Ph.D. Program in Information Engineering

Course Catalogue

A.Y. 2022/2023

Rev. 1.2 – 2/11/2022

Revision History

Revisions with respect to the reference version: 1.0 – 17/10/2022

Rev. 1.1 – 18/10/2022

- The course "*IE_BIO 3 Fluid mechanics for the functional assessment of cardiovascular devices*" has been confirmed by the instructor (first lecture on January 10th, 2023).

Rev. 1.2 – 2/11/2022

- Enrollment procedure now moved to a separate section, indexed in the summary

Summary

[back to **Summary**]

Control Theory and Applications (IE_AUT)

Computer Science (IE_CSC)

Applied Optics (IE_OPT)

Course Index by Instructor

Coursework Requirements

The following requirements are valid for Ph.D. Students starting in October 2022 (38° cycle). In summary, Students shall **take courses for a minimum of 20 credits** and shall **attend the seminars proposed by the Ph.D. Program**, following the rules detailed below.

Definitions

A **course** is a series of lectures given by an instructor (professor or university researcher), possibly accompanied by laboratory sessions, that includes an assessment of the student knowledge (final exam, graded homework or project, etc.). **A course gives credits**.

A **seminar**, in the context of the Ph.D. Program, is typically a talk, or a series of talks, given by an academic researcher or an eminent professional, that does not include an assessment. **A seminar does not give credits**.

Course requirements

- Take Ph.D. courses for **a minimum of 20 credits** by the end of the second year.

Specific constraints to earn the minimum of 20 credits of courses:

C.1 **Transversal Skills Area (TSK)**: at least 5 credits should come from courses belonging to the Transversal Skills area (labeled **TSK** in the course Summary) and to the Mathematical and Statistical Methods area (labeled **IE_MSM**).

C.2 **Information Engineering Area (IE_*)**: students shall earn at least 10 credits by taking courses belonging to the Information Engineering Area (labeled **IE_*** in the course Summary, with * being **MSM**, **BIO**, **ELE**, **TLC**, **AUT**, **CSC**, **OPT**).

C.3 **External Courses**: up to a maximum of 5 credits may be earned by taking external courses (i.e. courses not included in this catalogue) falling in the following categories:

- Courses appearing in the list of external courses approved by the Executive Board. The list of credited external courses is available on the Ph.D. Program main website.
- Additional external courses might be included into the list after submission of a written request signed by the Student and his/her Supervisor. Only courses including an exam with grading are considered.
- Courses from other Ph.D. School catalogues (provided they include a final exam with grading).
- In order to get credit recognition for external courses, students shall obtain a certificate stating that the student attended the course and successfully passed the exam. Alternatively, the student may fill a Certification of Attendance with the course data and have it signed by the course instructor.

Seminar requirements

- Attend the **seminars** promoted by the Ph.D. Program and advertised on the website during the three-year Ph.D. course. Students are expected to **attend a minimum of three seminars during their three-year Ph.D. course**, although it is strongly recommended that they attend more than the minimum number required.
- Attend all the lectures of the **Distinguished Lecturer Series** program offered by the Department during the three-year Ph.D. course.
- Attend at least two seminars of the **PhD Educational Week on Transferable Skills (PhDETSWeek) during their three-year Ph.D. course**. The PhDETSWeek is organized by the University of Padova and is typically offered every year. It consists of a series of seminars on transversal topics.

Study plan

Each first-year student **enrolled in the PhD Program in Information Engineering** must fill a tentative study and research plan form and upload it using the following link:

https://phd.dei.unipd.it/study-and-research-plan

within November 5th *(NOTE: PhD Students starting their program later than October 1st shall submit their program of study form within 30 days from their start date)*. The study plan may be subsequently modified by submitting a new form no later than six months before the end of the third year. Seminars, Distinguished Lectures and PhD Educational Week modules should not be included in the program of study. Please, use the Seminar Certificate of Attendance to collect the signature of the speaker or of a member of the Executive Board attending the event.

[back to **Summary**]

Course enrollment and attendance

Unless otherwise indicated in the course syllabus, Students are required to enroll in each course they plan to attend, be it for credits (i.e., taking the final exam) or otherwise, by filling the course enrollment form that can be found at the following link:

Course Enrollment Form

Students are expected to attend classes regularly. Punctuality is expected both from instructors and students. Instructors have to report to the Coordinator of the Ph.D. Program students missing classes without proper excuse.

Instructors shall complete student grading within 30 days after the end of lectures.

Note for students enrolled in PhD Programs other than Information Engineering

PhD Students enrolled in other PhD Programs are welcome to take courses from this Catalogue. External students planning to take a course shall submit a request to be enrolled by sending an e-mail message to: corso.dottorato@dei.unipd.it (PhD Secretariat) **at least two weeks in advance with respect to the date of the course first lecture**. Please note that attendance to a course is typically limited to a maximum number of participants, so the request of enrollment might not be accepted.

External students must be aware that the number of credits awarded by a course and its recognition inside the study plan depend on the rules of the PhD Program the students are enrolled in.

Class Schedule

The class schedule is embedded in the Ph.D. Program Calendar. You may add the Calendar to your Google account through the following link:

Class Schedule of 2022/23 PhD Courses for Google Calendar

You may also visualize the class schedule using any browser through the following link:

Class Schedule of 2022/23 PhD Courses

With very few exceptions, classes meet in classrooms and meeting rooms of the Department of Information Engineering, via Gradenigo 6/A, Padova. In order to locate the rooms, you may find helpful the map of the Department buildings:

Map of the Department of Information Engineering

Please, always check the class schedule in the calendar to verify the room where the class meets.

