1. Record Nr. UNINA9910254257203321

Titolo	Computational Intelligence and Quantitative Software Engineering / / edited by Witold Pedrycz, Giancarlo Succi, Alberto Sillitti
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2016
ISBN	3-319-25964-4
Edizione	[1st ed. 2016.]
Descrizione fisica	1 online resource (IX, 207 p. 41 illus., 26 illus. in color.)
Collana	Studies in Computational Intelligence, , 1860-949X ; ; 617
Disciplina	006.3
Soggetti	Computational intelligence Artificial intelligence Software engineering Computational Intelligence Artificial Intelligence Software Engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di contenuto	Intro -- Preface -- Contents -- 1 The Role of Computational Intelligence in Quantitative Software Engineering -- 1 Introduction---Software Development and the Art of Cappuccino -- 2 Persistent Problems in Software Development -- 3 Uncertainty -- 4 Irreversibility -- 5 Complexity -- 6 Handling Uncertainty, Irreversibility, and Complexity 2026 and Cappuccino! -- 7 The Pivotal Role of Computational Intelligence in Quantitative Software Engineering -- 8 Conclusions -- References -- 2 Computational Intelligence: An Introduction -- Abstract -- 1 Introduction -- 2 Computational Intelligence: An Agenda of Synergy of Algorithms of Learning, Optimization and Knowledge Representation -- 3 Neural Networks and Neurocomputing -- 4 Evolutionary and Biologically Inspired Computing: Towards a Holistic View at Global Optimization -- 5 Information Granularity and Granular Computing -- 6 Formal Platforms of Information Granularity -- 6.1 Information Granules of Higher Type and Higher Order -- 6.2 Hybrid Models of Information Granules -- 7 The Concept of Information Granulation-Degranulation -- 8 Clustering

as a Means of Design of Information Granules -- 8.1 Unsupervised Learning with Fuzzy Sets -- 8.2 Fuzzy C-Means as an Algorithmic Vehicle of Data Reduction Through Fuzzy Clusters -- 8.3 Knowledge-Based Clustering -- 9 Computational Intelligence and Software Engineering -- 10 Conclusions -- References -- 3 Towards Benchmarking Feature Subset Selection Methods for Software Fault Prediction -- Abstract -- 1 Introduction -- 2 Related Work -- 3 Feature Subset Selection (FSS) Methods -- 3.1 Information Gain (IG) Attribute Ranking -- 3.2 Relief (RLF) -- 3.3 Principal Component Analysis (PCA) -- 3.4 Correlation-Based Feature Selection (CFS) -- 3.5 Consistency-Based Subset Evaluation (CNS) -- 3.6 Wrapper Subset Evaluation (WRP) -- 3.7 Genetic Programming (GP).
4 Experimental Setup -- 5 Results and Analysis -- 6 Validity Evaluation -- 7 Conclusions -- References -- 4 Evolutionary Computation for Software Product Line Testing: An Overview and Open Challenges -- Abstract -- 1 Introduction -- 2 Background -- 2.1 SPL Foundations---Feature Models and Running Example -- 2.2 Basics of Evolutionary Algorithms -- 3 Overview of SPL Testing -- 4 Combinatorial Interaction Testing for Software Product Lines -- 4.1 Basic Terminology -- 4.2 SPL Genetic Solver (SPLGS) -- 4.3 State of the Art CIT for SPL Testing -- 5 Multi-objective SPL Testing -- 5.1 Multi-objective Optimization Formalization -- 5.2 An Example Scenario -- 5.3 Computation of Exact Pareto Fronts -- 5.4 Sate of the Art in Evolutionary Multi-objective Optimization for SPL Testing -- 6 Evolutionary Testing of SPLs in Practice -- 7 Open Challenges and Questions -- 8 Conclusions -- Acknowledgments -- References -- 5 Metaheuristic Optimisation and Mutation-Driven Test Data Generation -- Abstract -- 1 Introduction -- 2 Test Data Generation -- 3 Mutation Analysis -- 4 Metaheuristic Optimisation -- 5 Using Metaheuristic Optimisation to Kill Mutants -- 5.1 Fitness Functions Based on Mutation Analysis -- 5.2 Hill Climbing -- 5.3 Evolutionary Optimisation -- 5.4 Swarm Intelligence -- 5.5 Comparing Metaheuristic Techniques for Killing Mutants -- 6 Conclusions -- References -- 6 Measuring the Utility of Functional-Based Software Using Centroid-Adjusted Class Labelling -- Abstract -- 1 Introduction -- 2 Functional Programming -- 2.1 Functions and Java 8 -- 3 Pattern Classification -- 3.1 Design and Validation -- 3.2 Performance Assessment -- 3.3 Classifiers -- 4 Software Attributes and Measuring Utility -- 4.1 Metrics Used as Features -- 5 Adjusting Design Class Labels -- 5.1 Robust Location and Dispersion Measures -- 6 Experiment Design -- 6.1 Software System.
6.2 Quality Assessment by Architect -- 7 Results and Discussion -- 8 Conclusion -- Acknowledgment -- References -- 7 Toward Accurate Software Effort Prediction Using Multiple Classifier Systems -- Abstract -- 1 Introduction -- 2 Classifiers -- 2.1 Logistic Discrimination -- 2.2 k-Nearest Neighbour -- 2.3 Artificial Neural Network -- 2.4 Decision Trees -- 2.5 Naive Bayes Classifer -- 3 Multiple Classifier System Architectures -- 3.1 Static Parallel -- 3.2 Multi-stage -- 3.3 Dynamic Classifer Selection -- 3.4 Classifier Ensemble -- 4 Experimental Design -- 5 Experimental Results -- 6 Conclusion -- Acknowledgments -- References -- 8 Complex Fuzzy Logic Reasoning-Based Methodologies for Quantitative Software Requirements Specifications -- Abstract -- 1 Introduction -- 2 Uncertainty and the Software Development Process -- 3 Requirements Specification via Quality Function Deployment -- 4 Literature Review -- 5 Complex Fuzzy Systems -- 5.1 Complex Fuzzy Class -- 5.2 Degree of Membership of Order {\varvec N} -- 5.3 Generalized Complex Fuzzy Logic -- 5.4 Complex Fuzzy Propositions and Connectives Examples -- 5.5 Complex Fuzzy Inference Example -- 6 Conclusions -- References -- 9 Possibilistic Assessment of Process-

