Comprehensible Hierarchical Intelligent (CHI) Framework for Monitoring and Preventive Maintenance of Aircraft Systems

Peter Owotoki and Fritz Mayer-Lindenberg

Department of Distributed Systems,
Hamburg University of Technology, Hamburg Harburg, Germany
{owotoki, mayer-lindenberg}@tu-harburg.de

Abstract — *The peculiarities of the aircraft monitoring and maintenance domain are described and shortcomings of the current monitoring methodology are revealed. It is also shown why a new approach using computational intelligence models, as a replacement for the current BITE models, is paramount. In section 2 a brief review of developments in computational intelligence research is given. After which we present the comprehensible hierarchical intelligent framework, as a conceptual non monolithic intelligent approach utilizing distributed CI models for monitoring. Finally we conclude with discussions on the implementation and justification for our approach and direction for future work.*

1 Domain Description

The maintenance of airplanes and their component systems utilizes a scheduled maintenance policy based on strictly controlled replacement of components at specified time intervals. This is a rather cost intensive strategy because, in addition to costs accrued from the overhead of the fixed controls, the possible life span of many of the components are not used up when they are replaced. Moreover the specification for the complex of loads acting on a component, which is used to determine when to replace it, is not exactly the same as that experienced at run time.

Sometimes it occurs that systems fail before the planned scheduled maintenance and need to be replaced immediately. This unscheduled maintenance leads to immense loss in time and capital since additional, previously unexpected, resources are expended on them outside of the standard scheduled maintenance process.

In Airbus airplanes, failures are captured, detected and isolated by the built in test equipment (BITE) within every electronic unit. The BITE provides robust circuitry and software for model based continuous monitoring the performance of the system in which it is housed. Outputs from the BITE are collected in a central computer where, incorporating additional pre specified information; they are used to ease the actions of the maintenance team and help make the maintenance process run more smoothly.

The BITE finds at least 95% of the failures and isolates at least 85%. Although impressive, some deficiencies in this model of monitoring have been identified by us:

- There is no provision for autonomous adaptive improvement of its internal model.
- It doesn't monitor the correlation and interdependence between subsystems.

– Its internal models for failure detection do not take into full consideration the impact of environmental conditions at run time.
– There is no provision for discovering and incorporating expert knowledge of the maintenance crew or new knowledge from the normal operation of the systems.
– It fails to detect transient failures which results in the very vexing no fault found (NFF) problems where failures are reported by BITE, but their existence cannot be confirmed by the maintenance staff. This leads to delay and uncertainty in deciding whether to replace a component or not, and all of these leads to more costs.
– There is no inherent explanation of the process of generating the failure in a way that is comprehensible to humans

A new maintenance paradigm is needed. One based on continuous, comprehensible, dynamic and intelligent monitoring utilizing computational intelligence models that closely model the reasoning and decision process of human experts. With this new approach unscheduled maintenance is no longer unexpected because by continuously monitoring the systems, unexpected errors are preemptively discovered and communicated. New knowledge is discovered from the normal operations of the system and the interrelationships between its components. This knowledge is used to improve the monitoring process thus leading to a system that learns and improves over time. In addition there is improved understanding of the system as a whole unit. The combined effect will lead to reduction in the costs of the scheduled and unscheduled maintenance processes. The time interval between scheduled maintenance is extended because removal of parts is no longer dependent on guess work but on hard data derived from continuous monitoring at run time. NFF is reduced because the normal and faulty operation of the system and the interdependence between components is recorded. The introduction of comprehensibility into the models implies reduced personnel costs since less resource will be needed to train fewer personnel who'll make fewer mistakes. Finally less delay from troubleshooting failures and longer interval between the now more efficient scheduled maintenance implies reduced costs and less fuel consumption which results in reduced environmental degradation.