[back to **Summary**]

TSK 1. Entrepreneurship and Startups

Course Area: Transversal Skills

Credits: 5 (20 hours)

Instructors: Prof. Moreno Muffatto, Ing. Francesco Ferrati, Dipartimento di Ingegneria Industriale, Università di Padova

e-mail: moreno.muffatto@unipd.it, francesco.ferrati@unipd.it

Notes: the course delivers an Open Badge that can be redeemed at
https://bestr.it/badge/show/2670

Topics:

Entrepreneurship

The team and the early decisions

From the idea to the market

Intellectual Property Rights

Business Models

The financials of a startup

Funding a startup

Entrepreneurship

- Entrepreneurship and Entrepreneurial attitudes
- Entrepreneurship vs Management
- What is a technology based startup
- Venture creation: different options

The team and the early decisions

- The creation of the founders' team
- Types and characteristics of founders' teams
- Founders' decisions and their consequences
- Frequent mistakes and suggestions deriving from experience

From the idea to the market

- Innovation: technologies and markets
- Market size
- Customers profiles
- Value proposition
- Development of the product/service concept

[back to **Summary**]

Intellectual Property Rights

- Types of IPR (patent, copyright, trademark)
- The structure of a patent application (description, claims, etc)
- Getting a patent: the patenting process (step by step)
- When to file a patent application: priority date, Patent Cooperation Treaty (PCT)
- Where to protect an invention
- Different IPR strategies

Business Models

- Business models case studies
- Successful and problematic business models
- Revenue streams
- Cost of Customer Acquisition

The financials of a startup

- The structures of the financial statements
- Income Statement
- Balance Sheet
- Cash Flow Statement
- Evaluation of the value of the company

Funding a startup

- New ventures' funding options
- Different sources of funds: Angel Investors and Venture Capital
- Investment companies and funds: how they work
- How and what investors evaluate
- How to present a business idea to investors

References:

- Thomas R. Ittelson (2009), *Financial Statements: A Step-by-Step Guide to Understanding and Creating Financial Reports*, Career Press.
- Ferrati, F. & Muffatto, M. (2021). "Reviewing Equity Investors' Funding Criteria: A Comprehensive Classification and Research Agenda". *Venture Capital*, Vol. 23: No. 2, pp. 1-22.
- Noam Wasserman (2013) *The Founder's Dilemmas: Anticipating and Avoiding the Pitfalls That Can Sink a Startup*, Princeton University Press.

Schedule and room: please, see Class Schedule

[back to **Summary**]

Enrollment: students planning to attend the course must

1. register via the Moodle platform of the PhD Course in Industrial Engineering (in order to enter the Moodle platform click on "dettagli" of the course at the page http://www.cdii.dii.unipd.it/corsi). Registered students that, for any reason, are not able to attend the course, must inform the lecturer;
2. then, add the course to the list of courses they plan to attend using the Course Enrollment Form (requires SSO authentication) and, if taking the course for credits, to the Study and Research Plan.

Examination and grading: Attendance is required for at least 70% of the lecture hours (i.e. 14 hours). Final evaluation will be based on the discussion of a case study of a technology-based startup.

TSK 2. Python Programming for Data Science and Engineering

Course Area: Transversal Skills

Credits: 5 (20 hours)

Instructor: Dr. Stefano Tortora, Department of Information Engineering (DEI), University of Padova

e-mail: stefano.tortora@unipd.it

Aim: Python is an easy-to-learn and powerful high-level language and it is becoming more and more popular for scientific applications such as machine learning, statistics, manipulating and transforming data, but also computer vision and robotics. The first objective of the course is to become familiar with Python syntax, environments and basic libraries. Secondly, the learner will be guided in performing basic inferential data analyses and introduced to the application of common machine learning algorithms.

Topics:

- Introduction to the Python Programming Language
    - What is different in Python?
    - The Python Language Syntax
- Modules and Packages
    - NumPy and SciPy: Numerical and Scientific Python
    - Pandas: Labeled Column-Oriented Data
    - Matplotlib: MATLAB-style scientific visualization
    - Scikit-learn: Basics of Machine Learning in Python

References:

[1] J. VanderPlas, "A Whirlwind Tour of Python", O'Reilly Media Inc. 2016. [Online: https://www.oreilly.com/programming/free/files/a-whirlwind-tour-of-python.pdf]

[2] J. VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data" O'Reilly Media Inc. 2017.

[3] B. Miles, "Begin to Code with Python", Pearson Education, Inc. 2018. [Online: https://aka.ms/BeginCodePython/downloads]

[4] Z. Shaw, "Learn Python the Hard Way", Addison-Wesley. 2014.

[5] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", O'Reilly Media Inc. 2019.

Schedule and room: please, see Class Schedule

[back to **Summary**]

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Backgrounds in computing with some object-oriented programming language: C++, Java, MATLAB, etc. If you are starting from scratch, please have a look at [3] or [4].

Examination and grading: Homework assignments

IE_MSM 1. Statistical Methods

Course Area: Information Engineering (may also be taken as Transversal Skills course)

Credits: 6 (24 hours)

Instructor: Dr. Lorenzo Finesso (formerly of CNR-IEIIT Padova)

e-mail: lorenzo.finesso@unipd.it

Importan note: course not offered in A.A. 2022/23

IE_MSM 2. Statistics for Engineers

Course Area: Information Engineering (may also be taken as Transversal Skills course)

Credits: 5

Number of lecture:

Instructors: Prof. Luigi Salmaso, Prof. Rosa Arboretti, Prof. Marta Disegna, University of Padova.

e-mail: luigi.salmaso@unipd.it, rosa.arboretti@unipd.it, marta.disegna@unipd.it

Important note: enrollment in this specific course is reserved to Students of the Ph. D. Program in Information Engineering. Students of other Ph.D. Programs please refer to their Program secretariat.

Outline of lecture and lab: The course is structured into 2 on-campus lectures and a Summer School of 4 days. A total of 40 hours in-person course will be delivered.

The on-campus lectures will take place on Friday the 3th February 2023 and Friday the 10th February 2023. Classes will take place online in the morning, 9am to 1pm, and in the afternoon, 2pm to 4pm for a total of 6 hours per day.

The Summer School will take place in Villa San Giuseppe, Monguelfo, Bolzano province (https://www.villasangiuseppemonguelfo.com) from Tuesday the 27th June 2023 to Friday the 30th June 2023 for a total of 28 hours. The Summer school will start at 2pm on Tuesday and will finish at 1pm on Friday (plus lunch served in Villa San Giuseppe and included in the fees).

Villa San Giuseppe offers a full board accommodation and rooms are of different size. The cost of the Summer School is €150 (for the full board accommodation to be paid on site) for the entire period. Students of the PhD Program in Information Engineering are reimbursed the full cost of the School (full board accommodation and travel).

Payment can be made directly in “Villa San Giuseppe”, no need to pre-pay anything in advance.

Aim: The course is an introduction to statistical methods most frequently used for experimentation in Engineering. Lectures are planned both in the classroom and in computer lab also for an introduction to the use of the following statistical software:

- R and Rstudio, both open-source software.
- MINITAB, licensed to University of Padova.

