Evolving Intelligent Systems - eIS

Plamen Angelov, Nikola Kasabov

***Abstract*—The basic concept, formulation, background, and a panoramic view over the recent research results and open problems in the newly emerging area of research that is on the crossroads of computational intelligence and cybernetics is compressed in this short communication. Intelligent systems can be defined as systems that incorporate some form of *reasoning* that is typical for humans. Fuzzy Systems are well known for being able to formalize the approximate reasoning that still separates humans from machines. Artificial neural networks have proven to be a useful form of parallel processing of information that employs principles from the organization of the brain. Finally, the evolution is a phenomenon that was initially used to solve optimization problems inspired by the so called 'genetic algorithms' due to D. E. Goldberg and 'genetic programming' due to J. Koza. These types of evolutionary algorithms are mimicking the natural selection that takes place in populations of living creatures over generations. More recently, the *evolution of individual systems* within their life-span (self-organization, learning through experience, and self-developing) has attracted the attention. These systems called '*evolving*' came as a result of the research into the development of practical on-line algorithms that work in real-time and are close to the theoretically optimal, analytical solutions, suitable for non-stationary, non-linear problems of modeling, control, prediction, classification, clustering, signal processing. Due to the limited space and the specific purpose of this communication only the basic elements of the concept will be outlined. This concept represents, in fact, a higher level adaptation that concerns model structure as well as model parameters. It can also be considered as an extension of the multi-model concept known from the control theory, and of the on-line identification of fixed structure fuzzy rule-based models. It can also be considered as an extension of the learning neural networks methods in direction of on-line applications with a structure that can grow and shrink. This new concept of 'evolving intelligent systems' can also be treated in the framework of the knowledge and data integration. Evolutionary, population/generation based computation, can be applied to optimize parameters and features of an individual system, that learns incrementally from incoming data. The specific of this paper lays in the 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.* 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. A number of applications of this technique to a range of industrial and benchmark processes have been recently reported. Due to the lack of space only some of them will be mentioned primarily with illustrative purpose.**

Plamen Angelov is with the Intelligent Systems Research Laboratory, Department of Communication Systems, InfoLab21, South Drive, Lancaster University, Lancaster, LA1 4WA, UK; e-mail: p.angelov@lancaster.ac.uk).

Nikola Kasabov is with Knowledge Engineering and Discovery Institute, Auckland University of Technology, Private Bag 93000, Auckland, New Zealand. e-mail: nkasabov@aut.ac.nz

I. Introduction

It is widely accepted that systems that are capable of decision making and reasoning, that posses knowledge, are regarded as 'intelligent' [1]. Until 1970s it was assumed that this can be achieved by building so called 'expert' systems [2]. Currently, it is recognized that the techniques that contribute to increase of the 'machine intelligence quotient' [3] of a system are primarily fuzzy logic (introduced in 1965 by L.A. Zadeh [4] and used widely since 1980s), artificial neural networks (more widely used after seminal publications of P. Werbos in 1980s [5]), machine learning and evolutionary algorithms (initially introduced by J. Holland [6] and further developed by D. E. Goldberg [7] and J. Koza [8]). These branches form the triad of the so called *computational intelligence*.

A specific aspect of the development of 'intelligent' systems has attracted research attention recently. It concerns the problems of adaptivity of such systems, their use in on-line mode, for real-time applications in a wide range of real tasks originating from process industries, defense, advanced technology. This led during the last few years to the formation of the area of *evolving intelligent systems* (*eIS*)

The Oxford Advance Learner's Dictionary gives the following definition for '*evolve*': '*unfold; develop; be developed, naturally and gradually*' [9, p.294]. One can compare this with the more general '*evolutionary*' [9, p.294] 'development of more complicated forms of life (plants, animals) from earlier and simpler forms', which is naturally related to the '*genetic*' [9, p.358] 'branch of biology dealing with the heredity, the ways in which characteristics are passed on from parents to offspring'.

We use further the term '*evolving*' intelligent (*e*-ntelligent) systems (eIS) in the sense of *gradual* development of the system structure (rule-base or the architecture of the neural network that represents this system) and their parameters. This new paradigm was initially introduced for neural networks [10-11] and for fuzzy rule-based systems [12-13] by the authors and can be regarded as a *higher level adaptation*. Indeed, conventional adaptive systems known from control and system theory [43] deal predominantly with parameter adaptation of linear systems. By comparison, so called evolutionary algorithms (genetic algorithms [7], genetic programming [8] etc.) mimic the evolutionary processes that take place in **populations** of individuals and use operators based on paradigms such as 'crossover', 'mutation', 'selection', 'recombination' of chromosomes as mechanisms of adaptation. The emerging *evolving* systems paradigm mimics the evolution of **individuals** in the Nature during their life-cycle, especially humans: learning from experience, inheritance, *gradual* change, knowledge generation from routine operations, and knowledge generation from the data streams. If use a trivial analogy the approach mimics the way people learn during their life – starting with an empty rule-base or a neural network which is not trained; they learn new rules (develop the links of their neural network) during their life from experience based on the data streams that their preceptors generate to their brain. The development of the rule-base or neural network structure is *gradual*, but the rules/neurons are not fixed or pre-defined. They generate new rules (neurons) each time new data does not fit into the existing model/understanding (fuzzy rule-base or neural network), but at the same time only when this new data is informative enough (is not outlier).

It is well known that both fuzzy rule-based systems and neural networks are universal function approximators [15, 16]; they are suitable for extracting interpretable knowledge therefore, they are considered to be a promising framework for designing effective and powerful prognostic, classification, and control systems.

Traditionally, machine learning of 'intelligent systems' has been addressed primarily for the batch (off-line) case and the structure of the '*intelligent*' system was assumed to be fixed or known *a priori*. Recently, the attention has been shifted towards the on-line learning [17-19]. One can group the '*intelligent*' systems learning methods into two broad categories [20]:

1) direct (single phase) learning;

this approach performs supervised learning and addresses the identification task as a non-linear optimization problem that is solved numerically [12];

2) indirect (two phases) learning;

this approach presumes initial data partitioning using unsupervised data clustering methods and parameter identification using supervised learning method such as recursive least squares (RLS) [10,11,13,20-23],.

According to the *evolving systems* paradigm, the structure of the '*e*-ntelligent' system is not fixed, it gradually *evolves* (can expand or shrink).

Computationally efficient indirect learning methods were developed recently that are applicable to so called Takagi-Sugeno type of fuzzy rule-based systems as well as to neuro-fuzzy systems [11, 13, 21-22]. It should be noted that they apply to the more general multi-input-multi-output (MIMO) case [23] as well as to the so called simplified Mamdani type of fuzzy rules. Parallel results were reported for other types of neural networks (such as radial-basis functions, RBF type [24], self-organizing maps [25], etc.).

There is a certain parallel between the system identification (a terminology used in control theory and cybernetics) [14,43] and the more general and philosophical concept of *knowledge generation* or *discovery* from the data [26]. Both refer to the non-trivial process of identifying valid and understandable/interpretable *structure* in the data. In this respect system identification is meant in this paper mostly as a model structure identification rather than the more limited and practically more often used parameter identification. One can note that the parameter identification under a fixed model structure is nothing more than an adjustment, tuning and thus has obviously limitations related first of all to the choice of the model structure. Since the data streams are often non-stationary it is logical to assume the *structure* of the data to be also dynamic, that is, to *evolve*. An *e-ntelligent system* continuously learns new data to integrate this data with the existing models. It develops its structure and functionality continuously, always adapting and modifying its knowledge representation. The *e-ntelligent system* approach is demonstrated here through two system modelling techniques that the authors have introduced recently and are continuing to develop, namely the evolving connectionist systems [11] (ECOS) and evolving fuzzy systems [12,13] (EFS).

The remainder of the paper is organized as follows. Section II summarizes the EFS approach on a conceptual level and illustrates the approach on several industrial applications. Section III outlines the ECOS approach – some of its implementations and applications. Section IV discusses the issues of incremental (on-line) optimization of parameters and features for *e-ntelligent* systems using incremental PCA, incremental LDA and GA. Section V concludes the paper.

