reason of measuring model accuracy is that with this benchmark, we are trying to capture if the prediction methodology is capable of effectively predicting the actual stock price in the first place. For this reason, a new **Robust Simulated Trading Benchmark (RSTB)** is used to evaluate stock prices.

In this strategy, the following assumptions are made:

- Number of Stocks in portfolio at time *t* : $\#STK_t$
- The $amount in portfolio at time *t*: $Cash\$_t$
- The profit at the end of the period is calculated as follows: *Calculated Profit at t+n=* $[(\#STK_{t+n} * \text{Closing price}_{t+n}) + Cash\$_{t+n}] - Cash\$_t$
- Each trader starts with $100 cash.
- The decision to sell, buy or do nothing (hold) is based on the predicted decision rule base structure above.
- The trader either has cash or stock. This means that when the RSTB suggests selling, the trader/investor sells all the stocks in his/her portfolio. If the RSTB suggests buying, then the trader buys stocks using all the money in his/her portfolio. Based on this the following assumption is made:
- Estimated profit is calculated by multiplying the number of stocks (#STK) with its closing value when there are stock in the portfolio or it is equal to the $Cash in hand at the end of the day.

The RSTB decision is made at any *t* day according to the following rules (*t*-1: previous day closing price);

- IF Actual_{t-1} < Predicted-Closing Price_t AND $\text{Cash}_{t-1} > 0$ AND $\hat{e}(t) \leq (1\% * \text{Actual}_{t-1})$ THEN Buy
- IF Actual_{t-1} < Predicted-Closing Price_t AND $\text{Cash}_{t-1} > 0$ AND $\hat{e}(t) > (1\% * \text{Actual}_{t-1})$ → Hold
- IF Actual_{t-1} > Predicted-Closing Price_t AND $\text{Cash}_{t-1} > 0$ →Hold
- IF Actual_{t-1} < Predicted-Closing Price_t AND $\text{Cash}_{t-1} = 0$ →Hold
- IF Actual_{t-1} > Predicted-Closing Price_t AND $\text{Cash}_{t-1} = 0$ AND $\hat{e}(t) \leq (1\% * \text{Actual}_{t-1})$ → Sell
- IF Actual_{t-1} > Predicted-Closing Price_t AND $\text{Cash}_{t-1} = 0$ AND $\hat{e}(t) > (1\% * \text{Actual}_{t-1})$ → Hold

Several studies [17],[24] examine the relationships in stock price movements using several technical indicators. These indicators such as moving averages, volume spikes, are concerned with the dynamics of the market price and volume behaviors are used to estimate future stock prices. The results of these studies show that financial analysts today are using more than 100 different technical indicators [27] to get an insight into stock price trends.

We used some of the well-known technical indicators like moving average, exponential moving average as well as some of the new indictors (*stockcharts.com/education/IndicatorAnalysis/index.html*) to build models for stock prices using the proposed EIFF on three different historical stock prices extracted from Yahoo Finance, i.e., Toronto Dominion Bank, Sun Life (SLF) Insurance, and Loblaw's (LB) retail. The datasets are converted into a multi-input single-output data mining problem, where the input variables are just the summary values of the stock prices.

Stock prices collected for around 20-22 months are divided into two parts. Data from approximately the first 15-17 months are used to train models and to optimize model parameters. The last 5 months are held-out for testing model performances. We randomly separated 200 samples for training from the first part, 140 samples for validation of the optimum model parameters again from the first part and 100 samples to test the performance of the models from the held-out part, which has not been used for training or validation purposes. Experiments were repeated with 5 random subsets of the above sizes. Model performances using MAPE and RSTB, to be presented below, are measured for the hold-out dataset of the last five months and averaged over five repetitions.

It can be observed from the MAPE values in Table III that, most of the results of different models are very close to each other, i.e., less than 1%. In this respect it is rather hard to identify the best methodology when error measures such as MAPE values are used for comparison. This may indicate that MAPE or any other error measure based on the deviation of the estimated output from the actual output may not be a dependable performance measure for financial stock price models [17]. On the other hand, the proposed RSTB, which is based on three different properties including the directions, accuracy and robustness of the predicted outcome, yields comparable results between each methodology.