Topics:

1. Elements of univariate statistical methods:
    a. Elements of descriptive statistics: frequency, indices of synthesis (position, variability and shape) and graphical representations (histogram, boxplot, scatterplot).
    b. Elements of probability theory: discrete and continuous probability distributions.

   c. Elements of statistical inference: sampling distributions, point and interval estimation, hypothesis testing, One-way ANOVA.

2. Linear and non-linear regression models:
   a. Simple and multiple linear regression model
   b. Logit model
3. Multivariate data analysis:
   a. Cluster Analysis: idea and steps
   b. Multidimensional data, matrix representation and data preparation.
   c. Distance and dissimilarity matrices.
   d. Hard clustering algorithms: hierarchical clustering algorithms, non-hierarchical clustering algorithms and Bagged clustering algorithm.
   e. Fuzzy clustering algorithms: fuzzy C-means and fuzzy C-medoids.
   f. Validity indices and optimal number of clusters.
   g. Labelling and profiling the clusters: an application of suitable tests and regression models.
4. DOE: Introduction to Factorial Designs, Two level and general factorial designs. Tutorials in MINITAB.

Examination and grading: Attendance is required for at least 2/3 of the lecture hours. Final evaluation will be based on the discussion of a case study, preferably drawn from the individual PhD project of one of the group members.

For this course you are expected to describe and analyse a case study using the statistical techniques presented during the course or more advanced statistical techniques, when more suitable. You are required to present your case study on a **maximum of 15 slides** by the **30th of September 2022 at 12:00 noon**. The presentation should include: aim of your project, description of the dataset, description of the method, interpretation of the results and final discussion. You should provide both the presentation and the dataset analysed (if possible). You are allowed to work alone or in team of no more than 3 people.

You can upload your project in the "Presentation_2022" Google Drive folder using the following link

https://drive.google.com/drive/folders/1jkP1tb7VRgTpzVKXt5AgatLFA8gFqYyF?usp=sharing

If you have any problem regarding the upload of the files email to marta.disegna@unipd.it

Please state clearly in the presentation the name(s) of the participant(s).

Some projects will be selected for an oral presentation to the class. The selected projects and the data of the presentation will be communicated in September.

Materials (slides, datasets, etc.) of the course is available at the following link:

https://drive.google.com/drive/folders/1jkP1tb7VRgTpzVKXt5AgatLFA8gFqYyF?usp=sharing

[back to **Summary**]

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan. Please note that enrollment in this specific course is reserved to Students of the Ph. D. Program in Information Engineering. Students of other Ph.D. Programs please refer to their Program secretariat.

Schedule and room: please, see **Outline of lecture and lab** above or the Class Schedule

IE_MSM 3. Computational Inverse Problems

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Fabio Marcuzzi, Dept. of Mathematics, University of Padova.

e-mail: marcuzzi@math.unipd.it

Aim: We study numerical methods that are of fundamental importance in computational inverse problems. Real application examples will be given for distributed parameter systems in continuum mechanics. Computer implementation performance issues will be considered as well.

Topics:

- definition of inverse problems, basic examples and numerical difficulties.
- numerical methods for QR and SVD and their application to the square-root implementation in PCA, least-squares, model reduction and Kalman filtering; recursive least-squares; High Performance Computing (HPC) implementation of numerical linear algebra algorithms.
- regularization methods;
- underdetermined linear estimation problems and sparse recovery;
- numerical algorithms for nonlinear parameter estimation: nonlinear least-squares (Levenberg-Marquardt), back-propagation learning;
- underdetermined nonlinear estimation problems and deep learning;
- examples with distributed parameter systems in continuum mechanics: reconstruction of forcing terms and parameters estimation;

References:

[1]F.Marcuzzi "Computational Inverse Problems", lecture notes (will be posted on the moodle page of the course)

[2]G. Strang, "Linear Algebra and Learning From Data", Wellesley - Cambridge Press, 2019

[3]L. Trefethen and J. Bau, “Numerical Linear Algebra”, SIAM, 1997

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements:

- basic notions of linear algebra and, possibly, numerical linear algebra.

[back to **Summary**]

- the examples and homework will be in Python (the transition from Matlab to Python is effortless).

Examination and grading: Homework assignments and final test.

IE_MSM 4. Heuristics for Mathematical Optimization

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Domenico Salvagnin

e-mail: dominiqs@gmail.com, domenico.salvagnin@unipd.it

Aim: Make the students familiar with the most common mathematical heuristic approaches to solve mathematical/combinatorial optimization problems. This includes general strategies like local search, genetic algorithms and heuristics based on mathematical models.

Topics:

- Mathematical optimization problems (intro).
- Heuristics vs exact methods for optimization (intro).
- General principle of heuristic design (diversification, intensification, randomization).
- Local search-based approaches.
- Genetic/population based approaches.
- The subMIP paradigm.
- Applications to selected combinatorial optimization problems: TSP, QAP, facility location, scheduling.

References:

[1] Gendreau, Potvin "Handbook of Metaheuristics", 2010

[2] Marti, Pardalos, Resende "Handbook of Heuristics", 2018

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements:

- Moderate programming skills (on a language of choice)
- Basics in linear/integer programming.

Examination and grading: Final programming project.

IE_BIO 1. Statistical Learning for Big Data in Medicine

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Andrea Facchinetti (Department of Information Engineering, University of Padova), Dr. Martina Vettoretti (Department of Information Engineering, University of Padova)

Importan note: course not offered in A.A. 2022/23

[back to **Summary**]

IE_BIO 2. Introduction to Model Predictive Control with Case Studies in Automotive and Biomedicine

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Ruggero Carli, Dr. Mattia Bruschetta, Dr. Simone Del Favero, Department of Information Engineering, University of Padova

e-mail : carlirug@dei.unipd.it, mattia.bruschetta@dei.unipd.it, simone.delfavero@unipd.it

Aim: To provide the methodological tools needed to understand model-based control algorithms and to design a Model Predictive Control algorithm for a linear dynamical system. The course is tailored on students who have not received an extensive training on control theory. As case studies, the course focus on Automotive and Bioengineering applications.

