Evolving Computational Intelligence Systems

Plamen Angelov
Digital Signal Processing Group
Dept of Communication Systems, InfoLab21
Lancaster University
Lancaster, LA1 4WA, United Kingdom
E-mail: p.angelov@lancs.ac.uk www.infolab21.lancs.ac.uk

Nikola Kasabov
Knowledge Engineering and Discovery Institute
Auckland University of Technology
Private Bag 93000
Auckland, New Zealand
E-mail: nkasabov@aut.ac.nz www.kedri.info

***Abstract*—A new paradigm of the evolving computational intelligence systems (ECIS) is introduced in a generic framework of the knowledge and data integration (KDI). This generalization of the recent advances in the development of evolving fuzzy and neuro-fuzzy models and the more analytical angle of consideration through the prism of *knowledge evolution* as opposed to the usually used *data-centred approach* marks the novelty of the present paper. ECIS constitutes a suitable paradigm for adaptive modeling of continuous dynamic processes and tracing the evolution of knowledge. The elements of evolution, such as *inheritance* and *structure development* are related to the *knowledge* and data *pattern dynamics* and are considered in the context of an individual system/model. Another novelty of this paper consists of the comparison at a conceptual level between the concept of models and knowledge captured by these models evolution and the well known paradigm of evolutionary computation. Although ECIS differs from the concept of evolutionary (genetic) computing, both paradigms heavily borrow from the same source – nature and human evolution. As the origin of knowledge, humans are the best model of an evolving intelligent system. Instead of considering the evolution of *population* of spices or genes as the evolutionary computation algorithms does the ECIS concentrate on the *evolution of a single intelligent system*. The aim is to develop the intelligence/knowledge of this system through an evolution using inheritance and modification, upgrade and reduction. This approach is also suitable for the integration of new data and existing models into new models that can be incrementally adapted to future incoming data. This powerful new concept has been recently introduced by the authors in a series of parallel works and is still under intensive development. It forms the conceptual basis for the development of the truly intelligent systems. Another specific of this paper includes bringing together the two working examples of ECIS, namely ECOS and EFS. The ideas are supported by illustrative examples (a synthetic non-linear function for the ECOS case and a benchmark problem of house price modelling from UCI repository for the case of EFS).**

I. INTRODUCTION.

The scientists represent their knowledge of processes and interrelations as mathematical models - differential equations, regression formulas [12], or as information models - expert rules, neural networks, evolutionary computation, or hybrid models [16]. As new data is continuously collected, often this data neither fits into existing models, nor is sufficient to derive a new model. This fact was previously ignored with the assumption that one can train a model using all possible variety of available data concerning all possible situations. This is very often not the case and this was one of the reasons systems that were designed in laboratory conditions failed to perform satisfactory under new conditions. It was recognized that the ability to reason and make decisions does not ensure a true intelligence on its own [2,18]. Because the environment, in which real systems operate, is constantly and often unpredictably changing a stationary system (being represented by a fixed-structure conventional, rule-based or neural-network-based model) cannot be adequate [2,15]. This applies to technological processes, transport vehicles, communication, bio-medical, robotic systems etc. The research challenge is to address the problems of life-long *learning and adaptation* of intelligent systems. Similar challenges have been already addressed successfully in the domain of linear systems modelling, identification and control as well as design and real life implementation several decades ago, which lead to so called *conventional adaptive* systems [6]. These results, however, are valid for linear systems and few extensions only (Hammerstein models, bi-linear models etc.) while intelligent systems are complex and highly non-linear by their nature. During the recent few years this challenge brought to live the concept of *evolving* systems [2,18]. An *evolving system* is able to change its structure, to grow, update and shrink when necessary [2,3,17,24,25].

In this paper we treat evolving computational intelligence systems (ECIS) through the prism of the knowledge and data integration (KDI) approach, which constitutes a novelty comparing to the usually used data-centred approach and to the interpretability/transparency studies. KDI paradigm brings together the adaptation (which is relatively well covered by parameter adaptation techniques known from the 'conventional' adaptive systems theory [6]) with the problem of generalisation and knowledge capture. The latter concept is addressed in 'conventional' modelling disciplines (including both fuzzy and linear systems) by different cross-validation techniques. 'Conventional techniques, however, assume **all** of the data to be known *a priori* or to continue to support an assumed model structure (in the case of 'conventional' adaptive system theory [6]). The problem of acquiring new data that does **<u>not</u>** support the *a priori* assumed model structure has not been addressed before the introduction of the evolving modelling concept. An ECIS, by differ from a conventional model, continuously *learn new data to integrate this data with existing models*. The incoming data may contain new variables and missing values. ECIS develop their structure and functionality continuously, always *adapting and modifying its knowledge representation*. The ECIS approach is demonstrated here through two modelling constructs that the authors have introduced recently and are continuing to develop, namely the evolving connectionist systems [14,17,18] (ECOS) and evolving fuzzy systems [2-5] (EFS).

