The Use of Computational Intelligence Paradigms in Smart Software Engineering: Techniques, Applications and Challenges

Kariman Ramzy ElHelow, and Abdel-Badeeh M. Salem

***Abstract*—** Computational Intelligence (CI) is an efficient paradigm for development intelligent systems. This paradigm has resulted from a synergy between cognitive computing, fuzzy sets, Rough sets, bio-inspired computing, machine learning, computer science, engineering, statistics, mathematics, physics, psychology and social sciences. Recently, many researchers have attempted to develop CI methods and algorithms to support the decision-making in different tasks and domains. There has been a recent research in the application of CI paradigms, approaches and techniques to address software engineering(SE) problems. CI offers smart models and intelligent algorithms that can contribute greatly to design formalization and automation. In this paper we clarify many important SE issues, review some of CI techniques and their applications and also highlight challenges.

***Keywords*—** Artificial intelligence, Bio-inspired computing, Machine learning, Smart software engineering.

I. INTRODUCTION

There has been a long history of applying CI paradigms and techniques to address software engineering (SE) problems especially on tool automation. SE is a knowledge-intensive activity, requiring extensive knowledge of the application domain and itself target software. Many SE costs can be attributed to the ineffectiveness of current techniques for managing this knowledge, and artificial intelligence (AI) techniques can help alleviate this situation. In Search Based Software Engineering (SBSE), the goal is to re-formulate SE problems as optimization problems that can then be attacked with computational search [1], [2].

Kariman Ramzy ElHelow, is with the Higher Institute of Computer, King Mariout, Alexandria, King Mariout Academy, King Mariout, Alexandria, EGYPT, K.elhelowElhelow kr.elhelow@yahoo.com

Abdel-Badeeh M. Salem, is with theFaculty of Computer and Information sciences, Ain Shams University, Abbassia 11566, St.El-Khalifa El-Mamoon, Cairo, EGYPT, abmsalem@yahoo.comabsalem@cis.asu.edu.eg http://staff.asu.edu.eg/Badeeh-Salem

CI algorithms and techniques are well suited to such complex SE problems because they can replicate intelligent behavior. AI is about making machines intelligent, while software engineering is the activity of defining, designing and deploying some of the most complex and challenging systems mankind has ever sought to engineer, Though SE is one of the most challenging of all engineering disciplines [3].

On the other side, intelligent algorithms are well suited to such complex software engineering problems.

There are many areas in which computational intelligence techniques have proved to be useful in SE.

SE community has used many algorithms, methods, and techniques that have emerged from the AI community in every area of SE activities . The various ways in which CI techniques have been applied in SE reveal considerable overlaps. This paper discusses the exploiting the role of CI paradigms in the technical issues for developing smart software engineering and what are the techniques, applications and challenges[4],[5].

The rest of the paper is organized as follows:

Section2 is devoted to the advanced software engineering (SE) issues. Section 3 presents a brief overview about the well-known CI techniques and paradigms. Section 4 presents the applications of CI in different areas.

Section 5 discusses the challenges. The last section concludes the paper and illustrates the future work.

II. ADVANCED SOFTWARE ENGINEERING (SE) ISSUES

A. SE Issues in Big Data (BD) Environment

BD has become one of the most important concepts that are being studied in SE, necessitating the solution of major problems such as the collection, processing and storage of huge volume of data. BD frameworks are developed specifically to solve these problems that facilitate application developers by providing opportunities to collect, process, manage, monitor and analyze these data. Although the challenges rose from coordination of IT resources such as huge amounts of computation power, storage area, memory, and network bandwidth in a distributed manner solved by these frameworks, there still remain many SE problems in application development process.

On the other side, BD is about an enormous amount of data sets that cannot be processed by using conventional databases and conventional tools. Among the others, healthcare, telecommunication, e-commerce, social media, bank, trading, and streaming video services are a few of them. Even it can be used to enhance the business and production processes [6].

B. SE Problems and Their Relationship to Perceived Learning and Educational Tasks

The developers contributed quite a lot to solving problems with technology skills, contributing to solving problems increased learning moderately for most of the topics. Encountering problems with task management, customer expectations and customer communication affected customer satisfaction very negatively [7].