Topics:

1. Introduction to model-based control.
2. State Space Models: driving the state with inputs.
3. State Space Model: estimating the state form the output.
4. Linear Quadratic Regulator (finite and infinite horizon).
5. Model Predictive Control - Regulation: Formulation, Dynamic Programming Solution, Stability properties, MPC for Unconstrained Systems, MPC for Systems with Input Constraints, MPC for Systems with Input and State Constraints.
6. Offset-free Model Predictive Control: disturbance estimation, partial velocity form, full velocity form.
7. Elements of Nonlinear MPC.
8. Automotive case studies: Motion Cueing Algorithms, Virtual Rider, Autonomous Driver.
9. Biomedicine case study: the Artificial Pancreas, Automated Drug Infusion for Anesthesia.

References:

[1] J. B. Rawlings and D. Q. Mayne. Model predictive control: Theory and design. Nob Hill Publisher.

Other material and research papers will be available online for download.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basic Calculus and Linear Algebra.

Examination and grading: Homework and take home exam

[back to **Summary**]

IE_BIO 3. Fluid mechanics for the functional assessment of cardiovascular devices

Course Area: Information Engineering

Credits: 5 (18 hours)

Instructor: Prof. Francesca Maria Susin, Dept. of Civil, Environmental and Architectural Engineering (DICEA)

e-mail: francescamaria.susin@unipd.it

Aim:

The course is intended to give a survey of research approaches for the assessment of cardiovascular medical devices. Emphasis will be given to methods and techniques adopted for in vitro analysis of hemodynamic performance of prosthetic heart valves.

Topics:

Review of basic fluid mechanics concepts. Fluid mechanics of prosthetic heart valves (PHVs). Pulse duplicators for in vitro testing of PHVs and mock circulation loops for pre-clinical evaluation of VADs. Experimental techniques for the assessment of PHVs. CFD for functional assessment of PHVs.

References:

[1] M. Grigioni, C. Daniele, G. D'Avenio, U. Morbiducci, C. Del Gaudio, M. Abbate and D. Di Meo. Innovative technologies for the assessment of cardiovascular medical devices: state of the art techniques for artificial heart valve testing. Expert Rev. Medical Devices, 1(1) : 81-93, 2004.

[2] K.B. Chandran, A.P. Yoganathan and S.E. Rittgers. Bio fluid Mechanics: the human circulation. CRC Press, Boca Raton, FL, 2007.

[3] A.P. Yoganathan, K.B. Chandran and F. Sotiropoulos. Flow in prosthetic heart valves: state of the heart and future directions. Annals of Biomedical Engineering, 33(12) : 1689-1694, 2005.

[4] A.P. Yoganathan, Z. He and S. Casey Jones. Fluid mechanics of heart valves.

[5] A.P. Yoganathan and F. Sotiropoulos. Using computational fluid dynamics to examine the hemodynamics of artificial heart valves. Business briefing: US cardiology 2004 : 1-5, 2004.

[6] V. Barbaro, C. Daniele and M. Grigioni. Descrizione di un sistema a flusso pulsatile per la valutazione delle protesi valvolari cardiache. ISTI-SAN Report 91/7, Rome, Italy, 1991 (in Italian).

[7] M. Grigioni, C. Daniele, C. Romanelli and V. Barbaro. Banco di prova per la caratterizzazione di dispositivi di assistenza meccanica al circolo. ISTISAN Report 03/21, Rome, Italy, 2003 (in Italian).

[8] M.J. Slepian, Y. Alemu, J.S. Soares. R.G. Smith, S. Einav and D. Bluestein. The Syncardia total arti cial heart: in vivo, in vitro, and computational modeling studies. Journal of Biomechanics,46 (2013): 266-27, 2013.

[back to **Summary**]

Further references will be given during the course.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Fundamentals of Fluid Dynamics.

Examination and grading: Homework assignment with final discussion.

[back to **Summary**]

IE_BIO 4. Quantitative Neuroimaging: from Microparameters to Connectomics

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Alessandra Bertoldo, Prof. Mattia Veronese, Department of Information Engineering, University of Padova

e-mail : alessandra.bertoldo@unipd.it, mattia.veronese@unipd.it

Aim: The course aims to give the methodological knowledge necessary to define a directed or non-directed brain network (compartmental models, seed-based analysis, ICA, mutual information, dynamic casual modeling) and its topographic analysis (graph theory).

Topics:

- What is quantitative neuroimaging: methods to quantify PET microparameters and fMRI features (preprocessing, the role of the atlases, input/output models, compartmental models, seed based analysis, ICA);
- Functional and effective connectivity (static and dynamic): correlation, sliding windows, Hidden Markov Models, Dynamic Casual Modeling, Granger and Transfer entropy.
- Metabolic connectivity (static and dynamic): SICE methods & Non-SICE methods (Pearson, Cosine, Euclidean)
- Network theory applied on brain connectivity
- Hands-on: analysis of neuroimaging data to estimate functional connectivity maps and derived graph measures.

References: Lecture notes and a complete list of references will be made available by the Lecturers.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basics of modeling, system identification and statistics; basics of Matlab programming.

Examination and grading: Final project consisting in the definition and analysis of a brain network using MRI and/or PET data.

IE_ELE 1. Diagnostics of Electron Devices

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Giovanna Mura, Department of Electrical and Electronic Engineering (DIEE), University of Cagliari.

e-mail: gmura@diee.unica.it

Aim: this course provides an overview of the Failure Analysis techniques for the diagnostics of electron devices. Failure analysis is the process of analyzing the failed electron devices to determine the reason for degraded performance or catastrophic failure and to provide corrective actions able to x the problem. It is a proactive tool with three fundamental tasks: 1) Technical/scientific: 2) Technological 3) Economical. The purpose of this course is to teach what Failure Analysis should be and should do, to show how and why it often does not, to state that F.A. has Logics and has Rules.

Microscopy, in its several forms (optical, electron, scanning, transmission, emission, ionic) and tools is the playground for practical FA, and its fundamentals will be described. Device basic technology, working principle and failure physics are the other pillars for a successful study.

Several case studies will be proposed with the aim to demonstrate that if sometimes Failure Analysis looks unclear or not problem solving is merely because it was badly conducted.

Topics:

1. Reverse engineering
2. Failure modes and failure mechanisms
3. Principles and fundamental methods in Electron Microscopy
4. Methodology for the Failure Analysis

References: Failure Analysis of Integrated Circuits - Tools and Techniques, Springer International Series - Lawrence C. Wagner.