2 Review of Relevant Computational Intelligence Research

One of the earliest promising attempts at creating an intelligent machine that simulates the way humans solve problems was the General Purpose Problem Solver (GPS) [1], by Newell and Simon based on work from their Logic Theory Machine [2], in the late 1950s. Their method could solve some general problems but failed woefully at the simplest of tasks because of the fundamental flaw of employing a general purpose problem solving strategy. Humans use specialized domain specific knowledge to solve problems and these do not typically generalize to other fields. Being skillful in flying kites does not typically translate to being skillful in flying airplanes. This insight, despite the failure of their work, jumpstarted the development of expert or knowledge based systems. Expert systems provided some of the first successful applications of Artificial Intelligence to real world problems. DENDRAL [3] developed by Lederberg, Feigenbaum, Buchanan et al. at Stanford in 1967 was the first of these successful programs. It interpreted the mass spectra of organic compounds to determine their molecular structure and atomic constituents. Another famous and successful expert system was MYCIN [3] developed by Shortliffe at the Stanford Medical school in the early 1970s. It was used to diagnose and recommend treatment for blood infections.

As defined by Welbank [4] an expert system is "a program which has a wide base of knowledge in a restricted domain and uses complex inferential reasoning to perform tasks which a human expert could do". Despite initial successes, two major problems that militated against the widespread deployment of expert systems were

1. Their inability to autonomously learn and improve their performance.
2. The Knowledge Acquisition (KA) Bottleneck [5][6]. Every expert system is as good as the knowledge it contains (knowledge representation) and the knowledge of the experts (knowledge elicitation) used to build it. The combination of these limitations defines the KA bottleneck.

The Knowledge Elicitation Problem: Experts need to be available and willing to share their expertise with the knowledge engineer but this is not always the case. When they are willing they are often unable to easily and accurately express their implicit knowledge as rules for a machine. Sometimes experts have conflicting opinions about aspects of their knowledge. All of these define the challenge of eliciting knowledge from experts, and make this process very time consuming especially as it was done manually in the earlier days. MYCIN with only 400 rules required 100 man hours for its development. This is not acceptable for more complex systems. One approach for solving this problem is the process of knowledge discovery from databases [7]. It is believed that existing databases contain hidden knowledge that needs to be mined. Thus the field of data mining was born.

Knowledge Representation Problem: The knowledge elicited from the experts has to be translated into the form in which it is stored in the machine, validated and then transferred into the machine. This process is rife with uncertainties [8]. There is no guarantee that the knowledge elicited is exactly the same as that transferred into the system. Also experts do not think in rigid crisp logic but in gradations and degrees of concepts. Different methods used for knowledge representation have included rules in crisp and fuzzy logic, frames, semantic networks, decision trees, exemplars and objects [9][10][11][12].

Research on the solution of these problems led to enormous advances in the fields that today constitute computational Intelligence.

2.1 Computational Intelligence Methods

While there is yet no consensus on the definition of CI, many researches [13][14] agree that CI is the field of research that encompasses the following – (i) Artificial Neural Networks(ii) Fuzzy Computing (iii) Evolutionary Computing (iv) Data mining and Machine learning.

1. Artificial Neural Networks (ANN): Also known as connectionist systems develop models of the information processing in neurons for adaptive intelligent information processing in practical real world applications. The discovery of back propagation [15] in multi layer perceptrons by Rumelhart et al. [16] led to major advances in ANN research. Most ANN models are typical black box models, because they cannot provide explanations for their output. An exemption is the

ART [17] which also, unlike Back Propagation, does not suffer from catastrophic forgetting as described in the plasticity stability dilemma by Grossberg [18].

2. Fuzzy Computing: Computing model introduced by Lotfi Zadeh[19], whereby attributes are not limited to just crisp binary values but have degrees of membership that more closely mirrors the qualitative range of representations and perspectives used by humans. This model is a lot more suited to handle uncertainties and ambiguities.
3. Evolutionary Computing: Computing model based on evolutionary algorithms [20] and evolutionary programming [21] whereby a population is evaluated by a fitness function and the best individuals are chosen (survival of the fittest). These individuals reproduce (crossover) or mutate to create a new population which are re evaluated and the process continues until the desired fitness is attained.

