UNIVERSITÀ degli STUDI di NAPOLI «Federico II»
PhD program in COMPUTATIONAL INTELLIGENCE

PhD in Computational Intelligence *courses catalogue*

- 40th cycle, and till active cycles –

(last updated on October 10th, 2024)

- **Algorithms and Applications for Artificial Intelligence** — prof. Francesco Piccialli
- **Artificial Intelligence in Medical Applications** — prof. Daniel Riccio
- **Computational Inference and Statistical Validation** — profs. Roberta Siciliano e Antonio D'Ambrosio
- **Computational Intelligence for Big Data** — prof. Giuseppe Longo
- **Computational Intelligence for Medical Imaging** — prof. Giovanni Mettivier
- **Computational Thermodynamics** — profs. Dario Alfè e Andrea Zen
- **Embedded System Design: A Practical Approach to Architectures, Configuration and Coding Techniques** — prof. Mario Barbareschi
- **Evolutionary Computation and Applications** — prof. Autilia Vitiello
- **Fuzzy models, fuzzy systems, and approximate reasoning techniques** — prof. Ferdinando Di Martino
- **Game Design and Development** — prof. Marco Faella
- **Game Engines and Interactive Experience** — prof. Antonio Origlia
- **High Performance Parallel Computing** — prof. Marco Lapegna
- **Internet of Everything: Principles** — prof. Edoardo Giusto
- **Introduction to parallel architectures and parallel programming** — prof. Alessandro Cilardo

- **Machine Learning for Science and Engineering Research** — **proff. Anna Corazza, Roberto Prevete, Carlo Sansone**
- **Methods for Artificial Intelligence** — **prof. Silvia Rossi**
- **Natural Language Processing** — **prof. Francesco Cutugno**
- **Numerical Methods for Data Analysis** — **prof. Salvatore Cuomo**
- **Quantum Computational Intelligence** — **prof. Giovanni Acampora**
- **Using Deep Learning Properly** — **dr. Andrea Apicella**

Algorithms and Applications for Artificial Intelligence	
Lecturer	Prof. Francesco Piccialli
	francesco.piccialli@unina.it
Credits	6
Course length (in hours)	48
Prequisites	A basic knowledge of the Python language
Description:	
The course aims to provide both a theoretical and practical introduction to Artificial Intelligence, through methodologies and tools for data analysis and processing using Machine and Deep Learning techniques, in order to improve the effectiveness and @meliness of decision-making processes. The course aims to provide students with the knowledge and methodological tools to analyze and complete a data-driven problem	

Artificial Intelligence in Medical Applications	
Lecturer	Prof. Daniel Riccio
	daniel.riccio@unina.it
Credits	6
Course length (in hours)	30
Description:	
The main objective of the course is to provide the student with an overview of the currently most promising Artificial Intelligence techniques in Medical Imaging. Starting with an introduction to artificial intelligence, the first part of the course presents some of the most common features and classifiers used before the advent of Deep Learning. The second part focuses on the different deep architectures introduced for solving challenging problems in medical Imaging. The course also provides an experimental part in Matlab, where the student can experiment with methods and techniques applied to real problems. The course consists of 15 lectures of 2 hours each.	

Computational Inference and Statistical Validation	
Lecturer	Proff. Roberta Siciliano and Antonio D’Ambrosio
	roberta.siciliano@unina.it
Credits	3 – 6
Course length (in hours)	12

Prequisites	Foundations of mathematics, statistics and computer programming
Description:	
The course introduces the principles and techniques of statistical inference and validation for computational models. It covers the fundamental concepts of probability theory, hypothesis testing and statistical modelling. Students will learn how to apply statistical inference techniques to validate computational model and analyze their performance. The course covers the basic of popular statistical inference methods, including maximum likelihood estimation, Bayesian inference, and hypothesis testing and non-parametric statistics. The course emphasizes the practical aspects of statistical inference and validation, including the design and implementation o0f statistical models, the analysis of of their behaviour and performance, and their application to real-world problems. Students will learn how to use statistical software tools, such as R and Python, to analyze and validate computational models. Overall, the course provides students with a solid understanding of computational inference, validation and non-parametric statistics and their applications. By the end of the course students will hav gained practical experience in designing, implementing and evaluating statistical models for various problems.	