TABLE III.
AVERAGE TESTING MAPE PERFORMANCE MEASURES OF MODELS OF THREE REAL STOCK PRICES BASED. THE STANDARD DEVIATIONS FROM FIVE REPETITIONS ARE SHOWN IN PARENTHESIS.

MAPE (Stdev)	Toronto Dominion	SunLife	Loblaws
ANFIS	1.82 (1.69)	3.59 (0.77)	3.86 (1.62)
DENFIS	1.42 (0.29)	0.95 (0.07)	1.23 (0.27)
SVM	0.24 (0.08)	0.82 (0.01)	**0.84 (0.03)**
GFS	1.30 (0.62)	2.09 (0.48)	1.62 (0.21)
T1IFF	**0.2 (0.03)**	**0.81 (0.01)**	0.89 (0.06)

The values in **bold** indicate optimum method of the stock price model of the dataset in the corresponding column.

It is evident from Table IV that the proposed EIFF methodology outperforms the rest of the models in 2 out of 3 stock price datasets based on RSTB profitability measure. For instance the RSTB of the proposed EIFF model of the TD stock prices is measured as $110.61, which indicates profit at the end of the 100 days based on $100 investment. Compared to the rest of the models, with the EIFF model, we can make the best profit based on the TD stock prices. The rest of the two stocks show a falling trend at the end of 100 day. For Sunlife dataset, SVM is the most profitable model. For a $100 investment, with the SVM model, we loose the least amount of money. For the Loblaw's the proposed EIFF model proves to loose the least amount of money. This indicates that the proposed method is a comparable or a better alternative financial estimation tool for estimating the next day stock prices.

TABLE IV.
AVERAGE TESTING RSTB MEASURES OF MODELS OF THREE DIFFERENT REAL STOCK DATASETS. THE STANDARD DEVIATIONS OVER FIVE REPETITIONS ARE SHOWN IN PARENTHESIS.

RSTB (Stdev)	Toronto Dominion	SunLife	Loblaws
ANFIS	$106.3(0.60)	N/A*	$92.28(2.78)
DENFIS	N/A *	$87.18(3.42)	$92.09(4.03)
GFS	$100.40(3.38)	N/A*	N/A*
SVM	$109.54(2.99)	**$90.29**(1.52)	$92.06(1.97)
Proposed EIFF	**$110.61**(2.71)	$88.57(1.47)	**$93.8**(4.31)

* All five of the cross validation models of these methods predicted $0 profit; therefore, these models are denoted as (N/A).
The values in **bold** indicate optimum (profitable) method of the stock price model of the dataset in the corresponding column.

V. RESULTS

The following generalized list of results is presented to reflect a general summary of the research experiments:

- *A new performance measure for stock price estimation models.* In estimation problems of stock prices, the main objective is to build a decision support system that can predict the next day stock prices based on the previous period's stock prices and based on direction, accuracy and profitability concepts. Input variables types, the performance measure, and the cross validation method that are used to build models for stock prediction is slightly different from the rest of the regression domains. For this study, earlier published work from the literature is used as a guide while constructing the datasets and selecting variables. A new performance measure, Robust Simulated Trading Benchmark [7],[8], is used to evaluate stock prices. The results indicate that the proposed fuzzy systems are robust in estimating stock prices in comparison to the benchmark methods.
- *Linear versus Non-Linear fuzzy function approximators.* One of the noticeable results of the experiments is that in stock price estimation models, the best fuzzy functions are optimized with linear regression functions. The optimization methods reveal that the non-linear methods such as SVM, which implement linear and non-linear kernel functions, are not the optimum models for this domain. This is due to the structure of the stock price dataset. The input variables are derived from the previous closing values of the stocks, which have linear relationship with the current closing value, the output variable. The fuzzy function models predicted that the optimum fuzzy function approximators are the LSE method. The linear SVM methods did not reveal as a good performance as the fuzzy functions methods. Simpler linear regressions such as the least squares regression can be applied into proposed fuzzy functions approaches and yet better performances can be obtained compared to the rest of the benchmark methodologies of this thesis. One of the advantages of using a simpler regression method in fuzzy functions is the time it takes to optimize the method. In particular, when the LSE is used instead of the SVM regression, the algorithm converged much faster.
- *Optimality of Genetic Algorithms.* It is well known that the genetic algorithms' optimality may differ between runs, i.e., when the number of iterations is low, the algorithm may get stuck at the local minima. To analyze the optimality, all the methodologies that utilize genetic algorithms to optimize parameters, fuzzy functions or rules are rerun for a different number of iterations. The results of corresponding algorithms were better in terms of better performance in reducing error when the number of iterations is large. When the number of iterations was less than 30, the genetic algorithm could not always find a good performance when compared to the number of the iterations that were set to between 50 and 100. More than 100 iterations did not change the results. Hence, the optimum number of iterations is found to be around 50-100. We also changed the population size between runs to observe the performance change. The default population size is set to 50 in the genetic