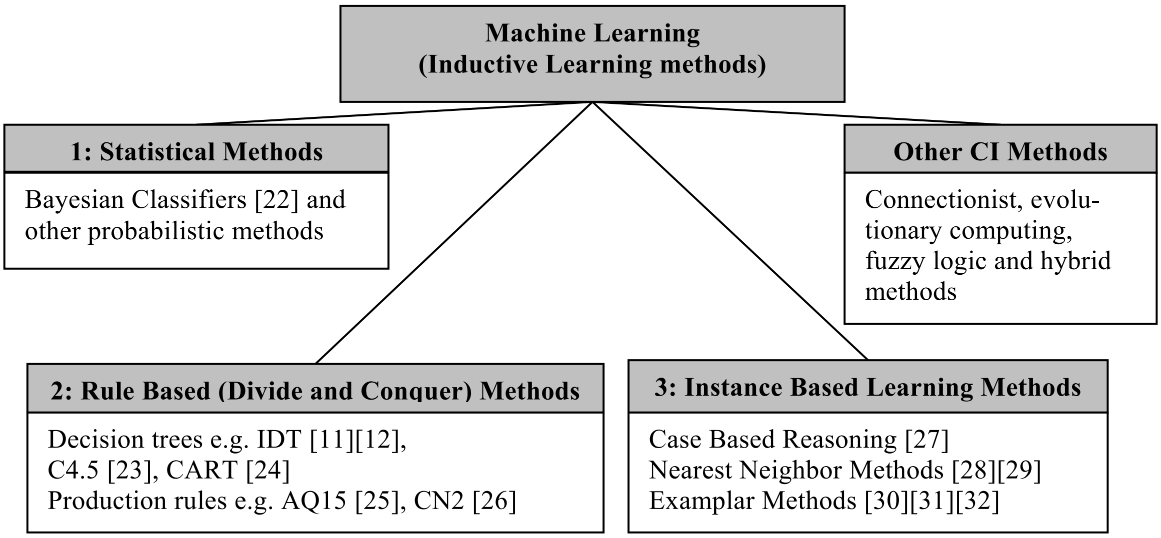

Figure 1: Major Classes Inductive Learning Methods

4. Data Mining and Machine Learning: We define machine learning as the inductive learning of rules and concepts from data and examples [33] Inductive in that we go from limited or no knowledge to general knowledge. By rules the importance of comprehensibility in the induced knowledge is underscored. While we use the terms machine learning and data mining interchangeably in this paper, technically there are differences between these terms. Holsheimer and Siebes [34] identified the difference as being in the source of data; for data mining the source is always a database. Another more important distinction describes the data mining process [35][36] with machine learning techniques as some of the methods used in the step that directly mines the data. Figure 1 above gives an overview of the major classes of the inductive learning methods.

The different classes and their methods above possess strengths and weaknesses that suit them to particular problem areas. Because of this intrinsic bias [37] it is often necessary to experiment with different algorithms before identifying the appropriate one for a given task. Another approach is to utilize hybrid methods combining different algorithms.

2.2 Hybrid Computational Intelligence Methods

The law of conservation of generalization performance prescribes that every algorithm has domains in which its performance is inferior to another algorithm [38]. This has informed the multi-strategy approach to data mining that combines the strengths and weakness of different CI methods to create new hybrid methods that perform better in complex domains [39]. A review of hybrid CI methods [40]shows how hybrids of the four different CI methods provided superior performance in diverse domains. Meta learning [41][42][43] is another approach to multi strategy learning. It applies different CI methods to different training data set and the predictions from the base learners are combined in the Meta learner to produce a more accurate final prediction.

The CI methods and approaches listed above have been successfully deployed in many different application areas that presume a centralized monolithic model [44] of intelligence. However there are particular problem situations such as the aircraft monitoring domain that require a non monolithic model of intelligence, where intelligence is both distributed in the components and centralized for knowledge common to the system as a whole. Our CHI framework is proposed for these problem situations.The CHI Framework