II. Evolving Fuzzy Systems

A. Problem Formulation

The problems of modeling of non-linear non-stationary processes is a generic one that includes the problems of prediction, tracking, estimation, control, classification, and clustering as special cases. This can be illustrated with simple block-diagrams if ignore the details and concentrate on the signal transformation only. Thus, it will be sufficient to consider the system being modeled (a process, time-series, a controller, a data stream etc.) as a simple 'black-box' as presented in Fig. 1.

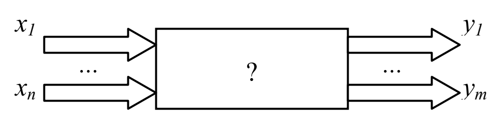

Figure 1 A generic representation of the MIMO system

Note, that this does not limit our further considerations to the so called 'black-box' type of models. As it will be seen later on, the proposed modeling technique has excellent transparency and interpretation capabilities.

In Fig.1 the following notations where used: $x = [x_1, x_2, ..., x_n]^T$ is the input vector; $y = [y_1, y_2, ..., y_m]^T$ is the *m*-dimensional output vector. For the case of clustering (unsupervised learning) the output vector is absent. In the case of control the output represents the control signal. In the case of prediction, the output represents the future value of the input vector. In case of classification, the output vector represents integer labels of respective classes, while input vector is composed by the set of the features. In the case of estimation/filtering the output represents the current value of an unobservable state of the system [27]. In so called state-space representation, the outputs represent the observations. Usually the subject areas of control, estimation, modeling, clustering, and classification are considered separately and often use different terminology for the same problems. By combining all of them we propose a generic solution that is suitable for all these types of problems and applicable in on-line and real-time. The aim is to identify a satisfactory candidate for the non-linear and non-stationary function *y=f(x)* from Fig.1.

The approaches that are widely used in practice are based on (in historical order):

i) First principles (deterministic approach, highly problem dependent, often cumbersome);
ii) Stochastic function approximation [27-28] – based on a number of assumptions that does not hold in practice, not transparent ('balck-box' type);
iii) Neural networks – usually off-line, not transparent ('balck-box' type) [24];
iv) Fuzzy rule-based models – transparent, usually off-line [29,30].

Since it was proven that both neural networks and fuzzy rule-based models are universal approximator [15,16], the last two group of models are attractive candidates to address the problem stated earlier. The main stumbling block until the end of the 20th century was the problem of their learning/design/identification. The existing state-of-art by that time was limited to off-line cases. At the same time, the classical control [43] and estimation theories [27,28] had already for quite some time developed well established algorithms and approaches addressing on-line estimation, control and prediction, albeit primarily limited to the assumption of a linear structure of the function *f(x)* and in any case to it having a *fixed* structure. The extensions for non-linear cases were limited to temporal linearization and other assumptions, such as Gaussian type of the noise [31]. The challenge in the beginning of this century was to develop and design modeling (in the broader sense as described above, including control, classification, prediction etc.) techniques that are flexible enough to cope with the real problems formulated by the industry, defense, and society.

Two of the successful extensions of the fuzzy system identification and neural networks learning problems where reported independently in the beginning of this century [10-13]. During last five years there are increasing number of publications that treat similar problems in both fuzzy systems and neural networks domains [17-25]. As recognition of the growing importance of this area the International Symposium on Evolving Fuzzy Systems brought together more than sixty high quality contributions [32].

In the remainder of this section, the evolving fuzzy rule-based systems (EFS) approach will be briefly outlined Let us consider a multi-input-multi-output (MIMO) set of fuzzy rules of the following form:

$$R^i : IF\left(x_1.is.close.to.x_1^{i*}\right)AND...AND\left(x_n.is.close.to.x_n^{i*}\right)$$
$$THEN(y^i = f^i) \qquad (1)$$

where R^i denotes the i^{th} fuzzy rule; *i=[1,N]; N* is the number of fuzzy rules; $\left(x_j.is.close.to.x_j^{i*}\right)$ denotes the i^{th} fuzzy sets of the j^{th} fuzzy rule; *j=[1,n];* x^{i*} is the focal point of the i^{th} rule antecedent; y^i is the output of the i^{th} linear sub-system.

Note that the type of the fuzzy rule depends on the type of the consequent:

- It is of so-called first order Takagi-Sugeno type [10] when the consequents are linear:

$$f^i = x_e^T \pi^i \,;\; x_e^T = [1, x^T] \qquad (2)$$

where $\pi^i = \begin{bmatrix} \alpha_{01}^i & \alpha_{02}^i & \dots & \alpha_{0m}^i \\ \alpha_{11}^i & \alpha_{12}^i & \dots & \alpha_{1m}^i \\ \dots & \dots & \dots & \dots \\ \alpha_{n1}^i & \alpha_{n2}^i & \dots & \alpha_{nm}^i \end{bmatrix}$ are the parameters of the m local linear sub-systems

- It is of zero order Takagi-Sugeno type (that can also be considered as a simplified Mamdani, type) when the consequents are singletons (crisp scalar values):

$$f^i = a^i \qquad (3)$$

where $a^i = \left[\alpha_{01}^i \quad \alpha_{02}^i \qquad \alpha_{0m}^i\right]^T$ are the parameters of the *m* local linear sub-models

Note that equations (1) and (3) describe simplified Mamdani model while the conventional Mamdani type fuzzy model assumes fuzzy consequents [18]. The overall output of the NF system, *y* is formed as a collection of loosely/fuzzily combined multiple simpler sub-systems, y^i. The degree of activation of each rule is proportional to the level of contribution of the corresponding sub-system to the overall output of the system.

$$y = \sum_{i=1}^{N} \lambda^i y^i \qquad (4)$$

where λ^i is the normalized activation level of the i^{th} rule; is the activation level of the i^{th} rule.

$$\lambda^i = \frac{\tau^i}{\sum_{j=1}^{N} \tau^j} \qquad (5)$$

This way of aggregating the partial contributions of the local models is also known as 'center of gravity' (CoG) [18]. There are other techniques to produce the overall output (to assume different structure of the model). For classification sub-problem a popular choice is so called 'winner-takes-all' mechanism:

$$\lambda^i = \begin{cases} \tau^j & j = \arg\max_{l=1}^{N}\{\tau^l\} \\ 0 & else \end{cases} \tag{5a}$$

The activation levels, τ^i can be determined as t-norms (Cartesian product) of the fuzzy sets that form the specific fuzzy rule [30]:

$$\tau^i = \prod_{j=1}^{n} \mu_j^i(x_j) \tag{6}$$

where μ_j^i is the membership value of the j^{th} input x_j, $j=[1,n]$), to the i^{th} fuzzy set ,$i=[1,N]$;

The membership function is usually of Gaussian type (this choice is justified by its generalization capabilities since it resembles normal distribution and covers the whole domain of the variables, thus avoids potential computational problems):

$$\mu_j^i = e^{-\frac{4\left\|x - x^{i*}\right\|_j^2}{\left(\sigma_j^i\right)^2}} \tag{7}$$

where $\left(\sigma_j^i\right)^2$, $i=[1,N]$ $j=[1,n]$ is the spread of the membership function, which also represents the radius of the zone of influence of the cluster/rule.