algorithm toolbox in MATLAB. When the population size was around 50-100, the algorithm always found the optimum model with the best performance. However, when the population size is below 50, the algorithm could not always find the optimum model. When the population size is increased to more than 100, the performance did not change. The issue with the population size in genetic algorithms is that, as its value is increased, the time it takes to find the optimum models increases also. In the experiments, it was found that the size of the populations should not exceed 100.

V. CONCLUSIONS

In this paper an Evolutionary Improved Fuzzy Functions approach is presented to be optimized with Genetic algorithms. Structurally, the novel fuzzy functions structure is different than traditional fuzzy rule base approaches. The advantage of the new concise method is that it does not require most of the computation steps of the structure identification of previous fuzzy inference systems. Consequently the experimental results indicated that the proposed approach is more robust based on cross validation error measure and has the least error compared to other hybrid fuzzy models.

With the implementation of the proposed decision support tool on the steel production process, the steel company and the consumer will benefit as a result of reduced excess use of expensive material due to ineffective models. Increasing the efficiency in massive industrial processes will have an extraordinary positive effect on the environment. In addition, financial applications of the new EIFF shows that the new method is versatile and robust tool that can be applied any regression problem domains easily.

REFERENCES

[1] R. Babuska and H.B. Verbruggen, "Constructing fuzzy models by product space clustering," *in H. Hellendoorn and D Driankow(Eds.) Fuzzy Model identification: Selected Approaches,* pp 53-90, Springer, Berlin, Germany, 1997.

[2] J.C. Bezdek, "Fuzzy Mathematics in Pattern Classification,", Ph.D. Thesis, Applied Mathematics Center, Cornell University, Ithaca, 1973.

[3] H.A. Camaro, M.G. Pires, P.A.D. Castro, "Genetic Design of Fuzzy Knowledge Bases – a study of different approaches," *23rd IEEE Intern. Conf of NAFIPS,* vol. 2, pp. 954-959, Alberta, Canada, 2004.

[4] A. Celikyilmaz, I.B. Turksen, "Enhanced Fuzzy System Models with Improved Fuzzy Clustering Algorithm," *IEEE Trans. Fuzzy Systems.,* DOI:0/1109/TFUZZ.2007.905919, 2007.

[5] A. Celikyilmaz, I.B. Turksen, "Fuzzy Function Approximation with Support Vector Machines for Fuzzy System Modeling," *Information Sciences*, vol. 177, pp. 5163-77, 2007.

[6] A. Celikyilmaz, I.B. Turksen, "A Fuzzy cluster validity index for the improved fuzzy clustering algorithm with fuzzy functions," *Pattern Recognition Letters,* vol. 29, pp. 97-108, 2008.

[7] A. Celikyilmaz, I.B. Turksen, "Discrete Interval Valued Type-2 Fuzzy Function Applications on Stock Price Estimations," *Information Sciences, submitted,* 2008.

[8] A. Celikyilmaz. *Modeling Uncertainty with Evolutionary Improved Fuzzy Functions.* Ph.D. Thesis, University of Toronto, Toronto, Canada, 2008.

[9] A. Celikyilmaz, I.Burhan Turksen, "Evolutionary Fuzzy System Models with Improved Fuzzy Functions and Its application to Industrial Process," IEEE Int. Conference on Systems, Man and Cybernetics, 7-10 Oct. 2007, pp. 541-546, 2007.

[10] O. Cordon, F. Herrera, F. Hoffman, and L. Magdalena. *Genetic Fuzzy systems – Evolutionary Tuning and Learning of Fuzzy Knowledge Bases.* Singapore, World Scientific, 2001.

[11] Deboeck, G., "Pre-processing and Evolution of Neural Nets for Trading Stocks," *Advanced Technology for Developers,* Aug. 1992.