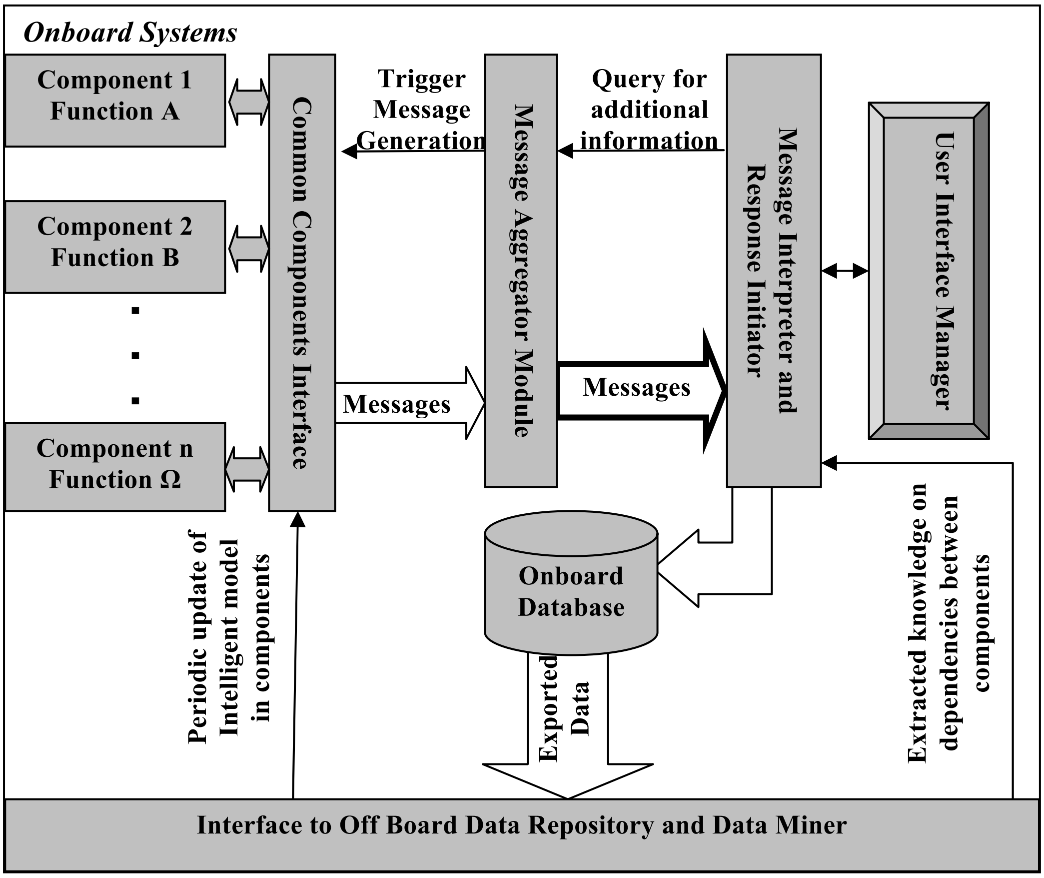

Figure 2: Overview of CHI Framework

The CHI Framework, Figure 2, employs a hierarchical, non monolithic, multi-strategy and intelligent approach to monitoring of the aircraft and its components. Each component provides a definite function like passenger entertainment or cabin pressure control. The components are monitored independently, while the probable interdependence between them is monitored at a higher layer. This interdependence is crucial to learning more about the system as a whole.

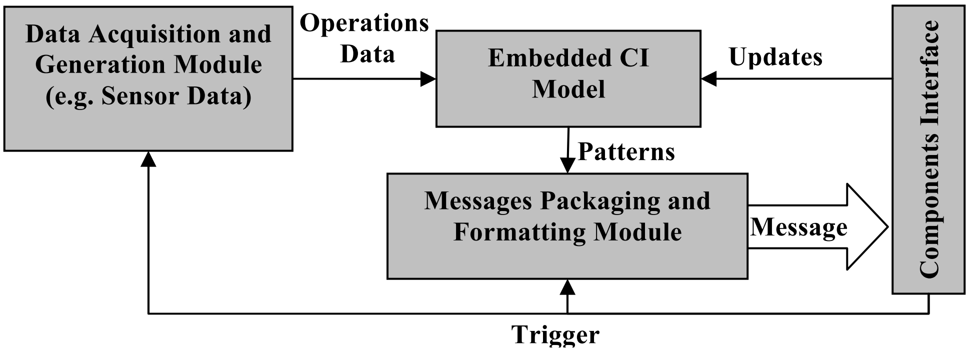

Figure 3: CHI Component Architecture

The components share a common architecture shown in Figure 3. The data acquisition and generation module (DAGM) is responsible for acquiring operations data from sensors, actuators and other devices through which the component interacts with its container system and the environment. These data serve as input to the embedded CI model in every component. The choice of the CI model to use is determined after series of tests to find the model best suited to the bias of the provided function. The output from the CI model is the pattern