Slides

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Electron Devices, Microelectronics, Optoelectronics devices.

Examination and grading: Written test/ presentation of a report at the end of the course.

[back to **Summary**]

IE_ELE 2. Physics and Operation of Heterostructure-Based Electronic and Optoelectronic Devices

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Matteo Meneghini, Dr. Carlo De Santi, DEI, University of Padova,
Prof. Edwin L. Piner, Texas State University

e-mail: matteo.meneghini@unipd.it, carlo.desanti@unipd.it, epiner@txstate.edu

Aim: This course provides an introduction to the physics and operating principles of advanced electronic and optoelectronic devices based on compound semiconductors. These devices are particularly important for several applications: high electron mobility transistors (HEMTs) represent excellent devices for the realization of high frequency communication systems, radars, satellite applications, and high efficiency power converters. On the other hand, LEDs and lasers are high-efficiency monochromatic light sources, that can be used both for lighting applications (with a considerable energy saving), in the biomedical field, and in in photochemistry. Special focus will be given to Gallium Nitride (GaN) based devices, that represent the most promising devices for future power electronics applications. This course will focus on the main aspects related to the physics of heterostructures, on the recombination processes in semiconductors, on carrier transport in heterostructures, on the structure and operating principles of MESFET, HEMTs, GITs, on the trapping and reliability in compound semiconductor devices, on the operating principles of LEDs and lasers, and on parasitics and reliability in LEDs and lasers. An overview of real applications highlighting the capabilities of these devices will also be given.

Topics:

- physics of heterostructures, band diagrams, carrier transport in heterostructures;
- recombination processes in semiconductors; properties of compound semiconductors;
- basic structure of heterojunction transistors, MESFET, HEMT, GIT; parasitics and reliability in HEMTs, LEDs and lasers;
- operating principles of LEDs and lasers;
- fabrication and development of nitride-based devices
- methods for advanced characterization of heterojunction based devices; applications of GaN based HEMTs, LEDs and lasers;
- modeling of semiconductor-based devices

References:

Umesh Mishra, Jasprit Singh, Semiconductor Device Physics and Design, Springer, 2008

Ruediguer Quay, Gallium Nitride Electronics, Springer 2008.

[back to **Summary**]

Tae-Yeon Seong, Jung Han, Hiroshi Amano, Hadis Morkoc, III-Nitride Based Light Emitting Diodes and Applications, Springer 2013

S. Pearton (ed.), GaN and ZnO-based Materials and Devices, Springer Series in Materials Science, Vol. 156 (2012). Berlin: Springer-Verlag. DOI: 10.1007/978-3-642-23521-4_7

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Introductory course of device physics: Microelectronics, Optoelectronic and Photovoltaic Devices.

Examination and grading: homeworks assigned during the course.

IE_ELE 3. Embedded Design with FPGA

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Dr. Andrea Stanco and Prof. Daniele Vogrig, Department of Information Engineering, University of Padova

e-mail: andrea.stanco@dei.unipd.it, vogrig@dei.unipd.it

Important note: not offered in A.A. 2022/23

[back to **Summary**]

IE_TLC 1. Machine Learning for Mobile Communication Systems

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Dr. Paolo Dini, Senior Researcher, Centre Tecnologic de Telecomunicacions de Catalunya

e-mail: paolo.dini@cttc.es

Aim: The course will introduce the requirements, scenarios and architectures for the next-generation mobile networks, together with their challenges and open issues. We will discuss the central role played by the historical data exchanged among the different network entities and how to distribute computing operations across them to enable automatic and energy efficient extraction of context information and network control. The core focus of the course is the application of Machine Learning (ML) tools to solve identified mobile networking and computing problems. It will be explained what the usage models are and what they imply in terms of stability, convergence and optimality guarantees. For this, fundamentals of Reinforcement Learning and Artificial Neural Networks / Deep Learning will be given. Moreover, Multi-task Learning, Knowledge Transfer Learning, Continual Learning and Federated Learning paradigms for networked systems will be introduced. Finally, several ML algorithms will be tailored for specific case studies. We will examine the automatic control of base station operation modes to solve the Energy-Quality of Service trade-off; and how to build models for mobile traffic prediction, classification and anomaly detection using real data from mobile operators. The course covers Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning applications to mobile networking and computing.

Topics:

- Introduction of next-generation mobile network scenarios and architectures
    - data-centric network scenario
    - multi-access edge computing and distributed learning
    - vertical markets and services
    - energy sustainability issues
- Identification of machine learning tools for mobile networking and computing
- Fundamentals of Artificial Neural Network architectures
    - Multi-layer perceptron
    - Recurrent neural networks
    - Convolutional neural networks
    - Auto-encoders
- Distributed Learning in networked systems
    - Multi-task learning
    - Knowledge Transfer learning
    - Continual learning
    - Federated learning (including centralized and decentralized architectures)

[back to **Summary**]

- Fundamentals of Reinforcement Learning
    - Dynamic Programming
    - Temporal-Difference methods
    - Deep-Reinforcement Learning
- Mobile traffic characterization and modeling
    - Applications of Artificial Neural Networks
    - Traffic prediction, classification and anomaly detection
- Mobile network on-line optimization methods
    - Applications of Reinforcement Learning
    - Multi-agent Reinforcement Learning

References:

[1] Anwer Al-Dulaimi, Xianbin Wang, Chih-Lin I, 5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management, October 2018, Wiley-IEEE Press

[2] R. Boutaba, M.A. Salahuddin, N. Limam, et al., A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Elsevier J. Internet Serv Appl (2018) 9: 16

[3] Richard Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, MIT press, 2017

[4] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT press

Schedule and room: lecture dates and hours will be published on Class Schedule when available.

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basic knowledge of probability theory, random processes, python scripting.

Examination and grading: Each student will develop a final project, possibly related to his/her research activity, addressing some topic presented in the Course.