The system considered in this paper and described by equations (1)-(2) or (1)-(3) can be graphically represented as a five-layer feed-forward neural network:

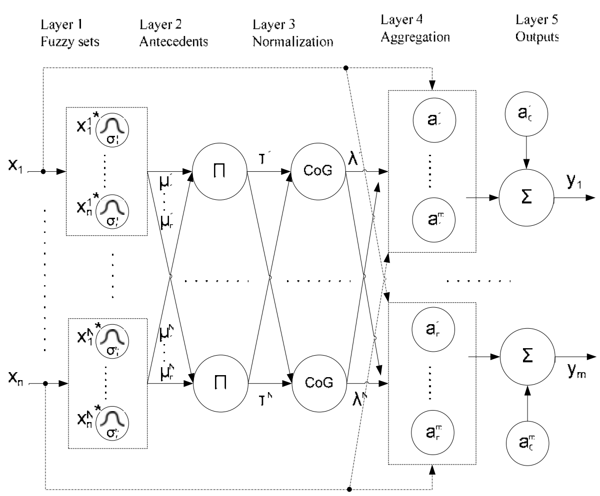

Figure 2. The proposed neuro-fuzzy system. Note, this structure is not pre-defined and fixed – it rather *evolves* 'from scratch' by learning from the data simultaneously with the parameter adjustment/adaptation

The first layer consists of neurons corresponding to the membership functions of particular fuzzy set. This layer takes as inputs the data, x and gives as output the degree, μ to which these fuzzy descriptors are satisfied. The second layer represents the antecedent parts of the fuzzy rules. It takes as inputs the membership functions values and gives as output the firing level of the ith rule, τ_i. The third layer of the network takes as inputs the firing levels of the respective rule, τi and gives as output the normalized firing level, λ_i as CoG of τ_i. As an alternative one can use 'winner takes all' operator. This operator is used usually in classification, while CoG is preferred for time-series prediction and general system modeling and control. The fourth layer aggregates the antecedent and the consequent part that represents the local sub-systems (singletons or hyper planes). Finally, the last fifth layer forms the total output of the NF system. It performs a weighed summation of local sub-systems according to (4)-(5).

B. Real-time Learning Methodology

Due to the lack of space only the basic concept of the learning methodology will be outlined in this paper. For more details, please refer to [13,20,22]. Learning is combining unsupervised learning in respect to the antecedent part of the model (1) (that is in terms of the model structure) with the supervised in terms of the consequent parameters, *a*.

In this way, the learning is o fthe second type according to the classification given in section I (a two-phase process). Note, that both phases (models structure identification using on-line clustering and model parameter identification using a version of coupled recursive least squares learning) are perfomed together at the model update stage, which is combined with the prediction step and both are performed per time instant. This type of (model update)-(model use for prediction) is typical for on-line estimation, adaptive control etc. [43].

Each one of the fuzzy rules of type (1) operate in certain sub-area of the input/output data space, $z = \left[x^T; y^T\right]^T$; $z \in R^{n+m}$. To identify these regions one can employ real-time clustering thus effectively learning the antecedent part of the fuzzy rules [33]. Two parameters are needed to define a membership function of the type (7), namely the focal point, x^{i*} and the spread, σ_j^i . If locate the focal points of the rules, x^{i*} at the cluster centre (note, only coordinates for the inputs are used to define the focal point although coordinates of the outputs are also used in the clustering) and if determine the spread, σ_j^i based on the data the antecedent part of the fuzzy rules are defined. Details about the real-time clustering approach used in this paper can be found in [33]. It can be noted that this clustering method has the following specific features that separate it from the other clustering approaches:

- ✓ It is non-iterative (no search is involved);
- ✓ It has very low memory requirements, because recursive calculations are used;
- ✓ It is fully unsupervised in the sense that number of clusters are not pre-defined (they are determined based on the data density alone);
- ✓ it can start 'from scratch' from the very first data

sample assumed to be the first cluster centre;
- ✓ changes of the cluster number and parameters are gradual, incremental, not abrupt.

Once the antecedent part of the fuzzy model is determined and fixed the parameters of the consequent part, a^i can be identified using fuzzily weighted RLS as detailed in [13]. The overall output of the evolving NF system can be given in vector form as follows:

$$y = \psi^T \theta \tag{8}$$

where $\theta = \left[(\pi^1)^T, (\pi^2)^T, ..., (\pi^N)^T\right]^T$ is a vector formed by the sub-system parameters; $\psi = [\lambda^1 x_e^T, \lambda^2 x_e^T, ..., \lambda^N x_e^T]^T$ is a vector of the inputs that are weighted by the normalized activation levels of the rules, λ^i *, i=[1,N]* for the first order Takagi-Sugeno model (2) and $\psi = [\lambda^1, \lambda^2, ..., \lambda^N]^T$ for the simplified Mamdani, (3).

For a given data point, $\left[x_k^T; y_k^T\right]^T$ the optimal in least squares sense solution, $\hat{\theta}_k$ that minimizes the following cost function:

$$\left(Y - \Psi^T \theta\right)^T \left(Y - \Psi^T \theta\right) \rightarrow \min \tag{9}$$

can be found applying fuzzily weighted RLS as detailed in [13].

It should be noted that the real-time algorithm must perform both tasks (data partitioning and parameter estimation) at the same time instant (per data point) for a time significantly shorter than the sampling period.

In this way, the antecedent part of the rules can be determined in a fully unsupervised way, while the consequent part requires a supervised feedback. Note that the clustering sub-problem does not require a consequent part. Classification sub-problem requires 'winner-takes-all' aggregation as given by equation (5a). The error feedback used in the supervised learning guarantees optimality (subject to fixed rule base structure) of the parameters of the consequent part.

C. *Knowledge and Data Integration*

In this paper we treat *e*-ntelligent systems through the prism of the knowledge and data integration (KDI) approach and so-called participatory learning [33]. KDI paradigm brings together the adaptation (which is relatively well covered by parameter adaptation techniques known from the 'conventional' adaptive systems theory [43]) 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 [43]). The problem of acquiring new data that does not support the *a priori* assumed model structure has not been addressed before the introduction of the evolving modelling concept. An e-ntelligent system, by differ from a conventional system, including a 'conventional' 'intelligent' system, continuously *learn new data to integrate this data with existing models*. The incoming data may contain new valuable information regarding new regimes or states of the system in question. The e-ntelligent system develops their structure and functionality continuously, always *adapting and modifying its knowledge representation*.

The introduction of the e-ntelligent system 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 system to accommodate new (reliable) data set 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 [36], a regression formula to predict outcome of cancer [37], an analytical formula to evaluate renal functions [38], a logistic regression formula for evaluating the risk of cardiac events [39], a set of rules for the prediction of outcome of trauma, gene expression classification and prognostic models [40,41], models of gene regulatory networks [42], and many more, e.g. [43].

One existing tool for reconciliation of the newly acquired information and the existing knowledge/understanding is the well known participatory learning approach by R. Yager [33]. It was recently applied to EFS learning [34,35].

D. *Illustrative Applications of EFS technique*

The proposed EFS approach has been applied to a wide range of problems (both benchmarks and real). Due to space limitations only one specific problem of an e-ntelligent sensor of exhaust gases of car engines will be given in this communication with illustrative purposes.

The real experimental data (courtesy of Dr. Edwin Lughofer, Johan Kepler University, Linz, Austria) concerns the problem of real-time modelling NOx emissions from the exhaust in a car engine using the following three input attributes measured in real-time: *N* - engine rotation speed, *rpm; P2* - pressure offset in cylinders, *bar; Te* - engine output torque, *Nm, Nd* – speed at the dynamometer, *rpm.* Based on the knowledge of the problem the following input/output dependence is expected [44]:

$$NOx_k = f(N_{k-4}, P2_{k-5}, Te_{k-5}, Nd_{k-6}, N_{k-6}) \tag{10}$$

The sampling period of 1s has been used and thus a prediction 4 second will be possible if a non-linear (and non-stationary) dependence of the above type can be identified in real-time. A test was carried out with a sequence of just over 4 hours of real data (sampled into 1491 samples with a sampling interval of 1s) [44]. Fig. 3 illustrates the comparison of the predicted in real-time versus the real data sequence of the NOx emissions in the car engines for about 2.5 minutes (samples between 667 and 824).