$$\langle I, c, \kappa, E \rangle \tag{1}$$

Where I is the input vector from the DGAM; c is the class – the result of classifying I with the embedded CI model; $\kappa \in [0,1]$ is the degree of confidence in c; and E is the explanation justifying c and κ. The packaging module reformats the pattern in (1) as messages for the higher layers in the framework according to Table 1.

Time_Stamp	Date and Time
Component_ID	Unique Id of the component in the system
Priority_Index	Higher values mean lower priority
Message_Class	Class derived according to Table 2
Message	Same as the pattern in (1)

Table 1: Message Format

The Priority_Index P in table 1 is derived according to this equation:

$$P = \gamma C + (1-\gamma)X \tag{2}$$

Where: γ – Weight assigned to the criticality index usually 0.5; C – Criticality Index, with lower values to the more critical components. X – Severity index, a measure of the severity of messages. It is the same as the value of the message class in Table 1 and is derived with heuristics similar to that of Table 2.

We differentiate between two types of models; those with categorical class values, which we use for state monitoring and others with continuous valued output which we use for life cycle prediction. The categorical values are normalized to just two values by the packaging module, either normal or abnormal state; however the actual value is not lost because it is enclosed in the message sent. The continuous valued classes are given higher severity index, which translates to lower priority because being life cycle predictors they are not as severe as the state monitoring messages.

Pattern Class	Message_Class
Trigger Response	0
Categorical Abnormal with confidence $\kappa > 60\%$	1
Categorical Abnormal with confidence $10\% < \kappa < 60\%$	2
Categorical Abnormal with confidence $\kappa < 10\%$	3
Categorical Normal with confidence $\kappa < 90\%$	3
Categorical Normal with confidence $\kappa > 90\%$	4
Continuous	5

Tabel 2: Example Message Class Derivation.

After packaging, the messages are sent to the message aggregator (MA) module which implements a queue that ensures no message is lost sorts them according to their P value. The message with lowest P is then retrieved and sent to the message interpreter and response initiator (MIRI).

MIRI analyzes messages and initiates response based on additional knowledge of the system it possesses. The response could be any combination of the following:

- Alert the crew. The mode (e.g. alarm, blinking message) and destination (e.g. cockpit, cabin crew) of the alert reflects the severity of the message and the criticality of the components involved.
- Send query to the MA module for more information from same or other component. If information is not in the MA module, a trigger is sent to the component to retrieve it.
- Repackage the message and send to the onboard database. The contents of this database are sent to an off board database after every flight.

The embedded CI models for MIRI and in the components are derived from off board mining of data accumulated after many flights and from many airplanes. The new models are introduced as regular updates to the onboard system.

3 Discussions and Conclusion

The CHI framework is a work in progress. Currently we are developing the CI models for the components. This task is made more difficult because of the constraints of comprehensible explanations and degrees of confidence imposed by equation (1). One class of CI methods that we find promising for this purpose is the class of generalized exemplars [30]. We are developing modifications of this method that suit the bias of the functions in the aircraft monitoring domain. In the future we will explore hybrids of the generalized exemplars with other methods. The appropriate CI model for the MIRI is also currently being investigated. Another important technical challenge is in the implementation of the message aggregator module such that it ensures that no messages are lost. Different methods from queuing theory and/or parallel queues will be explored for this purpose. Use of web services and similar standards for communication and exchange of information with the databases, user interface manager and off board systems will be explored. Finally getting all of these to work together constitutes a serious technical challenge, which we aim to realize because of the physiological plausibility of our approach.