[back to **Summary**]

IE_TLC 2. Introduction to Reinforcement Learning

Course Area: Information Engineering

Credits: 5 (18 hours)

Instructor: Dr. Juan José Alcaraz Espín, Associate Professor, Technical University of Cartagena, Spain.

e-mail: juan.alcaraz@upct.es

Aim: Reinforcement Learning (RL) is the branch of Machine Learning concerned with sequential decision processes. It has a wide range of applications, including robotics, autonomous driving, video games, and many others. This course will provide an introduction to RL, covering its mathematical foundations and the description of the most relevant algorithms. The main concepts and techniques will be illustrated with multiple examples and Python code. The course will start with the basic concepts of learning in sequential decision problems, formalized in the multi-armed bandit (MAB) problem. Then, we will introduce Markov decision processes (MDPs), which formalizes the kind of problems that RL aims to solve. The main RL approaches will be presented incrementally: 1) tabular methods, which can address relatively small problems, 2) value function approximation, which allows us to scale up previous algorithms to larger problems, and 3) policy gradient algorithms which follow a different scaling approach and can be used in combination with value function approximation (Actor-Critic methods). The course will end with an introduction to some of the most advanced methods such as TRPO, PPO and SAC.

Topics:

Unit 1. Introduction to Reinforcement Learning

Unit 2. Multi-Armed Bandits: Stochastic bandits. Incremental estimation. Boltzmann exploration, UCB algorithms, Thompsom sampling.

Unit 3. Markov Decision Processes: Elements and definitions. Policy evaluation. The Bellman operator. Q-functions. Bellman optimality equations. Solution algorithms for MDPs.

Unit 4. Monte Carlo methods: Introduction. MC prediction and control. Off-policy methods.

Unit 5. Temporal Difference methods: TD prediction and control. Convergence results. Improved control methods. N-step methods.

Unit 6. Value Function Approximation (VFA) methods: Introduction to VFA. Prediction methods. Feature construction. Control with VFA. Elegibility traces. Batch reinforcement learning. Deep Q networks.

Unit 7. Policy gradient algorithms: Elements of policy gradient methods. The policy gradient theorem, Monte Carlo policy gradient. Actor-Critic methods.

Unit 8. Advanced Policy Gradient and Actor-Critic Methods: Vanilla Policy Gradient, Trust Region Methods (TRPO, PPO), Maximum Entropy RL (Soft Q-learning, Soft Actor-Critic), Practical Considerations.

References:

[1] Reinforcement Learning: An Introduction, Second Edition, Richard S. Sutton and Andrew G. Barto, MIT Press, Cambridge, MA, 2018.

[2] Approximate Dynamic Programming: Solving the Curses of Dimensionality, Second Edition, Warren B. Powell, Wiley, 2011.

[3] Dynamic Programming and Optimal Control Vol I and Vol II, 4^{th} Edition, Dimitri P. Bertsekas, Athena Scientific, 2012.

[4] Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan and Claypool, 2010.

[5] Reinforcement Learning and Optimal Control, Dimitri P. Bertsekas, Athenea Scientific, 2019.

[6] Markov Decision Processes: Discrete Stochastic Dynamic Programming, Martin L. Puterman, Wiley, 2006.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course Requirements: Basics of linear algebra, probability theory, Python scripting

Examination and Grading: The grading will be based on a test to assess basic understanding of the topic.

IE_TLC 3. Information Theoretic Models in Security

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Nicola Laurenti, Department of Information Engineering

e-mail: nil@dei.unipd.it

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Nicola Laurenti, Department of Information Engineering

e-mail: nil@dei.unipd.it

Aim: The class aims at providing the students with an information theoretic framework that will allow formal modeling, understanding of the fundamental performance limits, and derivation of unconditionally secure mechanisms for several security-related problems.

Topics:

Lectures 1-6 will explore the following fixed topics:

- *All or nothing: security without compromise.* Quantitative definition of security. Unconditional security. Distinguishability. Composable security
- *The Holy Grail of perfect secrecy.* Shannon's cipher system. Perfect secrecy. Ideal secrecy. Practical secrecy. The guessing attack.
- *Secrecy without cryptography.* The wiretap channel model. Rate-equivocation pairs. Secrecy rates. Secrecy capacity for binary, Gaussian and fading channel models.
- *Alea iacta est.* Secure and true random number generation. Randomness extractors and smooth guessing entropy
- *Who's who?* Unconditionally secure authentication and integrity protection.
- *Security from uncertainty.* Secret key agreement from common randomness on noisy channels. Information theoretic models and performance limits of quantum key distribution.

Lectures 7-10 will introduce a few topics chosen by the students and the instructor among the following:

- *Who's who? Part 2* An information theoretic model for authentication in noisy channels. Signatures and fingerprinting.
- *The gossip game.* Broadcast and secrecy models in multiple access channels. The role of trusted and untrusted relays.
- *Secrets in a crowd.* Information theoretic secrecy in a random network with random eavesdroppers. Secrecy graphs and large networks secrecy rates.
- *A cipher for free?* Information theoretic security of random network coding.

[back to **Summary**]

- *The jamming game.* Optimal strategies for transmitters, receivers and jammers in Gaussian, fading and MIMO channels.
- *Writing in sympathetic ink*. Information theoretic models of steganography, watermarking and other information hiding techniques.
- *Leaky buckets and pipes.* Information leaking and covert channels. Timing channels.
- *The Big Brother.* Privacy and anonymity measures. Differential privacy. The privacy vs utility database tradeoff.
- *The dining cryptographers.* Unconditional secret sharing and secure multiparty computation.
- *Information theoretic democracy.* Privacy, reliability and verifiability in electronic voting systems.

References:

- Y. Liang, H.V. Poor, and S. Shamai (Shitz), *Information Theoretic Security*, Now, 2007.
- M. Bloch, J. Barros, *Physical-Layer Security: from Information Theory to Security Engineering*, Cambridge University Press, 2011.

A short list of reference papers for each lecture will be provided during class meetings.

Schedule and room: please, see Error: Reference source not found

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basic notions of Information Theory (e.g., those from the *Telecomunicazioni* class in the *Corso di Laurea in Ingegneria dell'Informazione*).

Examination and grading: Each student (or small group of students) must submit a project, and grading will be based on its evaluation. Students are encouraged to work from an information theoretic point of view on a security problem related to their research activities.

IE_AUT 1. Elements of Deep Learning

Course Area: Information Engineering

Credits: 6 (24 hours)

Instructor: Dr. Gian Antonio Susto

e-mail: gianantonio.susto@dei.unipd.it,

Aim: The course will serve as an introduction to Deep Learning (DL) for students who already have a basic knowledge of Machine Learning. The course will move from the fundamental architectures (e.g. CNN and RNN) to hot topics in Deep Learning research.