C. Harmful Problems

The worst situation is when the developers encounter problems with task management. Such problems decrease customer satisfaction very much and don't have much educational value. When such problems take place, e.g., the developers don't know clearly what they should do next, the developers may end up doing less important tasks or nothing at all. Encountering problems with customer expectations is debatable, but decrease customer satisfaction quite a lot for both roles. The same applies for the managers with task management and maintaining motivation, and for the developers with customer communication [8].

III. COMPUTATIONAL INTELLIGENCE (CI) PARADIGMS AND TECHNIQUES

CI is a group of computational models and intelligent tools that encompass elements of learning, adaptation and/or heuristic optimization. It is used to help study problems that are difficult to solve using conventional computational algorithms. Recently, emerging techniques such as swarm intelligence, artificial immune systems, deep learning, support vector machines, rough sets and others, have been added to the CI techniques [9]. Figure1 shows the interdisciplinary science of computational intelligence this section presents a brief account about the well-known CI techniques and paradigm.

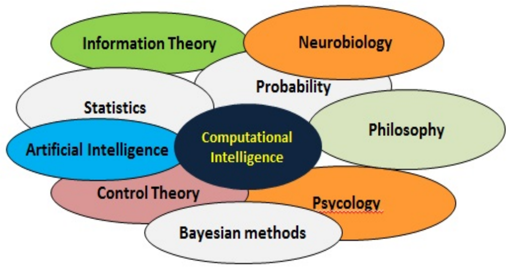

Fig.1. Interdisciplinary science of computational intelligence

A. Bio-Inspired Computational Intelligence Techniques

Figure 2 shows the most common bio-inspired computational techniques.

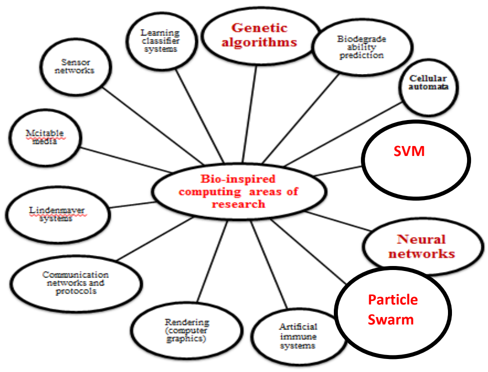

Fig. 2.Bio-inspired Computational Techniques

In what follows a brief overview about the most common used bio-inspired CI techniques, namely; artificial neural networks, genetic algorithms, particle swarm and support vector machines (SVM).

1) Artificial Neural Networks (ANN): ANN is inspired in biological models of brain functioning. They are capable of learning by examples and generalizing the acquired knowledge. Due to these abilities the neural networks are widely used to find out nonlinear relations which otherwise could not be unveiled due to analytical constraints. The learned knowledge is hidden in their structure thus it is not possibly to be easily extracted and interpreted. The architecture of the multi-layer perception determines its capacity, while the knowledge about the relations between input and output data is stored in the weights of connections between neurons. The values of weights are updated in the supervised training process with a set of known and representative values of input–output data samples [10],[11].
2) Genetic algorithms (GA): GA is random search algorithms inspired by natural genetics to find optimum solutions to problems. The basic concept is to maintain a population of chromosomes representing candidate solutions. Each chromosome in the population has an associated fitness. GAs has long been utilized in fuzzy inference systems for generating fuzzy rules and

training membership functions. There are different approaches to utilize GA in learning, the Michigan, Pittsburgh and the Iterative Rule Learning (IRL) approaches [12], [13].

3) Particle swarm optimization (PSO): PSO is as mart algorithm used for feature selection which is inspired from the natural social behavior and dynamic movements and communications of insects, birds, and fishes. The key concept of PSO is dealing with changes in velocity. The main strength of the PSO is its fast convergence compared with other global optimization algorithms like genetic algorithms.
Initially in PSO algorithm, the population is initialized by a user.

4) Support Vector Machines (SVM): SVM are an efficient learning paradigm for classification and regression problems [5]. A key feature of SVM is that the classification complexity does not depend on the dimensionality of the features space and that the sensitivity to the number of features is relatively low. Therefore, it can learn a larger set of descriptors and also might be able to scale the number of features and classes in a better way than neural networks. The main advantage of SVM lies in the structure of the learning algorithm which consists of a constrained quadratic optimization problem (QP), thus avoiding the local minima drawback of NN. [14],[15].