The introduction of the ECIS paradigm is done through the analysis of the integrating existing knowledge (e.g. formulas, rules) with new data. Despite of the advances in mathematical and information sciences, there is a lack of efficient methods to extend an existing model M to accommodate new (reliable) data set D for the same problem. Examples of existing models that need to be further modified and extended to new data are numerous: differential equation models of cells and neurons [23], a regression formula to predict outcome of cancer [22], an analytical formula to evaluate renal functions [26], a logistic regression formula for evaluating the risk of cardiac events [1], a set of rules for the prediction of outcome of trauma [7], gene expression classification and prognostic models [27,29], models of gene regulatory networks [11], and many more, e.g. [6].

One can consider several different approaches to solving the problem of integrating an existing model M and new data D. If a model M was derived from data D_M, D_M and D can be integrated to form a data set D_{all} a change (evolution) in the model structure leading to M^{new} may be needed. This change will be necessary **only** if the new data are *informative enough*. Otherwise, a parameter adaptation of the previous model, M may be satisfactory (then the evolution reduces to a 'conventional' adaptation). This change is *incremental* and *recursive* ($M^{new}=M+\Delta$). Another problem that is difficult to treat and is still open is the case when new data, D contains new variables [1,9,22,26]. The model M evolution leads to a global validity in the general case (if global criterion for identification of the model is used [4]). For understanding the dynamics of the problem and for better interpretability of the model [30] a local criterion of identification can be used [4]).

This concept is illustrated by a synthetic evolving modeling of a non-linear function using EfuNN as a form of ECOS and by evolving modeling of housing prices in Boston suburbs – a well known benchmark problem from UCI repository [36].

II. THE ECOS APPROACH TO THE KDI PROBLEM

A. Evolving Connectionist Systems (ECOS)

ECOS are connectionist (artificial neural network) systems that evolve their nodes (neurons) and connections between them through supervised incremental learning from data samples represented as input-output data vectors. One of the ECOS models, the evolving fuzzy neural network EFuNN [15], is shown in a simplified version in Fig.1.

It consists of five layers. Input nodes represent input variables. Fuzzy input nodes represent the degree to which input values belong to fuzzy membership functions that are used to define concepts such as *Low* value, or *High* value for a variable [31]. Rule nodes represent cluster centres of samples in the problem space and their associated local output functions. Fuzzy output nodes represent membership degrees of the output values to predefined output membership functions. Output nodes represent output variables.

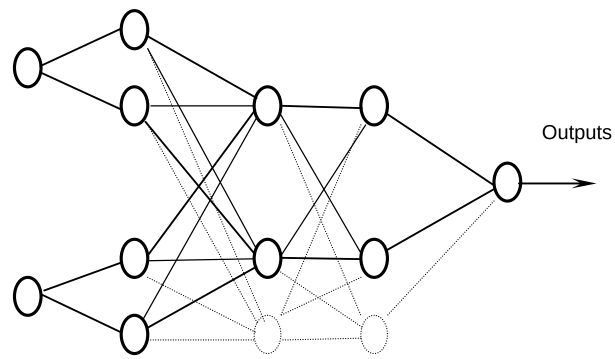

Fig. 1. A simplified version of an EFuNN

ECOS incrementally evolve rule nodes to represent cluster centres of the input data, where the first layer $\boldsymbol{W_1}$ of connection weights of these nodes represent their co-ordinates in the input space and the second layer $\boldsymbol{W_2}$ represents the local models (functions) allocated to each of the cluster (a group of similar data samples).

Data samples are allocated to rule nodes based on the similarity between the samples and the nodes calculated either in the input space (this is the case in some of the ECOS models, e.g. the dynamic neuro-fuzzy inference system DENFIS [18], and the zero instruction set computers ZISC [32]), or in both the input space and the output space (this is the case in the evolving fuzzy neural networks EfuNN [15] – Fig.1). Samples that have a distance to an existing cluster centre (rule node) N of less than a threshold R_{max} (for the EfuNN models the output vectors of these samples have to be also different from the output value associated with this cluster centre in not more than an error tolerance E) are allocated to the same cluster N_c. Samples that do not fit into existing clusters form new clusters. Cluster centres are continuously adapted to new data samples, or new cluster centres are created.

Both parameters of the ECOS (R_{max} and E) influence the convergence and stability of the new (evolved) model. They can be used for a pay-off between the model flexibility and robustness. In general, ECOS are more flexible in comparison with the EFS, which are more robust.

The distance between samples and rule nodes can be measured in different ways. The most popular measurement is the normalized Euclidean distance. In a case of missing values for some of the input variables, a partial normalized Euclidean distance can be used which means that only the existing values for the variables in a current sample $S(\mathbf{x,y})$ are used for the distance measure between this sample and an existing rule node $N\ (\mathbf{W}_{1N},\mathbf{W}_{2N})$:

$$d(S,N)=[\Sigma_{(i=1,..,n)}(x_i-W_{1N}(i))^2]/n \qquad (1)$$

for all n input variables x_i that have a defined value in the sample S and an already established connection $W_{1N}(i)$ to N.