Computational Intelligence for Big Data	
Lecturer	Prof. Giuseppe Longo
	giuseppe.longo@unina.it
Credits	3 – 6
Course length (in hours)	15 - 30
Prequisites	Foundations of computer programming, foundations of mathematics
Description:	
The course introduces the principles and techniques of computational intelligence that can be used to process and analyze largescale datasets. It covers the fundamental concepts of big data processing, including data preprocessing, feature selection, dimensionality reduction, and distributed computing. Students will learn how to apply computational intelligence algorithms to big data problems, such as clustering, classification, regression, and anomaly detection. The course covers the basics of popular algorithms, including fuzzy logic, evolutionary computation, artificial neural networks, and deep learning. The course emphasizes the practical aspects of computational intelligence for big data, including the design and implementation of algorithms, the analysis of their behaviour and performance, and their application to real-world problems. Students will learn how to work with big data tools and technologies, such as Hadoop, Spark, and NoSQL databases. Overall, this course provides students with a solid understanding of computational intelligence for big data and its applications. By the end of the course, students will have gained practical experience in designing, implementing, and evaluating computational intelligence algorithms for big data problems.	

Computational Intelligence for Medical Imaging	
Lecturer	Prof. Giovanni Mettivier
	giovanni.mettivier@unina.it/mettivier@na.infn.it
Credits	2-3
Course length (in hours)	10-15
Prequisites	Basic programming knowledge in Python and/or Matlab
Description:	
The role of medical image computing and machine learning in healthcare. DICOM format. Medical imaging techniques: X-ray imaging, Computed tomography, Magnetic Resonance and echography. Medical Image Analysis, Segmentation and registration. Medical Image computing. Artificial Intelligence. Neural Networks. Convolutional Neural Networks. Features and Classes. Radiomics. Applications in Radiology. Clinical Decision support. Metrics: Sensitivity, Specificity, Area Under Curve, ROC curve and	

accuracy. Quality, regulatory and Ethical Issues, Practical use cases: Breast cancer, Clinical trials, Neurological Diseases.

Computational Thermodynamics	
Lecturers	Profs. Dario Alfè / Andrea Zen
	dario.alfe@unina.it, andrea.zen@unina.it
Credits	3
Course length (in hours)	15
Prerequisites	General knowledge of the following topics: Basics of parallel architectures, computational cost of algorithms, fundamentals of matrix algebra (matrix product algorithm)
Description:	
This is a course of the Laurea Magistrale in Physics, with some contents that may be relevant to the PhD in Computational Intelligence. In particular, in the context of machine learning (ML) methods, it is important to provide accurate training sets for the ML potentials. We propose to use lectures on Density Functional Theory (DFT), and related computer practicals, together with additional material (not currently covered in the course) on Quantum Monte Carlo (QMC) methods. DFT and QMC are two electronic structure methods that solve the Schrödinger equation for molecules and solids, at different levels of approximation, and are usually taken as reference for ML training.	

Embedded System Design: A Practical Approach to Architectures, Configuration and Coding Techniques	
Lecturer	Prof. Mario Barbareschi
	mario.barbareschi@unina.it
Credits	3 - 6
Course length (in hours)	15 - 30
Prerequisites	Basic knowledge of C programming language; basic knowledge of computer architectures
Description:	
This module offers attendees the opportunity to delve into the intricacies of ARM-based microcontroller utilization, a cornerstone in modern embedded system design. Through its comprehensive program, the course aims to equip participants with a robust insight about Cortex-M core, programming principles, alongside fundamental concepts in C-Assembly optimization. Furthermore, students will engage in a practical exploration of designing practical applications, serving as a compelling case study to reinforce theoretical foundations. Participants in the course will receive embedded boards featuring processors based on the ARM Cortex M3 architecture	

Evolutionary Computation and Applications	
Lecturer	Prof. Autilia Vitiello
	autilia.vitiello@unina.it
Credits	3 - 6
Course length (in hours)	15 - 30
Prerequisites	Basic concepts of computer science
Description:	

Evolutionary computation is a subfield of the computational intelligence which includes a group of problem-solving techniques whose basic principles rely on the theory of biological evolution. Evolutionary computation methods are characterized by high performance in a wide range of problem settings. The goal of the course is to give an overview of the best-known evolutionary algorithms and show practical application examples in the scientific and engineering fields.