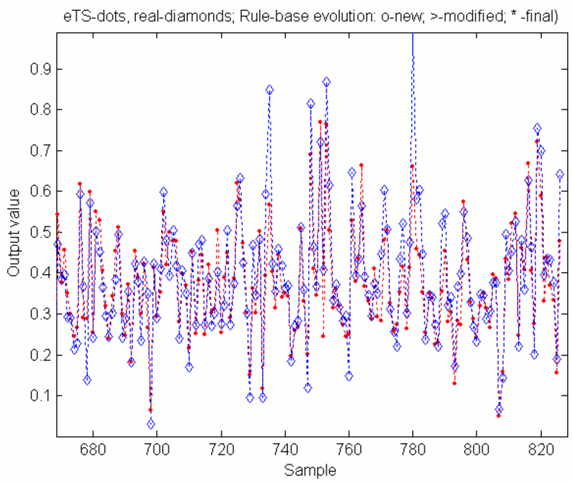

Figure 3. Validation of the e-ntelligent sensor with real data from car eng test

This result was achieved with a fuzzy rule-based system that started to learn 'from scratch' (with an empty rule-base) and have developed its rule-base by learning from data in real-time. At the end of the test it contained 7 fuzzy rules:

R_1: **IF**(N_{k-4} *is Medium)* **AND** ($P_2^{offset}{}_{k-5}$ *is Low)* **AND** (Te_{k-5} *is High)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Medium)* **THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_2: **IF**(N_{k-4} *is Low)* **AND** ($P_2^{offset}{}_{k-5}$ *is Low)* **AND** (Te_{k-5} *is Very Low)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Medium)* **THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_3: **IF**(N_{k-4} *is Medium)* **AND** ($P_2^{offset}{}_{k-5}$ *is Medium)* **AND** (Te_{k-5} *is High)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Low)* **THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_4: **IF**(N_{k-4} *is Low)* **AND** ($P_2^{offset}{}_{k-5}$ *is Very Low)* **AND** (Te_{k-5} *is Medium)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Medium)***THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_5: **IF**(N_{k-4} *is Very High)* **AND** ($P_2^{offset}{}_{k-5}$ *is Very Low)* **AND** (Te_{k-5} *is Low)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Low)* **THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_6: **IF**(N_{k-4} *is Low)* **AND** ($P_2^{offset}{}_{k-5}$ *is High)* **AND** (Te_{k-5} *is Low)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Very High)* **THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_7: **IF**(N_{k-4} *is Low)* **AND** ($P_2^{offset}{}_{k-5}$ *is Very High)* **AND** (Te_{k-5} *is Very High)* **AND** (Nd_{k-6} *is ...)* **AND** (N_{k-6} *is Very High)***THEN** $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

Figure 4 depicts four of the fuzzy sets of the first fuzzy rule (evolved automatically from the data):

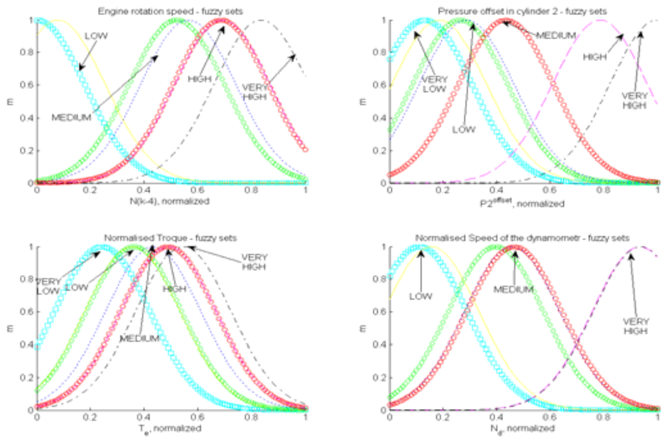

Figure 4. Fuzzy sets of Engine rotation speed, Pressure in the cylinders, torque, and the speed of the dynamometer of a car engine (data are real, but normalized).

Figure 5 depicts the evolution with time of the parameters in the consequent part of the fuzzy rules.

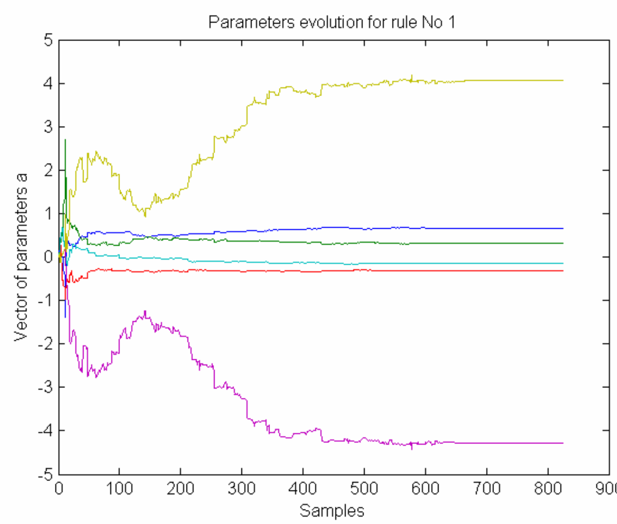

Figure 5. Parameters evolution of the consequents of one of the fuzzy rules.

The precision of the evolved model can be measured in different metrics:

- Variance Accounted For (VAF): 85.034
- Correlation **0.92213**
- MSE 0.0037546
- RMSE 0.061275
- NDEI 0.38991

It is well known that the ideal values for the VAF is 100%, for the correlation is 1, and for the remaining metrics is 0.

This e-ntelligent sensor has the following advantages:

- ✓ It is precise (the results were compared to the known off-line and on-line approaches that are based on a fixed structure models [44] and EFS demonstrated superior performance;
- ✓ It is flexible and has inherently build-in robustness and insensitivity to noise and outliers

[13];

- ✓ It is computationally efficient – a version of this approach was developed on chip (FPGA) [45];
- ✓ It does not require prior knowledge about the structure of the model/sensor – it can start learning 'from scratch';
- ✓ It is suitable for re-calibration of sensors;
- ✓ It can be used from extracting transparent linguistic rules from data streams in real-time.

III. Evolving Connectionist Systems (ECOS)

A. *General Principles of ECOS*

Evolving connectionist systems (ECOS) are modular connectionist-based systems that evolve their structure and functionality in a continuous, self-organised, on-line, adaptive, interactive way from incoming information. They can process both data and knowledge in a supervised and/or unsupervised way [10,11].

ECOS learn local models from data through clustering of the data and associating a local output function for each cluster. Clusters of data are created based on similarity between data samples either in the input space (this is the case in some of the ECOS models, e.g. the dynamic neuro-fuzzy inference system DENFIS [25]), or in both the input space and the output space (this is the case in the EFuNN models [10]). Samples that have a distance to an existing cluster center (rule node) N of less than a threshold Rmax (for the EfuNN models it is also needed that the output vectors of these samples are different from the output value of this cluster center in not more than an error tolerance E) are allocated to the same cluster *Nc*. Samples that do not fit into existing clusters, form new clusters as they arrive in time. Cluster centers are continuously adjusted according to new data samples, and new clusters are created incrementally.

The similarity between a sample $S = (\mathbf{x},\mathbf{y})$ and an existing rule node $N = (W_1,W_2)$ can be measured in different ways, the most popular of them being the normalized Euclidean distance:

$$d(S,N) = [\Sigma_{(i=1,..,n)} (x_i - W_1(i))^2] / n, \quad (11)$$

where n is the number of the input variables.

ECOS learn from data and automatically create or update a local output function for each cluster, the function being represented in the W_2 connection weights, thus creating local models. Each model is represented as a local rule with an antecedent – the cluster area, and a consequent – the output function applied to data in this cluster, e.g.:

IF (data is in cluster Nc) THEN (the output is calculated with a function Fc) (12)

Implementations of the ECOS framework require connectionist structures that support these principles. One implementation of ECOS is the evolving fuzzy neural network (EFuNN).

B. *Evolving Fuzzy Neural Network EFuNN*

A general EFuNN architecture has 5 feed-forward layers and a feedback layer of neurons, but the second and the fourth fuzzy representation layers and also - the feedback layer, are optional – fig. 6. The third layer contains rule nodes that evolve through supervised / unsupervised learning. The rule nodes represent prototypes of input-output data associations. Each rule node *r* is defined by two vectors of connection weights, $W_1(r)$ and $W_2(r)$, the latter being adjusted through supervised learning based on the output error, and the former being adjusted through unsupervised learning based on a similarity measure within a local area of the problem space. The fourth layer of neurons represents fuzzy quantization for the output variables, similar to the input fuzzy neurons representation. The fifth layer represents the real values for the output variables.