B. Vagueness Computational Intelligence

1) Rough Set (RS): RS theory was proposed as a new approach to vague concept description from incomplete data [16]. This theory is one of the most useful techniques in many applications such as pharmacology, engineering, banking, medicine, and market analysis. This theory provides a powerful intelligent tool to reveal and discover important structures and hidden patterns in data and to classify complex objects. One of the main advantages of RS theory is that it does not need any preliminary or additional information about data. A RS is a formal approximation of a vague concept by a pair of precise concepts, called lower and upper approximations. RSs handle uncertainty by computing the lower and upper approximations.The concept of crisp equivalence class is the basis for RS theory. A crisp equivalence class contains samples from different output classes [17].

2) Fuzzy Sets (FS): Fuzzy inference systems have been successfully applied in pattern classification, expert systems, decision analysis, and manufacturing and computer vision. Traditionally, the rules in a fuzzy inference system are generated from expert knowledge. If no expert knowledge is available, the usual system is to identify and train fuzzy membership functions in accordance with the data clusters in a training set [18].

3) Fuzzy-Rough Sets: Fuzzy-rough sets are a combination of fuzzy and rough sets. This integration is very useful for decision making processes where both vagueness and indiscernibility are present. A fuzzy similarity relation replaces an equivalence relation in rough sets to form the fuzzy-rough sets. Each fuzzy cluster represents an equivalence class containing patterns from different output classes. The definite and possible members of the output class are identified using the lower and upper approximations of the fuzzy equivalence classes .

Shen et al. [19] proposed a smart classifier that integrates a fuzzy rule induction algorithm with a rough set assisted feature reduction method. The classifier is tested on two problems, the urban water treatment plant problem and algae population estimation. This method reduces the decision table in both vertical and horizontal directions (both the number of features and feature dimensionality are reduced). A robust feature evaluation and selection algorithm, using a different model of fuzzy-rough sets namely soft fuzzy-rough sets, is provided in. This method is more effective in dealing with noisy data[20].

IV. APPLICATIONS OF CI PARADIGMS

A. CI for Healthcare

CI techniques could be successfully employed to tackle various problems in healthcare and life sciences, such as gene selection, DNA fragment assembly, gene expression , protein sequence classification, protein function prediction and its structure and medical diagnoses [21],[22].

At our research unit at Ain Shams, a rough set-based medical system for mining patient data for predictive rules to determine thrombosis disease was developed [23]. This system aims to search for patterns specific/sensitive to thrombosis disease. This system reduced the number of attributes that describe the thrombosis disease from 60 to 16 significant attribute in addition to extracting some decision rules, through decision applying decision algorithms, which can help young physicians to predict the thrombosis disease. In our research group we developed a hybrid classifier that integrates the strengths of genetic algorithms and decision trees. The algorithm was applied on a medical database of 20 MB size for predicting thrombosis disease [9]. The results show that our classifier is a very promising tool for thrombosis disease prediction in terms of predictive accuracy.

B. CI for Troubleshooting Software Bugs

This paradigm for incorporating AI in the recent software troubleshooting process that can reduce costs

[24]. In this respect paradigm, this model integrates the following three AI technologies:

1) machine learning: learning from source-code structure, revisions history and past failures, and software components that contain bugs,
2) automated diagnosis: identifying the software components that need to be modified in order to fix an observed bug, and
3) Automated planning: planning additional tests when such are needed to improve diagnostic accuracy.

C. CI for Ambient Intelligence (AmI)

AmI promotes interdisciplinary research encompassing the technological, scientific and artistic fields creating a virtual support for embedded and distributed intelligence [25]. From our analysis of the recent published research [26],[27], one can summarize the main advantages of AmI in the following;

1) AmI supports the design of the next generation of intelligent systems and introduces novel means of communication between human, machine, and the surrounding environment (man-made objects).
2) AmI provides basic criteria to develop intelligent environments where electronic devices can be so tiny, virtually invisible to us .

D. CI for Energy-Efficient Deployment of Intelligent Mobile Sensor Networks

This paradigm discusses a set of distributed energy-efficient deployment algorithms for mobile sensors and intelligent devices that form an ambient intelligent network. The integration of several mobile sensors and/or intelligent devices which need to communicate with each other is an essential constraint of intelligence environments that would use smart infrastructures to improve the quality of life and safety in emergency situations. Organization and optimization of network resources are fundamental to provide ubiquitous communication for a longer duration in large-scale networks [28], [29].