Fuzzy models, fuzzy systems, and approximate reasoning techniques	
Lecturer	Prof. Ferdinando Di Martino
	ferdinando.dimartino@unina.it
Credits	6
Course length (in hours)	30
Prerequisites	Basic knowledge of the Python language.
Description:	
The course provides the theoretical and practical foundation for understanding and implementing models and approaches that make use of fuzzy logic and approximate reasoning in data analysis and modeling of complex systems. Topics that will be covered will include fuzzy models in data analysis, unsupervised learning techniques with the implementation of fuzzy clustering models. Supervised learning techniques with the implementation of classification and regression models, fuzzy rule-based systems.	

Game Design and Development	
Lecturer	Prof. Marco Faella
	marco.faella@unina.it
Credits	5
Course length (in hours)	48
Prerequisites	//
Description:	
Objectives: The course aims at providing students with the following abilities: • distinguishing various types of computer games • describing the main aspects of the game experience, and their relationships with design objectives and techniques • designing real-time interactive graphical applications • using a 2D rigid-body physics simulation library • programming with sensors on mobile devices • deploying the appropriate AI component to achieve a design objective • building a game prototype from scratch Program: History and types of videogames. Elements of Game Design: design objectives and techniques. Mobile operating systems and programming environments. Introduction to game engines. 2D graphics programming. Sound and music programming. Using a 2D rigid-body physics simulation library. Interfacing with a touch-screen and motion/localization sensors. High-performance programming techniques. Architectures and patterns for games. AI algorithms and techniques for games.	

Guided development of a game prototype.

Game Engines and Interactive Experience	
Lecturer	Prof. Antonio Origlia
	antonio.origlia@unina.it
Credits	5
Course length (in hours)	48
Prerequisites	//
Description:	
Objectives: By the end of the course, students are expected to possess the following set of skills: • Describing the typical structure and the main services offered by modern game engines • Extending a game engine with custom or third-party features • Employing the basic notions of computational psychology to establish and maintain engagement • Analyzing and designing games as means of communication or persuasion • Designing game control systems coherently with the intended experience • Using advanced 3D techniques to represent the internal state of the game • Optimizing media contents to reduce the computational load Program: Internal structure of a game engine. C++ basics and interfaces development with industry grade game engines (e.g. Unreal Engine). Formal representation of emotional states and their influence in decision making. The impact of storytelling in designing control and interface systems. 3D modelling techniques optimized for real-time rendering. Guided development of a desktop game	

High Performance Parallel Computing	
Lecturer	Prof. Marco Lapegna
	marco.lapegna@unina.it
Credits	4-5
Course length (in hours)	20-25
Prerequisites	General knowledge of the following topics: Basics of parallel architectures, computational cost of algorithms, fundamentals of matrix algebra (matrix product algorithm)
Description:	
The course aims to provide students with in-depth methodologies and software tools necessary for the development and analysis of high-performance algorithms in the main parallel computing environments (shared memory, distributed memory, GPU), with special attention to the performance issues: • Introduction to high performance parallel computing: the problem of data access, block algorithms for linear algebra. • Issues related to the parallel computing: general ideas, performance metrics, programming models, software environments.	

- The shared memory programming model. Algorithms and software for matrix product in a multicore environment.

- The distributed memory programming model. Algorithms and software for matrix product in a distributed memory environment

- The CUDA programming model. Algorithms and software for matrix product in a GPU environment.

- Putting it all together: hybrid parallel computing. Existing software for High Performance Parallel Linear Algebra.