[12] M. Demirci, Fuzzy functions and their fundamental properties, *Fuzzy Sets Syst.* vol 106, pp. 239–246, 1996.

[13] D.E. Goldberg, The Design of Competent Genetic Algorithms: Steps Toward a Computational Theory of Innovation. Kluwer Academic Publishers, Dordrecht, 2002.

[14] S. Gunn, "Support Vector Machines for Classification and Regression," ISIS Technical Report, 1998.

[15] R. Hathaway and J. Bezdek, "Switching regression model and fuzzy clustering, " *IEEE Trans. Fuzzy Syst.,* vol. 1, pp. 195-204, 1993.

[16] S. Haykin. *Neural Networks:A comprehensive foundation.* Prentice Hall PTR, NJ, USA, 2nd edition, 1998.

[17] T. Hellstrom, K. Holmstrom, "Predicting the Stock Market," *Technical Report Series* IMa-TOM-1997-07, Malardalen University, 1998.

[18] H. Ince, T., Trafalis, "Kernel Principle Component Analysis and Support Vector Machines for Stock Price Prediction", *IEEE Intern. Joint Conf. Neural Networks,* Volume 3, July 2004, pp. 2053-2058.

[19] H. Ishibuchi, T. Nakashima, T. Kuroda, "A hybrid fuzzy genetics based machine learning algorithm: Hybridization of Michigan approach and Pittsburg Approach," in *Proc.. IEEE Int. Conf. Syst., Man, Cybern.,* Oct 1999, pp. 29-301.

[20] J-S.R. Jang, "ANFIS: Adaptive Network Based Fuzzy Inference System," *IEEE Trans. On System, Man and Cybernetics*, vol. 23, May 1993, pp. 665-685.

[21] N.K. Kasabov, Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. On Fuzzy Systems*, vol. 10, no. 2, pp. 144-154, April 2002.

[22] A. Kandel, "A Note on the simplification of fuzzy switching functions," *Information Sciences*, vol 13, pp. 91-94, 1977.

[23] H.E. Lee, K.-H. Park, Z.-Z. Bien, "Iterative fuzzy clustering algorithm with supervision to construct probabilistic fuzzy rule base from numerical data," IEEE Trans. On Fuzzy Systems, vol. 11, no 5, Oct 2003, pp. 652-665.

[24] W. Leigh, R. Hightower, N. Modani, "Forecasting the New York Stock exchange composite index with past price and interest rate on condition of volume spike," *Expert Systems with Applications,* vol 28, pp-1-8, 2005.

[25] Marinos, P. N., "Fuzzy Logic and Its Application to Switching Systems," *IEEE Trans. On Computers,* vol. C-18, no. 4, 1969.

[26] M. Mizumoto, "Method of Fuzzy Inference suitable for Fuzzy Control," *J. Soc. Instrumnt Cont. Eng.* vol. 584, pp. 959-63, 1989.

[27] J. Murphy. *Technical Analysis of Financial Markets*. New York Institute of Finance, NY, 1999.
[28] L. Sanchez, I. Couso, "Advocating the Use of Imprecisely Obseved Data in Genetic Fuzzy Systems," *IEEE Trans. Fuzzy Systems,* vol. 15, no. 4, pp. 551-62, Aug. 2007.
[29] M. Sasaki, "Fuzzy Functions," *Fuzzy Sets and Systems,* vol. 55, 295-01, 1993.
[30] M. Sato-Ilic, "Dynamic fuzzy clustering using fuzzy cluster loading," *Int. Jrnl. General Systems,* vol. 35, no. 2, Apr. 2006.
[31] P. Siy, C.S. Chen, "Minimization of fuzzy functions," *IEEE Trans. Comput.*, vol. 32 (1), pp. 100-102, 1972.
[32] T. Takagi, M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. Syst., Man, Cybern., B, Cybern,* vol. 15, no. 1, pp. 116-132, 1985.
[33] I.B. Turksen, "Fuzzy Functions with LSE," *Applied Soft Computing,* vol. 8, No. 3, June 2008.
[34] I.B. Turksen, A. Celikyilmaz, "Comparison of Fuzzy Functions with Fuzzy Rules Bases," *Intrn. Jrn. Fuzzy Systems*, vol. 6, no.3, 2006, pp. 137-149.
[35] C. Wagner, and H. Hagras, "A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots," *Proc. IEEE Intern. Conf. Fuzzy Systems,* London, UK, pp.193-198, July 2007.
[36] H. Wang, S. Kwong, Y. Jin, W. Wei, K.F. Man, "Agent Based Evolutionary Approach for Interpretable rule-based knowledge extraction," *IEEE Trans. Systems, Man, Cybern. Part C*, vol. 35, no.2, May 2005, pp. 143-155.
[37] L.A. Zadeh, "Fuzzy Sets," *Inform.& Control,* vol. 8, pp.338-53, 1965.
[38] L.A. Zadeh, "The concept of linguistic variable and its application to approximate reasoning-1," *Information Sciences*, vol. 8, pp. 199-249, 1985.
[39] Ziwei, X. "On the Representation and Enumeration of Fuzzy Switching Functions," *Information and Control*, vol. 51, pp. 216-226, issue 3, 1981.