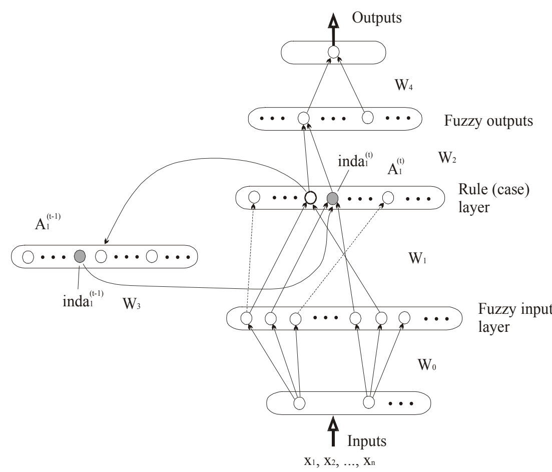

Figure 6. An EFuNN architecture with a short term memory and feedback connections (adapted from [10,11,46])

The evolving process can be based on either of the two assumptions: (1) rule nodes exist prior to learning and only connections are created during learning; (2) all nodes are created during the evolving process.

Each rule node (e.g., r_1) represents an association between a hyper-sphere from the fuzzy input space and a hyper-sphere from the fuzzy output space, the $W_1(r_j)$ connection weights representing the co-ordinates of the centre of the sphere in the fuzzy input space, and the $W_2(r_j)$ – the co-ordinates in the fuzzy output space. The radius of an input hyper-sphere of a rule node is defined as (1 – *Sthr*), where *Sthr* is the *sensitivity threshold* parameter defining the minimum activation of a rule node (e.g., r_1) to an input vector (e.g., (Xd_2, Yd_2)) in order for the new input vector to be associated to this rule node.

Through the process of associating (learning) a new data vector Xd to a rule node, the centre of this node hyper-sphere is adjusted in the fuzzy input space depending on a *learning rate* lr_1 and in the fuzzy output space depending on a *learning rate* lr_2. The adjustment of the centre r_1^1 to its new position r_1^2 can be represented mathematically by the change in the connection weights of the rule node r_1 from $W_1(r_1^1)$ and $W_2(r_1^1)$ to $W_1(r_1^2)$ and $W_2(r_1^2)$ as it is presented in the following vector operations:

$$W_1(r_1^2) = W_1(r_1^1) + lr_1 * Ds(Xd, W_1(r_1^1)) \tag{13}$$

$$W_2(r_1^2) = W_2(r_1^1) + lr_2 * Err(Yd, Yd') * A_1(r_1^1) \tag{14}$$

where: $Err(Yd, Yd') = Ds(Yd, Yd')$ is the distance between the desired and the obtained in the system output vectors in the output space; $A_1(r_1^1)$ is the activation of the rule node r_1^1 for the input vector Xd.

While the connection weights from W_1 and W_2 capture spatial characteristics of the learned data (centres of hyper-spheres), the temporal layer of connection weights W_3 from Fig.6 captures temporal dependences between consecutive data examples. If the winning rule node at the moment $(t-1)$ (to which the input data vector at the moment $(t-1)$ was associated), was $r_1 = inda_1(t-1)$, and the winning node at the moment t is $r_2 = inda_1(t)$, then a connection between the two nodes is established as follows:

$$W_3(r_1, r_2)^{(t)} = W_3(r_1, r_2)^{(t-1)} + lr_3 * A_1(r_1)^{(t-1)} * A_1(r_2)^{(t)} \tag{15}$$

where: $A_1(r)^{(t)}$ denotes the activation of a rule node r at a time moment (t); lr_3 defines the degree to which the EFuNN associates links between rules (clusters, prototypes) that include consecutive data examples (if $lr_3 = 0$, no temporal associations are learned in an EFuNN).

The following is a *new learning rule* that takes into account both spatial similarity and temporal correlation) through introducing two parameters Ss and Tc , such that the activation of a rule node r for a new data example d_{new} is defined as the following vector operation:

$$A_1(r,d_{new}) = f(Ss*D(W1(r),d_{new})+Tc*W_3(r^{(t-1)}, r)) \tag{16}$$

where: f is the activation function of the rule node r; $D(W1(r), d_{new})$ is the normalized fuzzy distance between the new input vector and the W1(r) representing the spatial component; $r^{(t-1)}$ is the winning neuron at time moment $(t-1)$. The second term in equation (16) represents the temporal component.

An EFuNN functional implementation can include pruning nodes and aggregating nodes [10,11]. An example of a pruning rule is:

IF (a rule node r_j is *OLD*) AND (average activation $A_{1av}(r_j)$ is *LOW*) AND (the density of the neighbouring area of neurons is *HIGH* or *MODERATE* THEN the probability of pruning node (r_j) is *HIGH*.

Nodes can also be aggregated [10,11].

C. *Dynamic Evolving Neuro-Fuzzy Inference Systems (DENFIS) [25]*

While EFuNN is a fuzzy neural network that evolves incrementally its structure and functionality using supervised clustering, DENFIS is a dynamic fuzzy inference system that incrementally creates Takagi-Sugeno fuzzy rules to accommodate data in unsupervised learned clusters. New fuzzy rules are created and updated during the operation of the system. At each time moment the output of DENFIS is calculated through a fuzzy inference system based on m-most activated fuzzy rules which are dynamically selected from the existing fuzzy rule set. As the knowledge, fuzzy rules can be inserted into DENFIS before, or during its learning process and, they can also be extracted during the learning process or after it. The fuzzy rules used in DENFIS are indicated as follows:

$$\begin{aligned} R_l:\ & \text{if } x_1 \text{ is } F_{l1} \text{ and } x_2 \text{ is } F_{l2} \text{ and } \dots \text{ and } x_P \text{ is } F_{lP}, \\ & \text{then } y_l = b_{l0} + b_{l1}x_1 + b_{l2}x_2 + \dots + b_{lP}x_P \end{aligned} \tag{17}$$

where "x_j is F_{lj}", $l = 1, 2, \dots m$; $j = 1, 2, \dots P$, are $M \times P$ fuzzy propositions that form m antecedents for m fuzzy rules respectively; x_j, $j = 1, 2, \dots, P$, are antecedent variables defined over universes of discourse X_j, $j = 1, 2, \dots, P$, and F_{lj}, $l = 1, 2, \dots M$; $j = 1, 2, \dots, P$ are fuzzy sets defined by their fuzzy membership functions $\mu_{Flj}: X_j \to [0, 1]$, $l = 1, 2, \dots M$; $j = 1, 2, \dots, P$. In the consequent parts of fuzzy rules, y_l , $l = 1, 2, \dots m$, are the consequent variables defined by linear functions.

In DENFIS, F_{lj} are defined by a *Gaussian* membership function. All fuzzy rules in DENFIS are created and updated during a possible 'one-pass' training process by applying the Evolving Clustering Method (ECM) and the Weighted Recursive Least Square Estimator with Forgetting Factors (WRLSE) [25].

The ECOS models have the following advantages: (1) incremental, fast learning (possibly 'one pass'); (2) on-line adaptation; (3) 'open' structure; (4) allowing for time and space representation based on biological plausibility; (5) rule extraction and rule insertion; (6) data and knowledge integration (as discussed below)

D. Integrating Knowledge (Old Models) and Data in ECOS

As the *eIS*, and ECOS in particular, are adaptive, knowledge-based systems, they can accommodate both existing knowledge on the problem (e.g. formulas, models) and new data, allowing for incremental adaptation of the system's rule representation.

In many domain areas, such as medical decision support, there are existing regression formulas and new data is accumulating in time, making the integration of both an important issue for a better decision support.

In [46,47] ECOS are used to accommodate regression formulas and new data in the following away. The formula is first used to generate "historical" data. This data is used to train an ECOS as an initial knowledge representation architecture. Then the ECOS system is further trained (adapted) on the new data.

In [48] a novel method of "Integrated kernel-regression knowledge-based neural networks" is presented for the integration of several, used in practice, regression formulas and new data into one system. The method optimizes the size of the clusters of the data using different kernels, and for each cluster – a suitable type of regression is chosen and the parameters are adapted on the data:

$$y(x) = G_1(x)\, F_1(x) + G_2(x)\, F_2(x) + .. + G_M(x)\, F_M(x) \quad (18)$$

where, $x = [x1, x2, ..., x_P]$ is the input vector; y is the output vector; G_l are kernel functions; and F_l are regression formulas, l = 1, 2, … M.