At any time of an ECOS continuous, incremental learning from data, rules can be derived from the ECOS structure. Each rule associates a cluster area from the input variable space to a local output function applied to the data in this cluster, e.g.:

IF [data is in cluster N_{cj} defined by a cluster centre N_j, a cluster radius R_j and a number of examples N_{jex} in this cluster] THEN [the output function is f_c]

In the case of DENFIS [17] first order local fuzzy rule models are derived incrementally from data, for example:

IF [the value of x_1 is in the area defined by a triangular membership function with a centre at 0.05, left point of -0.05 and right point at 0.14) AND (the value of x_2 is in the area defined by a triangular function (0.15,0.25,0.35) respectively] THEN [the output y is calculated by: $y=0.01+0.7x_1+0.12x_2$].

In the case of EFuNNs [15] local simple fuzzy rule models are derived, for example:

IF x_1 is (Low 0.8) and x_2 is (Low 0.8) THEN y is (Low 0.8), radius $R_1=0.24$; $N_{1ex}=6$ (see first rule from Table I), where *Low*, *Medium* and *High* are fuzzy membership functions defined for the range of each of the variables x_1, x_2, and y. The number and the type of the membership functions can either be deduced from the data through learning algorithms, or it can be predefined based on human knowledge [16,18,28].

B. Addressing KDI by ECOS

The ECOS approach is illustrated (Fig. 2a) with a simple model M that represents a non-linear function y of two variables x_1 and x_2 and a new data set D_0 generated from M: $y=5.1x_1+0.345x_1^2-0.83x_1\log_{10}x_2+0.45x_2+0.57\exp(x_2^{\ 0.2})$ in the sub-space defined by $x_1\in[0;0.7]$; $x_2\in[0;0.7]$, and new data D defined by $x_1\in[0.7;1]$; $x_2\in[0.7;1]$.

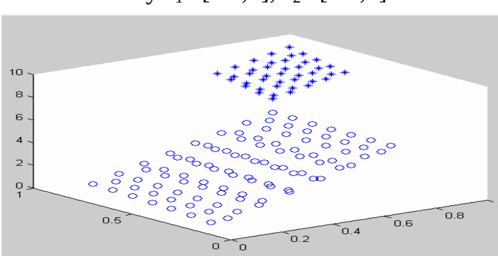

Fig. 2a. A 3D plot of data D_0 (circles) and new data D (asterisks)

Data D_{0tr} extracted from D_0 (randomly selected 56 samples from D_0) is first used to evolve a DENFIS model M_0 (parameter Rmax=0.15) and 7 rules are extracted, so the model M is transformed into an equivalent set of 7 local models. The model M_0 is further evolved on D_{tr} (randomly selected 25 samples from D) into a new model M_{new}, consisting of 9 rules allocated to 9 clusters, the first 7 representing data D_{0tr} and the last 2 - data D_{tr} (Fig.2b). While on the test data D_{0tst} both models performed equally well, M_{new} generalizes better on D_{tst} (Fig.2c). The new model M_{new} performs well on both the old and the new test data, while the old model M fails on the new test data.

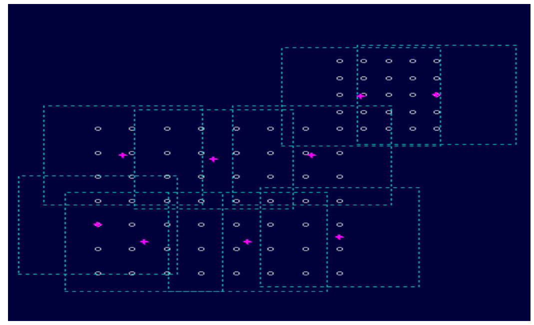

Fig. 2b. The 7 data clusters of D_0 on the left (defined by the centre denoted as "+" and a cluster area) and of the data D (the 2 upper right clusters) in the 2D input space of x_1 and x_2 input variables from Fig.2a

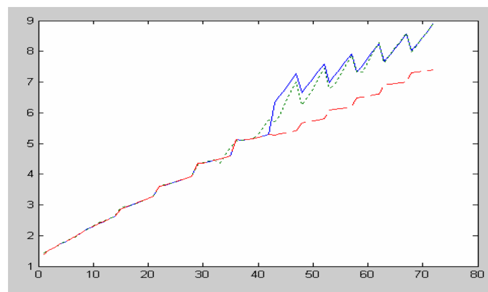

Fig. 2c. The test results of the initial model M (the dashed line) vs the new model M_{new} (the dotted line) on the test data D_{0tst} (the first 42 samples) and on the new test data D_{tst} (the last 30 samples) (the solid line).

TABLE I
LOCAL PROTOTYPE RULES EXTRACTED FROM DENFIS NEW MODEL M_{NEW}.