Physiological Plausibility: The motivation for our work was based on observations of the sensorimotor mechanisms of the vertebrate nervous system [44]. Consisting of the central nervous system and the peripheral nervous system, and comprising the seat of intelligence and consciousness in humans, the VNS is one of the most complex systems in the universe. It is known that the VNS organizes intelligence in a hierarchical, distributed and non monolithic fashion which informs our framework. An analogy of our CHI framework to the VNS is given below.

CHI Framework	Vertebrate Nervous System
Components	Organs
Data acquisition and generation module with embedded CI model	Nociceptors and Receptors (chemo, photo, mechano and thermo)
Packaging Module and Common Interface	Afferent Neural Pathways
Messages	Signals and their Integration
Message Aggregator Module	Dorsal Root or Spinal Ganglia
Message Interpreter and response initiator	Spinal Cord
User Interface Manager	Motor Response
Data mining process that supplies the CI models in the framework	Brain

Table 3: Analogy to the Vertebrate Nervous System

By modeling our framework on the "only example of a complex versatile system that is universally accepted as intelligent: humans" [44] and particularly on the seat of intelligence in humans which is the nervous system, we provide initial justification by physiological plausibility for this unique and intelligent approach to the monitoring of aircraft systems.

Acknowledgments

This work is supported by Airbus.

References

[1] A. Newell, J. C.Shaw and H. A. Simon. Report on a General Problem-Solving Program for a Computer. In Proceedings of the International Conference on Information Processing, pp. 256-264. Paris: UNESCO, 1960.

[2] A. Newell and H.A. Simon. The Logic Theory Machine. IRE Transactions on Information Theory, IT-2(3): 61-79, 1956.

[3] B.G. Buchanan and E.H. Shortliffe. Rule Based Expert Systems. Addison-Wesley. 1984.

[4] M. Welbank. A Review of Knowledge Acquisition Techniques for Expert Systems. British Telecom. Research Laboratories Technical Report, Ipswich England 1983.

[5] E.A. Feigenbaum. The Art of Artificial Intelligence: Themes and Case Studies Knowledge Engineering. In Intern. Joint Conf. of Artificial Intellig., pages 1014 - 1029, 1977.

[6] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. Building Expert Systems. Addison-Wesley. Reading, MA, 1983.

[7] G. Piatetsky Shapiro. Workshop on Knowledge Discovery in Real Databases. In International Conference of Artificial Intelligence, 1989.

[8] E. Santos Jr. and D. O. Banks. Acquiring Consistent Knowledge. Technical Report AFIT/EN/TR96-01, Air Force Institute of Technology, January 1996.

[9] H. Chen and K. J. Lynch. Automatic Construction of Networks of Concepts Characterizing Document Databases. IEEE Transactions on Systems, Man and Cybernetics, 22(5):885--902, September/October 1992.

[10] K. Parsaye, M. Chignell, S. Khoshafian, and H. Wong. Intelligent Databases. John Wiley and Sons, Inc., New York, NY, 1989.

[11] J.R. Quinlan. Induction of Decision Trees. Machine learning, 1(1):81-106, 1086

[12] J.R. Quinlan. Decision Trees and Decision making. IEEE Transactions on Systems, Man and Cybernetics, 20(2): 339 - 346, March/April 1990.

[13] H-J. Zimmermann, G. Tselentis, M. Van Someren, G. Dounias (Eds.). Advances in Comp. Intellig. and Learning: Methods and Applications. Kluwer Acad. Publishers, 2002.

[14] B.C. Craenen and A.E. Eiben. Computational Intelligence. Vrije Universiteit Amsterdam, Faculty of Exact Sciences, 2002.

[15] F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. Psychological Review, 65:386-408, 1958.

[16] D. E. Rumelhart, G. E. Hinton and R. J Williams. Learning Representations by Back Propagation Errors. Nature, 323:533—536, 1986.

[17] G.A. Carpenter and S. Grossberg. ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. Applied Optics, 26:4919--4930, 1987.

[18] S. Grossberg. Studies of Mind and Brain. Reidel Press, Boston, MA, 1982.

[19] L.A. Zadeh. Fuzzy Sets: Information Control, 8: 338-353, 1965.