E. Incremental parameter and feature oprimisation using GA, incremental PCA and LDA

An ECOS evolves its structure and functionality in time from incoming data, for which the dynamics may not be known in advance. That requires an incremental (possibly - on-line) parameter and feature optimization. One way to optimize these parameters and obtain an optimal for the time moment model according to certain criteria (e.g. classification accuracy) is through evolutionary computation, e.g. GA [7]. GA optimization can be applied on a population of individual models that are trained and tested on consecutive chunks of data, so that at any time of the operation of the ECOS the best model (e.g. the model with the highest accuracy/ fitness) is selected. A methodology and examples are given in [55, 59].

In fig. 7 a simple ECOS model, called ECF, is optimized with the use of GA. ECF is characterized by 4 parameters (maximum field radius Rmax, minimum field radius Rmin, number of nodes m to use for a new vector; number of membership functions, epochs to train) and initial 12 input features describing the outcome of DLBCL cancer of 56 patients [40] is optimized as shown in fig. 7.

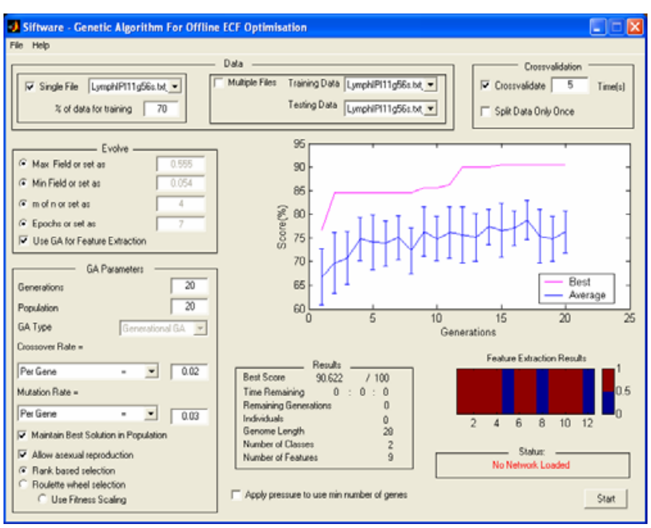

Figure 7. Using GA for parameter and feature optimization of a simple ECOS – ECF (experiments are done in a software environment NeuCom – www.theneucom.com)

The variables are a clinical variable (IPI index) and 11 genes selected in for the prognosis of DLBCL cancer outcome. In the experiment shown in Fig. 7 both the ECF parameters and features are optimised with the use of a GA which ran over 20 generations. At each generation, there are 20 ECF models in a population, having different parameter values and feature sets, and a fitness criteria of overall highest accuracy for the smallest number of features is used. The optimal ECF parameters are given in the figure and the best model has an overall accuracy of 90.66%, which is higher than any of the non-optimised models. The optimal values of all the ECF parameters and also the used variables (variables 5,8 and 12 are not included) are shown in the figure.

In some cases, PCA or LDA transformations need to be performed on the input feature set to obtain a more compact input vectors and to improve the accuracy of the model. Incremental PCA and incremental LDA methods are presented in [49,54]. After features are selected in an incremental way, the ECOS if adapted to these features.

F. A Framework of Multimodal eIS

So far, this paper described some methods for building adaptive, evolving, knowledge based models from data. eIS may require several evolving models. Fig. 8 presents a framework of eIS, that consists of several parts: several e-models (EM), higher level decision part, adaptation part, featiure selection part where new features may be added in time, knowledge (rules) extraction part; interaction with environment and an output module [11].

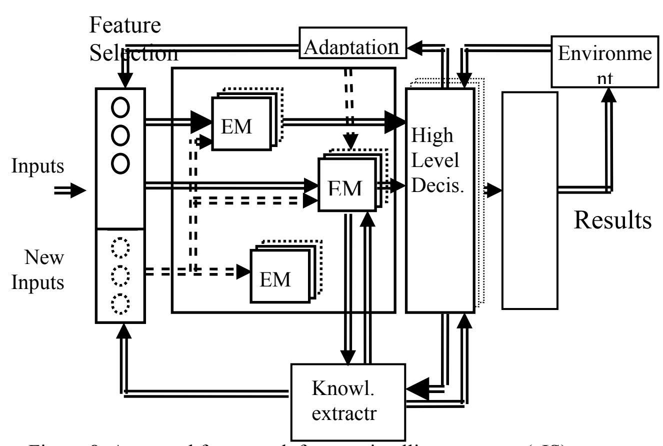

Figure 8. A general framework for an e-intelligent system (eIS)

IV. APPLICATIONS OF ECOS AND *EIS*

ECOS and eIS in general have been used for a range of applications so far, where adaptation to new data and knowledge representation are crucial requirements. Here we present only few examples of them.

A. Medical Decision Support

A renal function prognostic system with the use of DENFIS is presented in [51]. The initial data set used has 447 samples, collected at hospitals in New Zealand and Australia. Each of the patient records includes six variables (features): 1) age, 2) gender, 3) serum creatinine, 4) serum albumin, 5) race, 6) blood urea nitrogen concentrations, and one output - the glomerular filtration rate value (GFR) [38]. For every data cluster, a local model in DENFIS is derived as a logistic regression – fig.9.

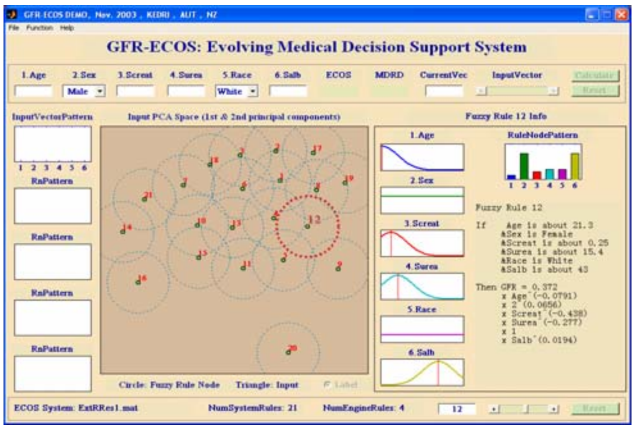

Fig.9. A snapshot of an adaptive medical decision support system for renal function evaluation [51]. The fuzzy rule on the right side is a regression model of the data derived and updated for the highlighted cluster.

B. Bioinformatics

Bioinformatics is the area concerned with the biological data storage, analysis, representation, modeling and knowledge discovery. A review of problems and possible solutions is given in [52, 59]. Several problems in bioinformatics have been successfully solved with the use of adaptive eIS:
(a) Micro-array gene expression data analysis and pattern discovery [40,59]. Figure 10 shows a graphical representation of 5 EFuNN rules, each representing a profile of samples clustered together, each of them belonging to the class of good prognosis (class 1) or - bad prognosis (class 2) [11, 46, 52, 56].

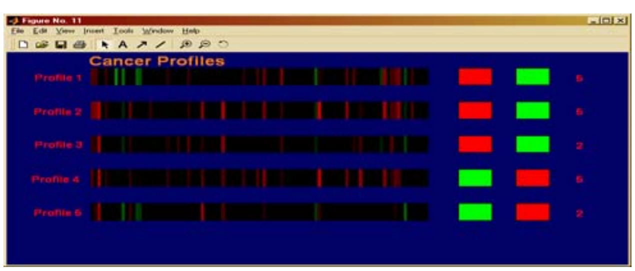

Figure 10. A graphical representation of 5 EFuNN rules, each representing a profile of samples clustered together, and belonging to the class of good prognosis (class 1), or - bad prognosis (class 2) [11,46,40].

(b) Gene regulatory network modeling (GRN)
GRNs describe the regulatory interaction between genes in a cell [42,59]. Co-expressed genes over time relate to each other – either one regulates the other, or both are regulated by same other genes. eIS are useful tools for building adaptive GRN models from time course gene expression data [50,53.