Related Disclosure Risks on the Cloud -- 1 Introduction -- 2 Related Work -- 3 Elements of Possibility Theory -- 3.1 Possibility Distributions -- 3.2 Possibility and Necessity -- 3.3 Possibility Propagation in Risk Assessment -- 3.3.1 Input Variables and Their Relationships -- 3.3.2 Reliability Propagation -- 3.3.3 Risk Assessment -- 4 Process Model -- 4.1 Process Model Assumptions -- 4.2 Garbling Outsourcing Scheme -- 5 Methodology for Disclosure-Risk Assessment in Cloud Processes -- 5.1 The Threat Space -- 5.2 Possibilistic Model of Information Disclosure -- 5.2.1 The Shapley Value -- 5.3 Impact Assessment by Value of Information Analysis.
5.3.1 Value of Information Analysis -- 5.3.2 Possibilistic Value of Information -- 5.4 The Overall Methodology -- 6 A Cloud-Based Case Study -- 6.1 The Cloud Process Model -- 6.2 Possibilistic Assessment of Likelihood and Impact -- 6.3 Sample Assessments -- 6.4 Mitigating Disclosure Risk -- 7 Conclusions -- Acknowledgements -- References.

Sommario/riassunto

In a down-to-the earth manner, the volume lucidly presents how the fundamental concepts, methodology, and algorithms of Computational Intelligence are efficiently exploited in Software Engineering and opens up a novel and promising avenue of a comprehensive analysis and advanced design of software artifacts. It shows how the paradigm and the best practices of Computational Intelligence can be creatively explored to carry out comprehensive software requirement analysis, support design, testing, and maintenance. Software Engineering is an intensive knowledge-based endeavor of inherent human-centric nature, which profoundly relies on acquiring semiformal knowledge and then processing it to produce a running system. The knowledge spans a wide variety of artifacts, from requirements, captured in the interaction with customers, to design practices, testing, and code management strategies, which rely on the knowledge of the running system. This volume consists of contributions written by widely acknowledged experts in the field who reveal how the Software Engineering benefits from the key foundations and synergistically existing technologies of Computational Intelligence being focused on knowledge representation, learning mechanisms, and population-based global optimization strategies. This book can serve as a highly useful reference material for researchers, software engineers and graduate students and senior undergraduate students in Software Engineering and its sub-disciplines, Internet engineering, Computational Intelligence, management, operations research, and knowledge-based systems.