[20] J.H. Holland. Adaptation in Natural and Artif. Systems. MIT Press, Cambr., MA, 1975.

[21] J.R. Koza. Genetic Programming – On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge, MA, 1992.

[22] W. Buntime. Learning Classification Rules using Bayes. In Proceedings of the Sixth International Workshop on Machine Learning: pages 94-98, Ithaca NY, 1989.

[23] R. Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[24] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone. CART: Classification and Regression Trees. Chapman & Hall / CRC Press, 1984.

[25] J. Hong, I. Mozetic and R. S. Michalski. AQ15: Incremental Learning of Attribute-Based Descriptions from Examples, the Method and User's Guide. UIUCDCS-F-86-949, Department of Computer Science, University of Illinois, Urbana, IL, 1986.

[26] P. Clark and T. Niblett. The CN2 Induction Algorithm. Mach. Learning, 3:261-283, 1989.

[27] A. Aamodt and E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 7(1):39-59, March 1994.

[28] T.M. Cover and P.E. Hart. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 21—27, 1967.

[29] S. Cost and S. Salzberg. A Weighted Nearest Neighbour Algorithm for Learning with Symbolic Features. Machine Learning, 10:57 -- 78, 1993.

[30] S. Salzberg. Learning with Nested Generalized Examplars. Kluwer Ac. Publ., MA, 1990.

[31] D.W. Aha, D. Kibler, and M.K. Albert. Instance-Based Learning Algorithms. Machine Learning, 6(1): 37-66, 1991.

[32] D.W. Aha and D. Kibler. Learning Representative Exemplars of Concepts; An Initial Study. In Proc, of the 4th int. workshop on Mach. Learning, pages 24-29, CA, June 1987.

[33] M. Kubat, I. Bratco. and R.S. Michalski: A Review of Machine Learning Methods, in R.S. Michalski, I. Bratco and M. Kubat (eds), Machine Learning And Data Mining – Methods and Applications, Wiley, pp. 3-69, 1997

[34] M. Holsheimer and A. Siebes. Data mining: The Search for Knowledge in Databases. Report CS-R9406, CWI, Amsterdam, The Netherlands, 1994.

[35] K. J. Cios and L. Kurgan. Trends in Data Mining and Knowledge Discovery, In: N.R Pal., L.C Jain, and N.Teoderesku (Eds.). Knowledge Discovery in Advanced Information Systems, Springer, 2002.

[36] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge Discovery and Data Mining: Toward a Unifying Framework. In Proceeding of The Second Int. Conference on Knowledge Discovery and Data Mining, pages 82--88, 1996.

[37] T.M. Mitchell. The Need for Biases in Learning Generalizations. Technical Report CBM-TR117, Computer Science Department, Rutgers University, New Brunswick, NJ, 1980.

[38] C. Schaffer. A Conservation Law for Generalization Performance. In Proceedings of the Eleventh International Conference on Machine Learning, pages 259 – 265, 1994.

[39] P. Domingos. Unifying Instance-Based and Rule-Based Induction. In Proceedings International Conference on Machine Learning, volume 24, pages 141--168, 1996.

[40] A. Tsakonas and G. Dounias. Hybrid Comp. Intelligence Schemes in Complex Domains: A Review of Selected Publications. Int. Conf. SETN-02, Thessaloniki, Greece, 2002.

[41] P. Chan and S. Stolfo. Meta-learning for Multistrategy and Parallel Learning. In Proceedings of the 2nd International Workshop on Multistrategy Learning, pages 150–165, 1993.

[42] R. Vilalta and Y. Drissi. A Perspective View and Survey of Metalearning, Artificial Intelligence Review, 18(2): 77--95, 2002.

[43] D. Wolpert. Stacked Generalization. Neural Networks Volume 5, pages 241-259, 1992.

[44] R. A. Brooks et al. Alternative essences of intelligence. In Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98). AAAI Press, 1998

[45] W. D. Willis Jr. and R. E. Coggeshall. Sensory Mechanisms of the Spinal Cord: Volume 1: Primary Afferent Neurons and the Spinal Dorsal Horn. Plenum US, January 2004.