Introduction to parallel architectures and parallel programming	
Lecturer	Prof. Alessandro Cilardo
	acilardo@unina.it
Credits	2
Course length (in hours)	16
Prerequisites	First-level knowledge of computer architecture and C/C++ programming
Description:	
Introduction to modern computer architectures and parallel architectures. Overview of High-performance Computing (HPC) system architecture: memory, storage, interconnect facilities. Introduction to parallel computing. Internal parallelism (pipelining, superscalar execution), explicit single-processor parallelism (simultaneous multithreading, vector extensions), multi-/many-core parallelism. Programming models for parallel computing. Multithreading. Communication and synchronization primitives. Examples in C/C++, Introduction to OpenMP and practical exercises. Quick overview of message passing models. Overview of GPU architectures and programming. The course will entail an introduction to the Internet of Everything concept, with related applications and trends. The course will highlight technical requirements needed to properly develop a system operating in this domain.	

Internet of Everything: Principles	
Lecturer	Prof. Edoardo Giusto
	edoardo.giusto@unina.it
Credits	2
Course length (in hours)	16
Prerequisites	Intro to computer science: the course will entail an introduction to the Internet of Everything concept, with related applications and trends. The course will highlight technical requirements needed to properly develop a system operating in this domain.
Description:	
Introduction to IoT; Applications and business trends in IoT; Overview of sensors and platforms; Introduction to microcontrollers and hardware architecture; Overview of sensor kits; Basic sensor manipulation and integration; Technical description, planning and business implications; Design and implementation; Network protocols. Final lecture will also entail learning assessment in the form of discussion.	

Machine Learning for Science and Engineering Research	
Lecturers	Profs. Anna Corazza, Roberto Prevete, Carlo Sansone
	anna.corazza@unina.it
Credits	4
Course length (in hours)	20
Prerequisites	Fundamental concepts of probability and statistics.

Description:

The course introduces the main topics in machine learning for both supervised and unsupervised approaches. In addition to a general introduction to the field, we discuss a few topics that are widely considered very effective and promising. In particular, the concept of explainable AI will be discussed, with special attention to the case of neural networks.

Lesson 1 – Introduction to the course. Supervised machine learning: definition of supervised machine learning with particular emphasis on classification. Decision trees, example of classification approaches in the vector space model (Rocchio, kNN), statistical methods, Bayes classification rule and MLE, Naive Bayes classifiers. (Anna Corazza).

Lesson 2 – **Unsupervised machine learning:**
introduction to clustering, flat clustering, K-means, clustering assessment, choice of the number of clusters. Hierarchical clustering: introduction, dendrograms, variants, discriminative cluster labelling, non-discriminative cluster labelling. (Anna Corazza)

Lecture 3 - Feature design: Introduction to the problem of dimensionality reduction; definition of the projection error; geometrical introduction to Principal Component Analysis and its statistical interpretation; introduction to the feature selection problem; the ada-boost algorithm; application to face detection. (Francesco Isgrò)

Lesson 4 – Support Vector Machines: performance assessment, overfitting and generalisation, linear versus non-linear classifiers, hard margin support vector machines (SVM), soft margin support vector machines, kernels. (Anna Corazza)

Lesson 5 – From shallow networks to deep networks:
Structure and behaviour of Multi-layer Feed-Forward Neural Networks. Shallow networks as universal approximators. Error Functions and Optimization methods based on gradient descent. Back-propagation algorithm to compute error gradient. Basic principles of Deep Learning. Unsupervised learning algorithms to pre-train multi-layered neural networks: Noised Stacked Auto-Encoders. (Roberto Prevete)

Lesson 6 – Deep Learning: Deep Network without pretraining: Rectified Linear Units (ReLU) and its variants. Convolutional Neural Networks. Graph Convolutional Neural Networks. GANN (Generative Adversarial Neural Network). (Roberto Prevete)

Lesson 7 – XAI Basic concepts and definitions about interpretation and explanation of autonomous (or semiautonomous) systems based on machine learning. Overview of explanation and interpretability methods for machine learning algorithms. LIME and Layer-wise Relevance Propagation (Roberto Prevete)

Lecture 8 - Selected topics in DL: Neural networks for sequences: Recurrent Neural Networks. Simple Recurrent Neural Networks (S-RNN). Problems with these simple models. Long Short-Term Memory (LSTM) neural networks. Transformers. (Anna Corazza)

Lecture 9 - Ensemble methods: Combining Multiple Models. Bagging. Randomization: Random Subspace