Rule 1: IF x_1 is (-0.05, 0.05, 0.14) and x_2 is (0.15,0.25,0.35) THEN $y=0.01+0.7x_1+0.12x_2$
Rule 2: IF x_1 is (0.02, 0.11, 0.21) and x_2 is (0.45,0.55, 0.65) THEN $y=0.03+0.67x_1+0.09x_2$
Rule 3: IF x_1 is (0.07, 0.17, 0.27) and x_2 is (0.08,0.18,0.28) THEN $y=0.01+0.71x_1+0.11x_2$
Rule 4: IF $x_1$is (0.26, 0.36, 0.46) and x_2 is (0.44,0.53,0.63) THEN $y=0.03+0.68x_1+0.07x_2$
Rule 5: IF $x_1$is (0.35, 0.45, 0.55) and x_2 is (0.08,0.18,0.28) THEN $y=0.02+0.73x_1+0.06x_2$
Rule 6: IF $x_1$is (0.52, 0.62, 0.72) and x_2 is (0.45,0.55,0.65) THEN $y=-0.21+0.95x_1+0.28x_2$
Rule 7: IF $x_1$is (0.60, 0.69,0.79) and x_2 is (0.10,0.20,0.30) THEN $y=0.01+0.75x_1+0.03x_2$
Rule 8: IF $x_1$is (0.65,0.75,0.85) and x_2 is (0.70,0.80,0.90) THEN $y=-0.22+0.75x_1+0.51x_2$
Rule 9: IF $x_1$is (0.86,0.95,1.05) and x_2 is (0.71,0.81,0.91) THEN $y=0.03+0.59x_1+0.37x_2$

An experiment was conducted with an EFuNN (error threshold E=0.15, and maximum radius R_{max}=0.25). The derived 9 local models (rules) that represent M_{new} are shown for comparison in Table II (the first 6 rules are equivalent to the model M and data D_{0tr}, and the last 3 – cover data D_{tr}).

TABLE II
LOCAL PROTOTYPE RULES EXTRACTED FROM EFUNN NEW MODEL M_{NEW}.

Rule 1: IF x_1 is (Low 0.8) and x_2 is (Low 0.8) THEN y is (Low 0.8), radius R_1=0.24; N_{1ex}= 6
Rule 2: IF x_1 is (Low 0.8) and x_2 is (Medium 0.7) THEN y is (Small 0.7), R_2=0.26, N_{2ex}= 9
Rule 3: IF x_1 is (Medium 0.7) and x_2 is (Medium 0.6) THEN y is (Medium 0.6), R_3 = 0.17,N_{3ex}=17
Rule 4: IF x_1 is (Medium 0.9) and x_2 is (Medium 0.7) THEN y is (Medium 0.9), R_4 = 0.08, N_{4ex}=10
Rule 5: IF x_1 is (Medium 0.8) and x_2 is (Low 0.6) THEN y is (Medium 0.9), R_5= 0.1, N_{5ex} = 11
Rule 6: IF x_1 is (Medium 0.5) and x_2 is (Medium 0.7) THEN y is (Medium 0.7), R_6= 0.07,N_{6ex}= 5
Rule 7: IF x1 is (High 0.6) and x2 is (High 0.7) THEN y is (High 0.6), R_7 = 0.2, N_{7ex} = 12 (new)
Rule 8: IF x1 is (High 0.8) and x2 is (Medium 0.6) THEN y is (High 0.6), R_8=0.1,N_{8ex}= 5 (new)
Rule 9: IF x1 is (High 0.8) and x2 is (High 0.8) THEN y is (High3 0.8), R_9= 0.1, N_{9ex} = 6 (new)

The models M_{new} derived from DENFIS and EFuNN are functionally equivalent, but they integrate M and D in a different way. Building alternative models of a same problem could help to better understand the problem and to choose the most appropriate model for the task.

C. Adding New Variables to ECOS

The ECOS approach above is applicable to a large-scale multidimensional data where new variables may be added at a later stage. This is possible as partial Euclidean distance between samples and cluster centres can be measured based on a different number of variables (1). If a current sample S_j contains a new variable x_{new}, having a value x_{newj} and the sample falls into an existing cluster N_c based on the common variables, this cluster centre N is updated so that it takes a coordinate value x_{newj} for the new variable x_{new}, or the new value may be calculated as weighted k-nearest values derived from k new samples allocated to the same cluster. Dealing with new variables in a new model M_{new} may help distinguish samples that have very similar input vectors but different output values and therefore are difficult to deal with in an existing model M. For example, samples S_1=[x_1=0.75, x_2=0.824, y =0.2] and S_2=[x_1=0.75, x_2=0.823, y=0.8] are easy to be learned in an new ECOS model M_{new} when a new variable x_3 is added that has, for example, values of 0.75 and 0.3 respectively for the samples S_1 and S_2.

The partial Euclidean distance (1) can be used not only to deal with missing values, but also to fill in these values in the input vectors. As every new input vector x_i is mapped into the input cluster (rule node) of the model M_{new} based on the partial Euclidean distance of the existing variable values, the missing value in x_i, for an input variable, can be substituted with the weighted average value for this variable across all data samples that fall in this cluster.