In [60] EFuNN and DENFIS have been used to derive a GRN of 4 genes from a cell line time course data. The GRN model is then used to predict future values of the genes over time. Rules can be extracted that explain the relationship between the expression of genes at different time moments, e.g.:

IF g13(t) is High (0.87) and g23(t) is Low (0.9)
THEN g87 (t+dt) is High (0.6) and g103(t+dt) is Low

C. Neuroinformatics and Brain Study
Brain models can be evolved incrementally from EEG brain data collected from individuals that belong to different categories, or from different brain states of the same individual, using standard EEG equipment – fig.11.

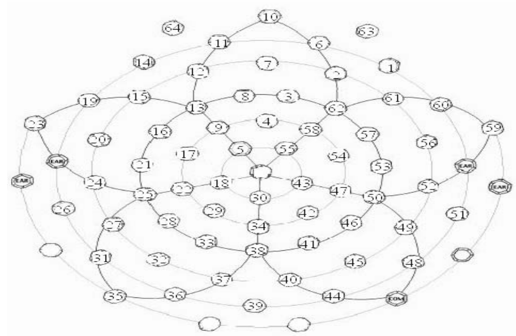

Figure 11. A standard set of EEG electrodes to collect data from a brain of an individual, used to evolve a model

After evolving models are trained on EEG channel data, rules Can be extracted in the form of: IF Channels 13 and 27 have high values, THEN the state of the brain is sleep. This type of research is reported in [11,46].

D. Multimodal Information Processing and Biometrics
Combining speech, image and other modalities in an adaptive way, where new speech samples can be added in time, new images, new modalities (e.g. fingerprints) for a person recognition, person identification and person verification is a promising area of application for eIS – fig. 12 [11,46].

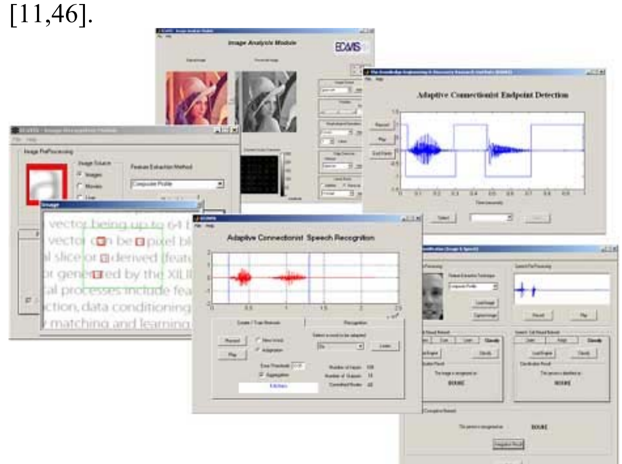

Figure 12. An example of a multimodal (speech and image) eIS

E. Financial and Business Forecast

Adaptive learning and future value prediction of financial and business time series with the use of eIS is reported in [46]. Fig.13 shows the weekly on-line prediction of the exchange rate Euro/US$ for 1-,2-,3- and 4 weeks ahead using 3 input variables: ERate, Euro/Yen, Stock-E/US, with 4 week time lags each [46]. The lower figure shows the number of rule nodes evolved in an EFuNN structure applying also aggregation of nodes.

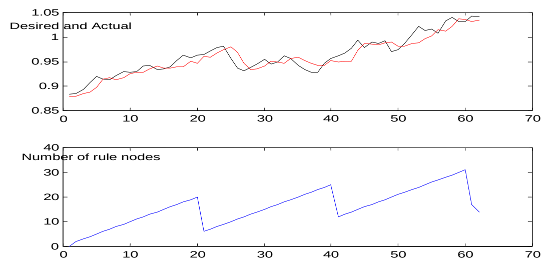

Figure 13. The weekly on-line prediction of the exchange rate Euro/US$ and the number of the evolved and aggregated rule nodes in an EFuNN architecture.

F. Autonomous mobile robots

Evolving, autonomous learning robots, that communicate between each other, is an area of growing interest and potential for eIS. In [66] ECOS are used to control the position of robocup robots that adapt on the spot to the opponent – fig. 14.

Figure 14. An eIS is used to locate the players on the field while adapting on-line to the opponent's strategy [66]

V. CONCLUSIONS

The concept of eIS has been presented as an effective tool to address the problem of modeling non-stationary, highly non-linear processes on-line, in real-time. The basic elements of the concept and its procedure has been outlined without going to details, which are available in a number of recently published papers by the authors and their collaborators. In essence, the concept of *evolving* intelligent systems *eIS* can be considered as a higher level adaptation that concerns model structure as well as model parameters.

The eIS approach is demonstrated here through two modelling constructs that the authors have introduced recently and are continuing to develop, namely the evolving connectionist systems, ECOS, and evolving fuzzy systems, EFS.

Further research is planned in application areas as highlighted in the paper, along with some novel generic methods of eIS to be developed, such as: transductive evolving systems [61,62]; evolving spiking neural networks [63]; evolving neurogenetic models [64,65]; evolving quantum inspired neural networks [46], and others.

The true intelligent systems must evolve their structure, functionality and knowledge – they can not be fixed *a priori.*

VI. ACKNOWLEDGEMENTS

The authors would like to thank Dr. E. Lughofer for kindly providing the experimental data from the car test engines. The research is also supported by the FRST/NERF funded project AUTX02001 in New Zealand.

REFERENCES

[1] Saridis, G N; Valavanis, K P, Analytical design of intelligent machines. Automatica. Vol. 24, no. 2, pp. 123-133. 1988

[2] Geyer-Schulz A. (1995) *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning*, Studies in Fuziness, **v.3**, Berlin, Germany: Physica Verlag

[3] Zadeh L.A., Fuzzy logic, neural networks, and soft computing, Communications to the ACM, vol.37(3), March 1994, pp.77-84, ISSN:0001-0782

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

[5] Werbos P. (1990) Backpropagation Trough Time: What it Does and How to do it, *Proc. of the IEEE Conference on Neural Networks*, v.78 (10), pp.1550-1560

[6] Holland J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[7] Goldberg D.E. (1989) *Genetic Algorithms in Search, Optimization and machine Learning*, Reading, MA, USA: Addison-Wesley

[8] Koza J., *Genetic Programming: On the Programming of Computers by Means of Natural Selection,* USA: MIT Press,1992

[9] Hornby A. S., *Oxford Advance Learner's Dictionary*, Oxford University Press, 1974.

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

[11] Kasabov, N. *Evolving connectionist systems: Methods and applications in bioinformatics, brain study, and intelligent machines*, Springer Verlag, London, Heidelberg, NY, 2002

[12] Angelov P, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.

[13] Angelov, P., D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models*", IEEE Trans. on Systems, Man and Cybernetics, part B,* vol.34, No1, pp. 484-498, 2004.

[14] Ljung L. (1987) *System Identification: Theory for the User*, New Jersey, USA: Prentice-Hall

[15] Wang, L.-X. "Fuzzy Systems are Universal Approximators," *Proc. FUZZ-IEEE,* San Diego, CA, USA, pp.1163-1170, 1992.

[16] Hornik K. (1991) Approximation Capabilities of Multilayer Feedforward Network, *Neural Network*, v.4, pp.251-257.

[17] Leng G., T.M. McGuinty, G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organizing fuzzy neural network," *Fuzzy Sets and Systems,* vol. 150 (2), pp.211-243, 2005.

[18] Juang C.-F., X.-T. Lin "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. on NN,* vol. 10, pp. 828-845, 1999

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

[20] Angelov P., C. Xydeas, Fuzzy Systems Design: Direct and Indirect Approaches, *Soft Computing*, vol. 10 (9), pp.836-849, July 2006, special issue on New Trends in the Fuzzy Modelling part I: Novel Approaches.

[21] Kim K., J. Baek, E. Kim, M. Park, "TSK Fuzzy model based on-line identification," *Proc. 11th IFSA World Congress,* Beijing, China, 2005, pp.1435-1439.

[22] Angelov, P. , D. Filev, "Simpl_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models," *The 2005 IEEE Intern. Conf. on Fuzzy Systems FUZZ-IEEE*, Reno, NE, USA, 2005, pp.1068-1073.