Topics:

- Introduction to Deep Learning: context, historical perspective, differences with respect to classic Machine Learning.
- Feedforward Neural Networks (stochastic gradient descent and optimization).
- Convolutional Neural Networks.
- Neural Networks for Sequence Learning.
- Elements of Deep Natural Language Processing.
- Elements of Deep Reinforcement Learning.
- Unsupervised Learning: Generative Adversarial Neural Networks and Autoencoders.
- Laboratory sessions in Colab.
- Hot topics in current research.

References:

[1] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. CoRR, abs/1701.07875.

[2] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. CoRR, abs/1409.0473.

[3] I. Goodfellow, Y. Bengio, A. Courville 'Deep Learning', MIT Press, 2016

[4] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural computation, 9 8, 1735-80.

[6] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. ACL.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM, 60, 84-90.

[8] LeCun, Y. (1998). Gradient-based Learning Applied to Document Recognition.

[back to **Summary**]

[9] Mikolov, T., Sutskever, I., & Chen, K. (2013). Representations of Words and Phrases and their Compositionality.

[10] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, 11, 3371-3408.

[11] Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent Neural Network Regularization. CoRR, abs/1409.2329.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basics of Machine Learning and Python Programming.

Examination and grading: Final project.

IE_AUT 2. Applied Functional Analysis and Machine Learning: from regularization to deep networks

Course Area: Information Engineering

Credits: 7 (28 hours)

Instructor: prof. Gianluigi Pillonetto

e-mail: giapi@dei.unipd.it

Aim: The course is intended to give a survey of the basic aspects of functional analysis, machine learning, regularization theory and inverse problems.

Topics: Review of some notions on metric spaces and Lebesgue integration: Metric spaces. Open sets, closed sets, neighborhoods. Convergence, Cauchy sequences, completeness. Completion of metric spaces. Review of the Lebesgue integration theory. Lebesgue spaces.

Banach and Hilbert spaces: Finite dimensional normed spaces and subspaces. Compactness and finite dimension. Bounded linear operators. Linear functionals. The finite dimensional case. Normed spaces of operators and the dual space. Weak topologies. Inner product spaces and Hilbert spaces. Orthogonal complements and direct sums. Orthonormal sets and sequences. Representation of functionals on Hilbert spaces.

Reproducing kernel Hilbert spaces, inverse problems and regularization theory: Representer theorem. Reproducing Kernel Hilbert Spaces (RKHS): definition and basic properties. Examples of RKHS. Function estimation problems in RKHS. Tikhonov regularization. Support vector regression and classification. Extensions of the theory to deep kernel-based networks: multi-valued RKHSs and the concatenated Representer Theorem.

References:

[1] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, L. Ljung. Regularized System Identification – learning dynamic models from data, Springer Nature 2022

[2] W. Rudin. Real and Complex Analysis, McGraw Hill, 2006

[3] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2006

[4] H. Brezis, Functional analysis, Sobolev spaces and partial differential equations, Springer 2010

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

[back to **Summary**]

Course requirements: The classical theory of functions of real variable: limits and continuity, differentiation and Riemann integration, infinite series and uniform convergence. The arithmetic of complex numbers and the basic properties of the complex exponential function. Some elementary set theory. A bit of linear algebra.

Examination and grading: Homework assignments and final test.

IE_AUT 3. Applied Linear Algebra

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Luca Schenato, Dipartimento di Ingegneria dell'informazione, Università di Padova (http://automatica.dei.unipd.it/people/schenato.html)

e-mail: schenato@dei.unipd.it

Aim: We study concepts and techniques of linear algebra that are important for applications with special emphasis on the topics: solution of systems of linear equations with particular attention to the analysis of the backward error and computational cost of the basic algorithms and matrix equation. A wide range of exercises and problems will be an essential part of the course and constitute homework required to the student.

Topics:

1. Vectors: inner products, norms, main operations (average, standard deviation, ...)
2. Matrices: matrix-vector and matrix-matrix multiplication, Frobenius norm,
3. Complexity, sparsity
4. Special matrices: Diagonal, Upper Triangular, Lower triangular, Permutation (general pair), inverse and orthogonal
5. A square and invertible: LU decomposition (aka gaussian elimination), LU-P decomposition, Cholesky decomposition
6. Ax=b via LU-P decomposition: forward and backward substitution
7. (sub)Vector spaces: definitions, span, bases (standard, orthogonal, orthonormal), dimension, direct sum, orthogonal complement, null space, orthogonal complement theorem
8. Gram-Smith orthogonalization and QR decomposition (square and invertible A, general non-square)
9. Ax=b via QR decomposition. LU-P vs QR
10. Linear maps: image space, kernel, column and row rank
11. Fundamental Theorem of Linear Algebra (Part I): rank-nullity Theorem, the 4 fundamental subspace
12. Eigenvalues/eigenvector and Shur decomposition
13. Projection matrices: oblique and orthogonal, properties
14. Positive semidefinite matrices: properties and quadratic functions square root matrix
15. Properties of A'A and AA' and Polar decomposition
16. Singular Value Decomposition: proofs and properties
17. Pseudo-inverse: definition and relation to SVD
18. Fundamental Theorem of Linear Algebra (Part II): special orthogonal basis for diagonalization
19. Least-Squares: definition, solution and algorithms

[back to **Summary**]

20. Ill-conditioned problems vs stability of algorithms, numerical conditioning of algorithms, numerical conditionings

Objectives:

- Theory: formal proofs of many results (theorem-proof type problems)
- Algorithms: understanding of most commonly used algortithm used in MATLAB and Python for Linear Algebra
- Implementation: MATLAB implementation of algorithms and performance evaluation on Big Data

References:

Textbooks and Internet Notes:

1. S. Boyd, L. Vanderberghe, "**Introduction to Applied Linear Algebra**", Cambridge University Press, 2018
2. G. Strang, "**The Fundamental Theorem of Linear Algebra**", *The American Mathematical Monthly*, vol. 100(9), pp. 848-855, 1993
3. G. Strang, "**Linear Algebra and Learning From Data",** Wellesley - Cambridge Press, 2019

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: A good working knowledge of basic notions of linear algebra as for example in [1]. Some proficiency in MATLAB.

Examination and grading: Grading is based on Homeworks, Written final exam, Short presentation based on a recent paper of Linear Algebra Algorithms for Big Data.