III. EVOLVING FUZZY SYSTEMS (EFS)

In section II the ECIS approach was illustrated through the ECOS paradigm, where the rules/clusters/neurons formation is more relaxed/flexible and regulated by one or two thresholds (R_{max} and E). In this section a more robust and conservative approach that avoids outliers (and thus part of the noise) in a natural way will be presented. It has only one algorithm-dependent parameter (cluster radii, r) and the decision to form a new rule/cluster/neuron relates to ***all*** previously seen data instead of just existing centres. In this sense it makes use of ***accumulated*** proximity measure concentrated in the so-called sample *potential*. For more details, please see [4] and [5].

A. EFS approach to KDI

EFS approach [2] is a "*direct evolution*" approach, which differs from the other approaches for modelling dynamic systems with flexible structure [14-20,23,24] by the mechanism of model structure evolution. In EFS the new data, D are collected incrementally on-line [2-5,25,30,33] sample by sample. The changes to the existing model, M are according to the following principles:

a) inherit the previous model structure (M_{new} =M) **iff** the new data, D do not bring *valuable* new information;

b) *gradually upgrade* the previous model structure (M_{new} =$M+\Delta$), **iff** the new data, D bring *valuable* new information;

c) *gradually modify* the previous model structure (M_{new} =$M-\Delta$), **iff** the new information the data D brings, combined with the information the previous data, D_M leads to information redundancy; the model structure modification take place as rule/cluster/neuron removal or replacement by a new one formed around the new data sample [2-5];

The information value of the new data, D is measured by the informative potential, P [2-4,34], which is inversely proportional to the accumulated spatial distance. It can also be measured by the average accumulated Euclidean distance called *scatter* [5]. The EFS approach has following features:

- it is ***evolutionary*** in the sense that it *inherits* previous model structure, M, the changes are *gradual*, and the model, M *develops* it structure from data, D_M and D;
- it extracts the ***accumulated*** information from the data, D_{all} due to its *fully recursive* nature and the use of accumulated spatial proximity instead of just the rules/neurons focal points/cluster centres;
- it is very ***restrictive, robust*** and naturally ***excludes outliers***, because only data samples with high enough potential form new rules/neurons/clusters;

B. Evolving Fuzzy Systems

The EFS approach can be interpreted as a neural network with five layers as shown in [3]. It is more effective with zero and first orders Takagi-Sugeno fuzzy models [27], although it is applicable to other type of models, e.g. Mamdani is also possible [2]. The main difference between the EFS and a conventional fuzzy system described by Takagi-Sugeno model

$$Rule^i\text{: }\mathbf{IF}(x_1 \text{ is } LT_1^i)\mathbf{AND}\ldots\mathbf{AND}(x_n \text{ is } LT_n^i)\ \mathbf{THEN}(y^i = x_e^T \pi^i)$$

is that in EFS neither the number of rules ($i=1,2,\ldots,R$), neither the linguistic terms (model antecedents), LT_j^i; $j=1,2,\ldots,n$, neither the model consequent parameters, π^i are known or fixed. The data (x_1 , x_2, x_n, y) is collected incrementally. x_e^T here denotes an extended vector of inputs, which makes possible to consider affine systems [2,4]. The overall model output, y is formed by a weighted sum of local sub-models, which have produced local predictions, y^i.[2-5, 28]. The general pseudo-code of the EFS can be given as:

```
Begin EFS
  Initialize the rule-base;
  Read first data sample;
  Establish it as the first rule centre;
    DO for each new data sample
      Read next input;
      Estimate the next output;
      At next time step read the output
      Calculate recursively Pnew
      Update the P of the centres, P*
      Compare the Pnew and P*
      IF (Pnew>P*) AND (the new data is
      close to an existing centre)
      THEN (add new rule)
      ELSEIF (Pnew>P*)
      THEN (replace the nearest rule)
   END DO
   Estimate consequent parameters by RLS;
 End (EFS)
```

The details of the evolution process are considered elsewhere [2-5], http://www.etsfm.info. In the previous sub-section we gave the basic principles in the context of KDI. This will be followed by a case study, which illustrates the use of EFS.