E. CI for Computer Music Composition

In this paradigm, there is a great challenge related to the creativity. Thus, the term of "Artificial Creativity" or "Computational Intelligent Creativity" is introduced.

This term aims to partially, or fully, automating the process of art works synthesis (for example music composition). Computer music composition became highly correlated with the term "intelligent algorithmic composition" which is a style of artificial creativity that, from its name, uses algorithms in automating the music composition process. Recently there is a big research computational intelligence algorithmic composition the various techniques used in these paradigms such as: grammars, Neural Nets, Swarm and evolutionary algorithms.

V. DISCUSSIONS

The field of smart SE is a highly dynamic in terms of research and knowledge, and it depends heavily upon the experience of experts for the development and advancement of its methods, techniques and tools.

Many AI systems are non-trivial to build and thus require careful problem analysis, modeling, system of system design and engineering, and test and evaluation. Software engineers and developers may also learn these skills in SE - it depends on the SE curriculum.

Computational intelligence and machine learning are enabling less technical domain experts to develop technologies using either natural language or visual interfaces. Smart programming assistants can offer just-in-time support and recommendations; machine learning models extrapolate important features and patterns in data. Software is code written in the form of “neural network weights” not by humans but by machine learning methods such as back propagation and stochastic gradient descent.

Updating models entails retraining algorithms with new data, which will change how the model will behave and perform. The critical limitations of many machine learning approaches is our human inability to fully comprehend how such complex systems work, leading them to appear to us as "black boxes".

Using CI techniques such as neural networks and deep learning, software engineer does not give the computer rules for how to make decisions and take actions. A smart machine learning (SML) model can deduce from data what features and patterns are important, without a human explicitly encoding this knowledge.

The outputs of SML models can even surprise humans and highlight perspectives or details we haven't thought of ourselves. Thus, the most profound impact of AI on computer programming is the unraveling of how humans perceive, define, and execute software development.

Recent breakthroughs in software engineering have helped AI capabilities to be effectively reused through cloud computing and internet of things (IoT) solutions.

AI will play a key role in the design, creation and testing of software. AI can also help in code generation. Further that if an AI software system is given a business requirement in natural language, it can write the code to implement it — or even come up with its own idea and write a program for it.

VI. CONCLUSIONS AND FUTURE WORK

Computational intelligence paradigms can be used to examine code and automatically optimize it for interpretability and performance. Reliable estimates require deep expertise, understanding of context, and familiarity with the implementation team.

Many new applications and research fields of interest to both disciplines will develop covering knowledge-based systems for learning software organizations, the development of computational intelligence and knowledge discovery techniques for software artifacts.

Moreover CI and knowledge engineering would have a better significance in a real time sharing, if it includes (uses) the SE. Because SE can enable it to analyze, design, manage and abstract from a real world and make easily understandable and usable. The combination of CI and SE are to achieve of optimal path towards market economy. To meet user's various requirements, a series of innovations in software development have been carried out.

REFERENCES

[1] Mark Harman, "The Role of Artificial Intelligence in Software Engineering", First International Workshop on Realizing AI Synergies in Software Engineering (RAISE), 2012.

[2] Xinyu Feng, Markus Müller-Olm, and Zijiang Yang, "Dependable software engineering, theories, tools, and applications", Proceedings of the third International Symposium, SETTA 2017, Beijing, China, October 23-25, 2017.

[3] Chang S. K. , "Handbook of Software Engineering & Knowledge Engineering", World Scientific connecting great minds , Singapore, Vol. 2, 2002, pp. 21-53.

[4] Wang Y. , and Patel D. ," introduction: comparative software engineering: review and perspectives", Annals of Software Engineering, Vol. 10, Issue 1–4, pp. 1–10, 2000.

[5] Tao Xie," Intelligent software engineering: synergy between AI and software engineering", Conference: the 11th Innovations in Software Engineering Conference, India, February, 2018.

[6] Ziya Karakaya , "Software engineering issues in big data application development", proceeding of International Conference on Computer Science and Engineering (UBMK) , Antalya, Turkey , October, 2017.

[7] Carlo Ghezzi, and Dino Mandrioli, "The challenges of software engineering education", the International Conference on Software Engineering, USA, December 2006, pp. 115-127.

[8] Jari Vanhanen, Casper Lassenius, "Software engineering problems and their relationship to perceived learning and customer satisfaction on a software capstone project", A1 Journal article-refereed, Vol. 137, No. 17, 1 Mar. 2018.