Asli Celikyilmaz received B.Sc. degree in Industrial Engineering from Istanbul Technical University, Istanbul, Turkey in 1997, M.A.Sci. and Ph.D. degrees in Industrial Engineering from University of Toronto, Canada in 2005 ad 2008, respectively.

During 2002-2004, she has worked as a Systems Architect/Research Assistant in IIC at Innovations of Foundation of University of Toronto. From 2004 until 2008, she is working as a Research Assistant in the Department of Mechanical and Industrial Engineering at University of Toronto. In 2008, she was appointed Postdoctoral Employee Position at the Electrical Engineering and Computer Science Division at the University of California, Berkeley under the supervision of Prof. Lotfi Zadeh. She has numerous publications on fuzzy system modeling, modeling uncertainty, pattern recognition, management information systems, and decision support systems. Currently she is working on intelligent question answering algorithms based on precisiated natural language processing.

Dr. Celikyilmaz is members of IEEE , IEEE Computational Intelligence Society, NAFIPS, IFSA, IEEE Women in Engineering. She has received Best Student Paper award in NAFIPS 2007 and awards from Natural Science and Engineering Research Council (NSERC) of Canada for her Ph.D. and postdoctoral studies.

I. B. Turksen received his BS and MS degrees in Industrial Engineering and his PhD degree in Systems Management and Operations Research all from the University of Pittsburgh, PA.

He is Professor Emeritus of the Faculty of Applied Science and Engineering at the University of Toronto. Since 1987, he has been the Director of the Knowledge/Intelligence Systems Laboratory at the University of Toronto. In December 2005, he was appointed as the Head of Department of Industrial Engineering at TOBB Economics and Technology University in Ankara Turkey. During the 1991–1992 academic years, he was a Visiting Research Professor at LIFE, Laboratory for International Fuzzy Engineering, and the Chair of Fuzzy Theory at Tokyo Institute of Technology.

Prof. Turksen is or was a member of the Editorial Boards of the following publications: Fuzzy Sets and Systems, Approximate Reasoning, Decision Support Systems, Information Sciences, Expert Systems and its Applications, Journal of Advanced Computational Intelligence, Information Technology Management, Transactions on Operational Research, Fuzzy Logic Reports and Letters, Encyclopaedia of Computer Science and Technology, Failures and Lessons Learned in Information Technology. He received the outstanding paper award from NAFIPS in 1986, "L.A. Zadeh Best Paper Award" from Fuzzy Theory and Technology in 1995, "Science Award" from Middle East Technical University and an "Honorary Doctorate" from Sakarya University and Azerbaijan Devlet University. He is a Foreign Member, Academy of Modern Sciences. He is a Fellow of IFSA, International Fuzzy Systems Association and IEEE, Institute of Electrical and Electronics Engineers, and WIF, World Innovation Foundation. His current research interests centre on the foundations of fuzzy sets and logics, measurement of membership functions with experts, extraction of membership functions with fuzzy clustering, fuzzy system modeling and philosophical grounding of fuzzy theory. His contributions include, in particular, Type 2 fuzzy knowledge representation and reasoning, fuzzy truth tables, fuzzy normal forms, T-formalism, which is a modified and restricted Dempster's multi-valued mapping, "Mixed Fuzzy Functions" instead of fuzzy rule bases and system modeling applications for intelligent manufacturing and processes, as well as for management decision support and intelligent control in financial and medical decision making domains.