C. A benchmark test problem- housing data

EFS has been applied to the benchmark problem of Boston housing data. This dataset concerns housing values in suburbs of Boston. It contains 458 data instances for training and 50 instances used for model validation/testing. The output that is modelled is the median value of the owner-occupied homes (in $1000). It is influenced by a number of factors. In the full dataset 12 real-valued and 1 binary features are given. It was found that several features are not very relevant (for example so called 'Charles River dummy variable' [36], which has value 1 if tract bounds river and 0 otherwise). The 5 most relevant features were found to be:

- TAX – full value property-tax rate per $10,000;
- PTRATIO – pupil-teacher ratio by town;
- RM – average number of rooms per dwelling;
- NOX – nitric oxides concentration (parts per million) – an indicator of the pollution;
- LSTAT – lower status of the population

The only parameter that EFS requires is the vector value of the cluster radii which was considered to be $r=0.1$. The other parameter that needs to be specified is the initialization parameter for the RLS algorithm, which was set up to $\Omega = 100$ [4]. A fuzzy model started with the first data sample from the data set established as the centre/focal point of the first rule/first neuron/cluster. Then the price for the second case was modelled and predicted based on this rule alone. The error was recorded and the true price for the second case was read and so on until the 458th case. The data vectors at samples 3, 6, 7, 8, 9, 25, 27, 29, 35, 36, 91, 136, 137, 139, 193, 215, 339, 340, 375, 447, 448 were established as new rules/neurons, (Figure 3) because their potential was higher than that of the rules that existed at the moment of their appearance (Figure 4). The rule 17 that was formed around 215th sample was replaced by a rule formed around 220th sample. In this way 22 fuzzy rules were eventually in the rule base, while their number was different throughout the training phase. During the validation phase the samples from 459th till 508th were used with a model that has 22 fuzzy rules and parameters learned on-line using a modification of RLS during the training phase. Both the model structure and parameters were fixed in this phase.

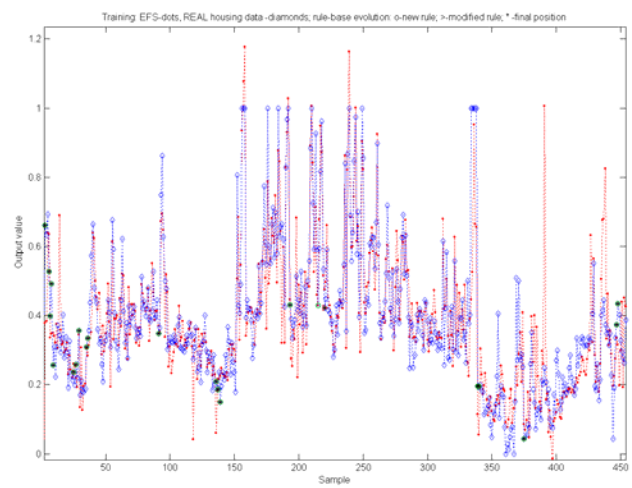

Fig.3a. Model **training** (first 458 samples) Housing price data in normalised values (dots); evolving model (EFS) (solid line); data samples that originate new rule ('o'); data samples that replace existing centres ('>'); final position of the focal point ('*').

It should be noted that such validation test is needed **to compare the results** with the *off-line* training approaches **only**. The predictions were made at each sample based on the available information (rules and parameters) so far. So, in this sense this is **not** the typical off-line training. In a normal on-line learning one does not need to stop and fix the model. Even with the model structure and parameters fixed the EFS approach shows a superior approximation quality, which is measured in this work using the standard correlation coefficient subroutine of MATLAB to calculate the correlation between the actual and the predicted output (median value of the owner-occupied homes). The results for the test of the conventional *off-line* approaches (ANFIS, genfis2, FMCLUST) and one new on-line learning (in fact a ECIS type of approach) called FLEXFIS were kindly provided by Dr. Edwin Lughofer [37].

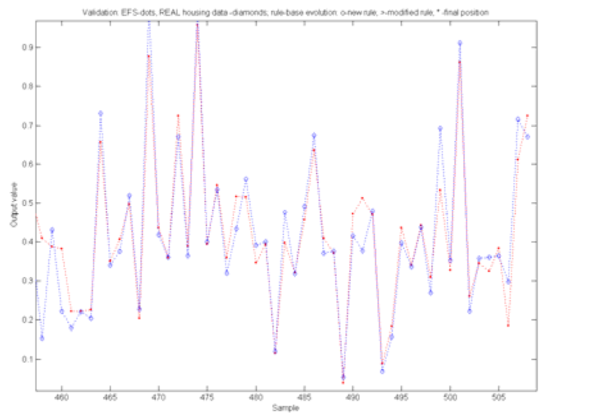

Fig.3b. Model **validation** (first 458 samples) Housing price data in normalised values (dots); evolving model (EFS) (solid line); data samples that originate new rule ('o'); data samples that replace existing centres ('>'); final position of the focal point ('*')

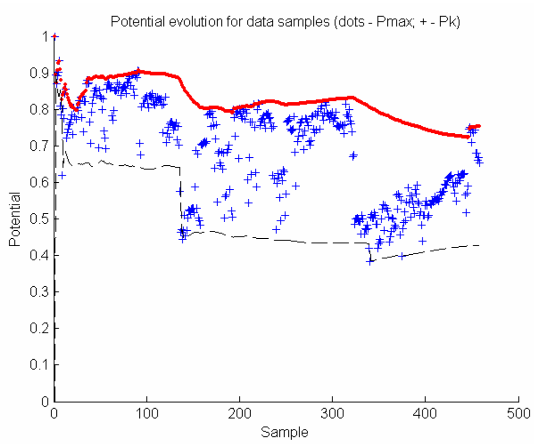

Fig.4 Potential evolution with samples

The results of the use of several well known methods for fuzzy models design (ANFIS [38], genfis2 [34],[39], FMCLUST [40], and FLEXFIS [37]) are tabulated in Table III.