Ensemble, Random Forest, Rotation Forest. Boosting, Additive Regression. Stacking. Error Correcting Output Codes. (Carlo Sansone). **Lecture 10 –** Seminar to be confirmed

Methods for Artificial Intelligence	
Lecturer	Prof.ssa Silvia Rossi
	silvia.rossi@unina.it
Credits	10
Course length (in hours)	48
Prerequisites	//
Description:	

The objective of this course is to provide the students with a full and comprehensive knowledge of AI methods and techniques.
We will introduce classic AI problems, as well as the models and the algorithms devised to address them. The course is divided in three main parts. In the first one, we will study algorithms for the resolution of informed search problems in state space, online search with/without the presence of an opponent, and constraint satisfiability problems. The second part will focus on the reasoning and decisional processes in the case of uncertainty.

We will discuss ways to represent knowledge, including incomplete and uncertain knowledge of the real world. We will then focus on the logical reasoning over the acquired knowledge, using probabilities, and on using these reasoning methods and models to decide what to do. In the last part of the course, we will introduce distributed decision problems. Particularly, we will address game theory approaches for non-cooperative interaction decision problems, and the enforcement of such methods to concrete challenges.

<u>Program</u>

Introduction
• History of AI
Problem Solving
• Types of problems: single-state, multiple-states, context-dependent, heuristic problems, and exploration problems.
• Representation structures and problem formalization for the first three typologies.
• Uninformed Search Strategies:
o Breadth-first search;
o Depth-first search, and Iterative deepening depth-first search;
o Comparing uninformed search strategies.
• Informed (Heuristic) Search Strategies:
o Greedy best-first, A* and admissible heuristics;
o Consistent heuristics;
o Iterative Deepening A*;
o Local search algorithms: Hill-climbing search, Local beam search, Simulated Annealing and Genetic.
• Multi-agent games: zero-sum games as problems to solve;
o Utility functions and minimax strategy;
o Alpha beta pruning.
• Constraint Satisfaction Problems
o CSP as a search problem;
o Backtracking for CSP, Variable and value ordering;
o Constraint propagation: Forward Checking and AC3;

o The Structure of Problems; o Local Search for CSPs. Uncertain knowledge and reasoning • Uncertainty representation: o Introduction to decision theory, Basic Probability Notation, Atomic Events, Unconditional Probability, Inference Using Full Joint Distributions, Conditional Probability; o Independence and conditional independence, Bayes' Rule and Its Use; • Bayesian networks: o The Semantics of Bayesian Networks; o Conditional independence relations in Bayesian networks; o Exact Inference in Bayesian Networks: Inference by enumeration, variable elimination algorithm. • Reasoning in conditions of uncertainty: o Actions and expected utility, Decision Networks and representing a decision problem with a decision network, Value of Information; o Sequential Decision Problems, the utilities of states and Bellman equation for utilities, Value Iteration, Policy Iteration. Multi-Agent systems and Game Theory • Introduction to multi-agent systems; • Introduction to game theory: o Games in normal form and solutions o Pure and mixed strategies; Nash equilibrium, Pareto optimality, social welfare; o Zero-sum games and extended representation; o Computation of Equilibria and Dominant Strategies; o Zero-sum games solutions, minmax and alfabeta. o Repeated games. • Computational social choice: o Voting mechanisms: social choice and social welfare functions; o Arrow property and theorem; o Muller-Satterthwaite theorem; o Ranking functions and PageRank; o Voting strategies and design mechanism (dominant strategies and Bayes Nash); • Allocating scarce resources: o Auction mechanisms, individual-item auctions (English auction, Dutch auction, Japanese auction, Vickrey); o Equilibrium and dominant strategies in auction mechanisms; • Reaching agreements: o Negotiation, domains; o Negotiation in task-oriented domains; o Theoretic games approaches, alternate bidding protocols; o Heuristic approaches, monotone concession protocol, Zeuthen strategy; o Multi-issue negotiation.