[9] Lakhmi C. Jain, Shing Chiang Tan, and Chee Peng Lim, "An Introduction to Computational Intelligence Paradigms", Computational Intelligence book series (SCI, Vol. 137) 2008, pp.1-23.

[10] Harsh Kukreja, Bharath N. , Siddesh C. S., and Kuldeep S., "An introduction to artificial neural network", International Journal of Advance Research and Innovative Ideas in Education, Vol.1 Issue.5, 2016.

[11] Frank Landman, "Everything you need to know about artificial neural networks", IoT Solution Builder Conference, USA, 25 Jan, 2019.

[12] M. Harman, "Software engineering meets evolutionary computation," IEEE Computer, Vol. 44, No. 10, , Oct., pp. 31–39, 2011.

[13] A. Arcuri, and X. Yao, "A novel co-evolutionary approach to automatic software bug fixing", in Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Hong Kong, China: IEEE Computer Society, June 1-6, 2008, pp. 162–168,

[14] Theodoros Evgeniou, "*Support Vector Machines: Theory and Applications*", book Machine Learning and Its Applications: Advanced Lectures, January, 2001, pp.249-257.

[15] Chih-Wei Hsu, and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines", IEEE Transactions on Neural Networks, Vol. 13, NO. 2, MARCH, 2002.

[16] Z. Pawlak, "Rough sets", International Journal of Computer and Information Sciences, Vol.11, No. 5, 1982.

[17] Zdzislaw Pawlak, "Vagueness and uncertainty: a rough set perspective", Computational Intelligence an International Journal, Vol.11, Issue2, pp. 227-232, May, 1995.

[18] Lee, Jonathan, "Software Engineering with Computational Intelligence", Fuzziness and Soft Computing book series (STUDFUZZ), Vol. 121, 2003.

[19] Yiyu Yao, "A Comparative study of fuzzy sets and rough sets", Information Sciences (INFORM SCIENCES), Vol.109, Issues 1-4, August, 1998.

[20] Qinghua Zhang, Qin Xie , Guoyin Wang, " A survey on rough set theory and its applications", CAAI Transactions on Intelligence Technology, Vol. 1, Issue 4, pp. 323-333, October, 2016.

[21] Xinyu Feng, Markus Müller-Olm, and Zijiang Yang, "Dependable software engineering, theories, tools and applications", Proceedings of 4th International Symposium, SETTA 2018, Beijing, China, September 4-6, 2018.

[22] Fei Jiang, Yong Jiang, and HuiZhi, "Artificial intelligence in healthcare: past, present and future", Journal of Neurology, Neurosurgery & Psychiatry (BMJ Journals) , Vol. 2, Issue 4, 2017.

[23]Abdel-Badeeh M. Salem, " Towards of intelligence education and learning", IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 12-14 Dec., 2015.

[24] Amir Elmishali, Roni Stern, and Meir Kalech , "An artificial intelligence paradigm for troubleshooting software bugs", Engineering Applications of Artificial Intelligence, Vol. 69, pp. 147-156, March 2018.

[25]P. Remagnino, G. L. Foresti, "Ambient intelligence: a new multidisciplinary paradigm", IEEE Transactions on Systems, Vol. 35 Issue 1, Jan., 2005.

[26]Hani Hagras, Martin Colley, Victor Callaghan, and Graham Clarke, "Creating an ambient-intelligence environment using embedded agents, Intelligent Systems", IEEE (IEEE INTELL SYST), Vol. 19, No. 6, 12 - 20 December, 2004.

[27]Hakan Duman, Hani Hagras, Vic Callaghan, "Intelligent association exploration and exploitation of fuzzy agents in ambient intelligent environments", Journal of Uncertain Systems, Vol.2, No.2, pp.133-143, 2008 .

[28]L. C. Chen, "Energy-efficient deployment of intelligent mobile sensor networks", IEEE (TSMCA 2005) IEEE Transactions on Systems, Man, and Cybernetics, Vol. 35, Issue 1, Nov. 3, 2006.

[29]RabiaIram, Muhammad Irfan Sheikh, Sohail Jabbar, and Abid Ali Minhas, "Computational intelligence based optimization in wireless sensor network", *International Conference on Information and Communication Technologies*, Karachi, Pakistan, 2011.