TABLE IIII
RESULTS OF MODELLING HOUSE PRICE DATA

Method	**Mode**	**Correl.**
ANFIS [38]	Off-line	0.805
FMCLUST [40]	Off-line	0.890
Genfis2 [34],[39]	Off-line	0.922
FLEXFIS [37]	**On-line**	0.903
EFS (this paper)	**On-line**	**0.963**

The following figures illustrate the consequents (local sub-models) parameter evolution (Fig. 5) which is driven by a modification of the recursive least squares method, which is given in more details in [4]-[5].

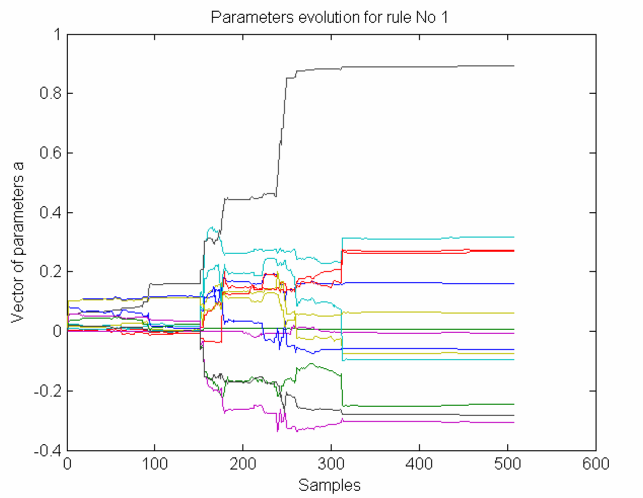

Fig.5 Local sub-model parameters evolution for rule 1

IV. DISCUSSION

This paper introduces a unified concept for *evolving computational intelligent systems* (ECIS) that develop their structure and functionality through Knowledge and Data Integration. This new concept is compared at a conceptual level with the well known evolutionary computation paradigm and case studies are considered that illustrates the application of two specific cases of ECIS: ECOS and EFS. Both ECOS and EFS were also compared on a conceptual level with EFS being more robust and stable (less sensitive to noise and outliers and more conservatively producing new clusters/rules/neurons) and having only one algorithmic parameter and ECOS being more flexible and using two algorithmic parameters for rules/neurons evolution.

ECIS is a useful paradigm for integrating new data and existing models related to the same problem into incrementally adaptive models. This is an important issue across scientific areas where new multidimensional data is being recorded continuously and new variables are being added incrementally (e.g., new genes related to a cardio-vascular disease). The approach includes ECOS [18], EFS [2-5] and their relation to KDI and to the evolution of the model structure. Further development of ECIS which are in their infancy will address the demands for more *intelligent* products in areas such as metabolomic and proteomic [10,12,19,35], gene regulatory network reengineering, clinical decision support systems [1,9,26] embedded systems with evolving intelligence, autonomous vehicles, high-tech industries, multi-media etc.

ACKNOWLEDGMENTS

The work is partially supported by the New Zealand Foundation for Research, Science and Technology, grant NERF-AUT02001. The work is also partially supported by the

Lancaster Univeristy's grant SBA 7662. ECOS software can be downloaded from www.kedri.info (books → ECOS → software). More details about ESF and the software for EFS will be available at www.etsfm.info. Dr. E. Lughofer is acknowledged for the information in regards to the tests of the housing data using off-line models and FLEXFIS.

REFERENCES

[1] Anderson, K.M., Odell, P.M., Wilson, P.W.F., Kannel, W. B. Cardiovascular disease risk profiles. *American Heart Journal* **121**, 293-298 (1991).

[2] Angelov P.P. *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems,* Springer-Verlag, Heidelberg, New York (2002).

[3] Angelov P., Filev, D. Models with Flexible Structure, International Journal on Intelligent Systems (2004).

[4] Angelov, P., Filev, D. An approach to on-line identification of evolving Takagi-Sugeno models, *IEEE Trans. on Systems, Man and Cybernetics, part B*, **34** (1) 484-498 (2004).

[5] Angelov P., D. Filev, Simpl_eTS: A Simplified method, *FUZZ-IEEE'05*, Reno, Las Vegas, USA, to appear (2005).

[6] Astroem K., B. Wittenmark, Adaptive Systems, Addison-Wessley, USA (1988).

[7] Baldi, P. & Brunak, S. *Bioinformarics - A Machine Learning Approach* (The MIT Press, 2001).

[8] Boyd, C. R., Tolson, M. A. & Copes, W. S. Evaluating Trauma Care: The TRISS Method. *The Journal of Trauma* **27**, 370-378 (1987).