Natural Language Processing	
Lecturer	Prof. Francesco Cutugno
	cutugno@unina.it
Credits	5
Course length (in hours)	48
Prerequisites	//
Description:	

The Natural Language Processing class is finalized to give students solid bases and strong knowledge in the field of Computational Linguistics. To pursue this aim the course deals with the following arguments: general linguistics signal processing, speech processing, and pattern matching techniques, ASR and TTS, corpus linguistics. Moreover, lessons will offer know-how in machine learning and artificial intelligence applied to text processing and automatic human-machine dialogue management. <u>Program</u> Basis of General Linguistics, discipline articulation and terminology. Text processing: Regular Expression, text normalization, Ngrams, POS-Tagging, Syntactic Parsing, word sense disambiguation, word embedding, wordnet, lexicons and graph databases. Speech Processing: Acoustic phonetics, PRAAT software, Hidden Markov Models, Speech synthesis, Speech recognition, Spoken dialogue systems, spoken language understanding.

Numerical Methods for Data Analytics	
Lecturer	Prof. Salvatore Cuomo
	salvatore.cuomo@unina.it
Credits	3
Course length (in hours)	15
Prerequisites	Basics on Numerical Linear Algebra and Calculus
Description:	
The teaching includes an Introduction to the analysis of one-dimensional and two-dimensional signals. The sampling in time and frequency of a function. The continuous Fourier transform and the discrete one (DFT). Applications of the DFT: the convolution product; the matrix-vector product with Circulant and Toeplitz matrices; The Fast Fourier Transform (FFT). The continuous and discrete Wavelet transform (DWT). Applications of the DWT to the compression of digital images and the Wavelet packet. Fundamentals of numerical linear algebra for data analysis-oriented problems. Numerical algorithms for the calculation of Singular Values (SVD), QR factorization with orthogonal and iterative transformations, and main theorems of eigenvalue localization. Applications of the SVD to the analysis of dominant information in datasets and Principal Component Analysis (PCA). Non-negative matrix factorization with applications to Text Mining. The student must demonstrate the ability to: - Know and understand the numerical techniques studied, with a clear view of the fields of application; - Be able to use the knowledge acquired to solve specific problems, both using software libraries and with codes designed and produced ad hoc; - Be able to communicate ideas and solutions clearly, rigorously, and effectively to both specialists and non-specialists; - Be able to identify the most appropriate methods to analyze and solve a problem related to the topics of the course and interpret the results correctly.	

Quantum Computational Intelligence	
Lecturer	Prof. Giovanni Acampora
	giovanni.acampora@unina.it
Credits	3 – 6
Course length (in hours)	15 – 30

Prerequisites	Basic knowledge of linear algebra and computer programming
Description:	
This module introduces the basic concepts of the design of quantum algorithm. Specifically, the module deals with the following topics: introduction to Hilbert spaces; differences among deterministic, probabilistic and quantum systems; a brief introduction to the quantum architecture: from qubits to quantum gates; introduction to quantum algorithms: Deutsch-Josza algorithm, Simon's periodicity algorithm, Grover's search algorithm, Shor's factoring algorithm; theoretical computer science and classes of problem complexity; hybrid approaches of quantum computing and artificial intelligence; an overview of quantum programming languages and libraries	

Using Deep Learning Properly	
Lecturer	Dott. Andrea Apicella
	andrea.apicella@unina.it
Credits	2.4
Course length (in hours)	12
Prerequisites	//
Description:	
Designing and implementing a Deep Learning system is not an easy task. The process requires several choices regarding model design, data engineering, parameter modification and testing. This process is easily subject to errors that are not easily identifiable and, in some cases, may lead to over-estimating the performance of the proposed solution. This course aims to provide a general pipeline for designing and validating a machine learning system, avoiding the most common errors that can easily be made. To this end, it will be shown how to implement the experimental evaluation of simple classification tasks, highlighting their peculiarities and points to pay attention to. The practical part of the course is based on PyTorch, one of the best-known packages for neural networks. An introductory view of it is given. **Content details** Lesson 1 – Introduction and first examples. Machine Learning basics. Data leakage. Balancing, normalization and standardization of the data. Randomness and nondeterministic factors. Lesson 2 – Python & Numpy fundamentals for Machine Learning. Preparing the environment. Numpy fundamentals. Common errors with Numpy. Lesson 3 – Pytorch fundamentals. Common libraries for Deep Learning. Pytorch general description. Main modules.	