[23] Angelov P., C. Xydeas, D. Filev, *On-line Identification of MIMO Evolving Takagi-Sugeno Fuzzy Models, Intern. Joint Conf. on NN and Intern. Conf. on Fuzzy Systems, IJCNN-FUZZ-IEEE, Budapest, Hungary, 25-29 July, 2004, 55-60, ISBN 0-7803-8354-0*.

[24] Huang G.-B., P. Saratchandran, N. Sundarajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. on NN,* vol.16 (1) 57-67, 2005.

[25] Kasabov N., Q. Song "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans. on Fuzzy Systems, Vol.10* (2), pp. 144-154, 2002.

[26] Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth, *From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining,* MIT Press. 1996.

[27] Kalman R. E., A New Approach to linear filtering and prediction problem, Transactions of the ASME, Ser. D, Journal of Basic Engineering, vol. 82, pp.34-45, 1960.

[28] Crisan D., Particle Filters – A Theoretical perspective, Sequential Monte Carlo Methods in Practice, A Doucet, J.F.G., de freitas, N. J. Gordon, Eds., Berlin: Springer Verlag, 2001.

[29] Yager R., D. Filev, *Essentials of Fuzzy Modeling and Control*, NewYork, USA: John Wiley and Sons, 1994.

[30] Takagi, T., M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. on Syst., Man & Cybernetics,* vol. 15, pp. 116-132, 1985.

[31] Chen R., J. S. Liu, Mixture Kalman Filters, Journal of the Royal Statistical Society, ser. B, vol. 62, pp.493-508, 2000.

[32] Angelov, P., D. Filev, N. Kasabov, O. Cordon (Eds.) Evolving Fuzzy Systems, Proc. 2nd Intern. Symposium on Evolving Fuzzy Systems, 7-9 Sept. 2006, Ambelside, Lake District, UK, IEEE, 1-350pp., ISBN, 0-7803-9719-3, to appear

[33] Yager R., "A model of participatory learning," *IEEE Transactions on Systems, Man, and Cybernetics,* 20 (5) 1229-1234, 1990.

[34] Lima E., Gomide F., R. Balini, 2006, to appear

[35] Angelov, P., X.-W. Zhou Evolving Fuzzy Systems from Data Streams in Real-Time, In: Proc. 2nd Intern. Symposium on Evolving Fuzzy Systems (Angelov P., D. Filev, N. Kasabov, O. Cordon Eds.), 7-9 Sept. 2006, Ambelside, Lake District, UK, IEEE, 1-350pp., ISBN, 0-7803-9719-3, to appear

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

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

[38] 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).

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

[40] 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).

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

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

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

[44] Lughofer, E., Klement, E.P. Lujan, J.M. Guardiola, C., Model-based fault detection in multi-sensor measurement systems, Intelligent Systems, Proc. 2nd Intern. IEEE Conference, 22-24 June 2004, Vol.1, pp. 184-189.

[45] Angelov, P., M. Everett, EvoMap: On-Chip Implementation of Intelligent Information Modelling using EVOlving MAPping, 2005, Lancaster University, Lancaster, UK, pp.1-15.

[46] Kasabov, N. *Evolving connectionist systems: Brain-, gene-, and , quantum inspired computational intelligence*, Springer Verlag, London, Heidelberg, NY, 2006

[47] Kasabov, N. Adaptation and Interaction in Dynamical Systems: Modelling and Rule Discovery Through Evolving Connectionist Systems, *Applied Soft Computing*, 2006, Volume 6, Issue 3, pages 307-322

[48] Song, Q., N. Kasabov, T. Ma, M. Marshall, Integrating regression formulas and kernel functions into locally adaptive knowledge-based neural networks: a case study on renal function evaluation, *Artificial Intelligence in Medicine*, 2006.

[49] Pang, S., S. Ozawa and N. Kasabov, Incremental Linear Discriminant Analysis for Classification of Data Streams, *IEEE Trans. SMC-B,* vol. 35, No. 5, 2005, 905 – 914

[50] Chan, Z., N.Kasabov, A Two-Stage Methodology for Gene Regulatory Network Extraction from Time-Course Gene Expression Data, *Expert Systems with Applications:* An International Journal (ISSN: 0957-4174), Special issue on Intelligent Bioinformatics Systems, December, 2005.

[51] Marshall, M.R., Q. Song, T.M. Ma, S. MacDonell, N.Kasabov, Evolving Connectionist System versus Algebraic Formulae for Prediction of Renal Function from Serum Creatinine, *Kidney International*, vol. 67 (2005), pp. 1944 – 1954

[52] Kasabov, N. Knowledge based neural networks for gene expression data analysis, modelling and profile discovery, *Drug Discovery Today: BIOSILICO*, vol. 2, No. 6, November 2004, pp. 253-261.

[53] Kasabov, N., Z . S.H. Chan, Igor Sidorov and Dimiter Dimitrov, Gene Regulatory Network Discovery for Time Series Gene Expression Data – A Computational Intelligence Approach, *Lecture Notes in Computer Science*, Vol.3316, 2004, Springer Verlag, 1344-1353.

[54] Ozawa, S., S. Pang and N. Kasabov, A Modified Incremental Principal Component Analysis for On-Line Learning of Feature Space and Classifier, *Lecture Notes in Artificial Intelligence* LNAI, Volume 3157, Springer-Verlag, Berlin, Heidelberg, 2004, 231 – 240.

[55] Chan Z. and N.Kasabov, Evolutionary computation for on-line and off-line parameter tuning of evolving fuzzy neural networks, *Int. J. of Computational Intelligence and Applications*, Imperial College Press, vol. 4, N.3, September 2004, 309-319

[56] Futschik, M.., A.Reeve, and Kasabov, N. Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue, *Artificial Intelligence in Medicine*, 28 (2003) 165-189

[57] Kasabov, N., Zeke Chan, Vishal Jain, Igor Sidorov and Dimiter Dimitrov, Computational Modelling of Gene Regulatory Networks, Chapter 8, in: Bajic., V and Tan Tin Wee (eds*), Information Processing and Living Systems*, Imperial College Press, Singapore, 2005, 673-686.

[58] Kasabov, N.., Z.Chan, Q.Song and D.Greer, Evolving neuro-fuzzy systems with evolutionary parameter self-optimisation, chapter in: *Do Adaptive Smart Systems exist?* Springer Verlag, Series Study in Fuzziness, vol.173, 2005

[59] Dimitrov, D. S ., Igor A. Sidorov and Nikola Kasabov Computational Biology, in: M. Rieth and W. Sommers (eds) *Handbook of Theoretical and Computational Nanotechnolog*y, Vol. 1 (1) American Scientific Publisher, Chapter 21,2004

[60] Kasabov, N. and D. Dimitrov, Evolving connectionist systems for gene regulatory network modelling and discovery, chapter in: L.Wang and Rajapakse (eds) *Advances in neuro-information processing*, Springer Verlag, 2004

[61] Song Q., and N. Kasabov, TWNFI: Transductive Weighted Neuro-Fuzzy Inference Method for Personalised Modelling, *Neural Networks*, 2006

[62] Song Q. and N.Kasabov, TNFI: Transductive neuro-fuzzy inference method for personalized modelling, *IEEE Transactions on Fuzzy Systems,* vol.13, No.6, December, 2005

[63] Kasabov, N., L. Benuskova L and Wysoski SG (2005) Computational neurogenetic modeling: integration of spiking neural networks, gene networks, and signal processing techniques. In: ICANN 2005, *LNCS* 3697, W. Duch et al (Eds), Springer-Verlag, Berlin Heidelberg, pp. 509-514

[64] Kasabov, N., and L. Benuskova, Computational Neurogenetics, *International Journal of Theoretical and Computational Nanoscience*, Vol. 1 (1) American Scientific Publisher, 2004, 47-61.

[65] Benuskova L., Wysoski S.G., Kasabov, N. (2006) Computational neurogenetic modeling: integration of spiking neural networks and gene networks. Proc. IJCNN, IEEE Press, 2006

[66] Huang, L., Song, Q., Kasabov, N., Evolving Connectionist Systems Based Role Allocation of Robots for Soccer Playing, Joint 2005 International Symposium on Intelligent Control & 13th Mediterranean Conference on Control and Automation (2005 ISIC-MED), June 27-29, 2005, Limassol, Cyprus