[9] D'Agostino, R. B. et al. Primary and Subsequent coronary risk appraisal: New results from The Framingham Study. *American Heart Journal* **139**, 272-281 (2000).

[10] D'Haeseleer, P., Liang, S. & Somogyi, R. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **16**, 707-726 (2000).

[11] de Jong, H. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology* **9**, 67-102 (2002).

[12] Goh, L. & Kasabov, N. in *Int. Joint Conf. on Neural Networks IJCNN'2003* (IEEE Press, Portland, Oregon, USA, 2003)

[13] Harraway, J. *Introductory Statistical Methods for Biological, Health and Social Sciences* (University of Otago Press, Dunedin, 1997).

[14] Jordan, M. & Jacobs, R. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* **6**, 181-214 (1994).

[15] Kasabov, N. Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. *IEEE Trans. SMC - part B, Cybernetics* **31**, 902-918 (2001).

[16] Kasabov, N. *Foundations of neural networks, fuzzy systems and knowledge engineering* (MIT Press, 1996).

[17] Kasabov, N. & Song, Q. DENFIS: Dynamic, evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Trans. on Fuzzy Systems* **10**, 144-154 (2002).

[18] Kasabov, N. *Evolving connectionist systems - methods and applications in bioinformatics, brain study and intelligent machines* (Springer Verlag, London-New York, 2002).

[19] Kasabov, N. & Dimitrov, D. in *ICONIP'2002 - International Conference on Neuro-Information Processing* (IEEE Press, Singapore, 2002).

[20] Kasabov, N., Postma, E. & van den Herik, J. AVIS: a connectionist-based framework for integrated auditory and visual information processing. *Int. Journal on Information Sciences* **123**, 127-148 (2000).

[21] Kasabov, N., Futschik, M., Sullivan, M. & Reeve, A. in *Provisional patent* (USA, 2002).

[22] Kattan, M., Leung, D., Brennan, M. Postoperative Nomogram for 12-Year Sarcoma-Specific Death. *J.of Clinical Oncology* **20**,791-796 (2002).

[23] Kohn, K.W., Dimitrov, D.S. Mathematical Models of Cell Cycles. *Computer Modeling and Simulation of Complex Biological Syst.* (1999).

[24] Lin, F.-J., C.-H. Lin, P.-H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, *IEEE Trans. on Fuzzy Systems* **9** (5) 751-759 (2001).

[25] Liu P.X., M. Q.-H. Meng, On-line Data-Driven Fuzzy Clustering with Applications to Real-Time Robotic Tracking, *IEEE Trans. On Fuzzy Systems*, **12** (4) 516-523 (2004).

[26] Levey, A. S. et al. A more Accurate Method To Estimate Glomerular Filtration Rate from Serum Creatinine: A new Prediction Equation. *Annals of Internal Medicine* **130**, 461-470 (1999).

[27] Shipp, M. et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* **8**, 68-74 (2002).

[28] Takagi, T. & Sugeno, M. Fuzzy Identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics* **15**, 116-132 (1985).

[29] van Veer,L., Dai,J.H., Vijver, M.J.v.,He,Y.D. at. al, Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**, 530 (2002).

[30] Victor Ramos J., Dourado, A. On-line Interpretability by Fuzzy Rule-Base Simplification and reduction, Proc. EUNITE'04 Conference, Aachen, Germany (2004) .

[31] Zadeh, L. A. Fuzzy Sets. *Information and Control* **8**, 338-353 (1965).

[32] ZISC. (Silicon Recognition Ltd, www.silirec.com, 2001).

[33] Lughofer, E, Klement, P.-E. Lujan, J.M., Guardiola C., Model-based Fault Detection in Multi-Sensor Measurement Systems, IEEE Symposium on Intelligent Systems, Varna, Bulgaria, June 2004.

[34] Yager R., D. Filev ``Generation of fuzzy rules by mountain clustering," Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, Tech. Report MII-1318R (1994).

[35] Angelov P., F. von Eggeling, R. Guthke, Classification of Carcinoma Kidney Tissue Status based on the Data of Protein Expression using LS – SVM, *International Workshop on Intelligent Technologies for Gene Expression-based Individualized Medicine*, Jena, Germany, 9 May 2003.

[36] UCI Machine Learning Repository http://www.ics.uci.edu/~mlearn/MLRepository.html

[37] E. Lughofer, Data-Driven Incremental Learning of Takagi-Sugeno Fuzzy Models, PhD Dissertation, Department of Knowledge-based Mathematical Systems, Johanes Kepler University, Linz, Austria, 2005

[38] J.-S. Jang, ANFIS: Adaptive-network-based fuzzy inference systems, *IEEE Trans. Syst. Man Cybern.*, vol.23, pp. 665-685, 1993.

[39] S.Chiu, Fuzzy model identification based on cluster estimation, Journal of Intelligent and Fuzzy Systems, vol.2, no.3, 1994.

[40] R.Babuska, *Fuzzy Modeling for Control*. Boston: Kluwer Academic Publishers, 1998.