IE_AUT 4. Adaptive Control

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Andrea Serrani, The Ohio State University

e-mail: serrani.1@osu.edu

Important note: to be confirmed.

IE_AUT 5. Applied Causal Inference

Course Area: Information Engineering

Credits: 4 (16 hours)

Instructor: Prof. Reza Arghandeh, Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway

e-mail: rajo@hvl.no

This course is an overview of applied causal inference. The course starts with an introduction to causal inference. Then, we talk about moving from observation to intervention. We learn about directed acyclic graphs and non-parametric structural equation models to create causal models. Furthermore, we use various realistic examples to understand better the concepts we introduced in each chapter. By the end of this course, students will be able to develop familiarity with causal models for investigating a wide range of questions about the world works.

Goals:

- Information-era literacy: learning to be informed citizens, consumers, and hopefully producers of information.
- Understand sources of bias in data (a big challenge in data science).
- Understand the importance of the causal discovery.
- Understand the basics of causal models.
- Ask causal questions:
    - for the sake of science
    - for better decision-making in daily life.

Learning Objectives:

- Understand the difference of causal inference with statistics and machine learning.
- Translate scientific questions and background knowledge into a causal model.
- Understand properties of causal models.
- Begin to develop familiarity with the uses of causal models for investigating a wide range of questions about the real world.

References:

1. Github page: https://github.com/Ci2Lab/Applied_Causal_Inference_Course
2. Elements of Causal Inference, Book, 2017 (open access)

https://mitpress.mit.edu/books/elements-causal-inference

3. Causal Inference What If, Book, 2020 (open access)

[back to **Summary**]

https://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/

Schedule and room: May/June 2023. Lectures exact dates and times will be published in Class Schedule as soon as they are fixed by the instructor.

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: familiarity with basic statistics. Knowledge of graph theory also helps, but it is not a requirement.

Examination and grading: a final presentation or a take-home exam.

IE_AUT 6. Distributed Optimization and Applications

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Subhrakanti Dey, Signals and Systems, Uppsala University, Sweden

e-mail: Subhra.Dey@signal.uu.se

Aim:

The aim of this course is to introduce postgraduate students to the topical area of Distributed Optimization. As we enter the era of Big Data, engineers and computer scientists face the unenviable task of dealing with massive amounts of data to analyse and run their algorithms on. Often such data reside in many different computing nodes which communicate over a network, and the availability and processing of the entire data set at one central place is simply infeasible. One needs to thus implement distributed optimization techniques with message passing amongst the computing nodes. The objective remains to achieve a solution that can be as close as possible to the solution to the centralized optimization problem. In this course, we will start with some history on the origins of distributed optimization algorithms such as the Alternating Direction Method of Multipliers (ADMM), discuss its properties, and applications to both convex and non-convex problems, and explore distributed statistical machine learning methods, and finish with discussions on very recent and largely open areas such as networked optimization. This course will provide a glimpse into this fascinating subject, and will be of relevance to graduate students in Electrical, Mechanical and Computer Engineering, Computer Science students, as well as graduate students in Applied Mathematics and Statistics, along with students dealing with large data sets and machine learning applications to Bioinformatics.

Topics:

- Lectures 1-3: Precursors to distributed optimization algorithms: parallelization and decomposition of optimization algorithms (dual de- composition, proximal minimization algorithms, augmented Lagrangian and method of multipliers), The Alternating Direction Method of Multipliers (ADMM): (Algorithm, convergence, optimality conditions, stopping criteria, constrained convex optimization)
- Lectures 4-5: Applications of ADMM to machine learning problems: l1 norm problems, ADMM based methods for solving consensus and sharing problems, ADMM for non-convex problems and examples
- Lectures 6-8: Applications of distributed optimization to distributed machine learning, Federated Learning, distributed Newton methods

[back to **Summary**]

- Lectures 9-10: Networked Optimization (e.g. over a graph) and fully distributed optimization under communication constraints

References:

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, 3(1):1122, 2011.

[2] Dimitri Bertsekas and John N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.

[3] S. Boyd and L. Vandenverghe, Convex Optimization, Cambridge University Press.

[4] M. Zhu and S. Martinez, Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments, Springer, 2015.

Relevant recent papers will be referred to and distributed during the lectures.

Schedule and room: May/June 2023. Lectures exact dates and times will be published in Class Schedule as soon as they are fixed by the instructor.

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Advanced calculus, and probability theory and random processes.

Examination and grading: A project assignment for students in groups of 2 requiring about 20 hours of work.

IE_CSC 1. Bayesian Machine Learning

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Giorgio Maria Di Nunzio

e-mail: dinunzio@dei.unipd.it

Aim: The course will introduce fundamental topics in Bayesian reasoning and how they apply to machine learning problems. In this course, we will present pros and cons of Bayesian approaches and we will develop a graphical tool to analyse the assumptions of these approaches in classical machine learning problems such as classification and regression.

Topics:

- Introduction of classical machine learning problems.
  - Mathematical framework
  - Supervised and unsupervised learning
- Bayesian decision theory
  - Two-category classification
  - Minimum-error-rate classification
  - Bayes decision theory
  - Decision surfaces
- Estimation
  - Maximum Likelihood Estimation
  - Expectation Maximization
  - Maximum A Posteriori
  - Bayesian approach
- Graphical models
  - Bayesian networks
  - Two-dimensional visualization
- Evaluation
  - Measures of accuracy

References:

[1] J. Kruschke, Doing Bayesian Data Analysis: A Tutorial Introduction With R and Bugs, Academic Press 2010

[2] Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2007

[3] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification (2nd Edition), Wiley-Interscience, 2000

[back to **Summary**]

[4] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin, Learning from Data, AMLBook, 2012 (supporting material available at http://amlbook.com/support.html)

[5] David J. C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003 (freely available and supporting material at http://www.inference.phy.cam.ac.uk/mackay/

[6] David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 (freely available at http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=

[7] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012 (supporting material http://www.cs.ubc.ca/ murphyk/MLbook/)

[8] Richard McElreath, Statistical Rethinking, CRC Presso, 2015 (supporting material https://xcelab.net/rm/statistical-rethinking/)

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Basics of Probability Theory. Basics of R Programming.

Examination and grading: Homework assignments and final project.

IE_CSC 2. Learning from Networks

Course Area: Information Engineering

Credits: 5 (10 lectures, 2 hours each)

Instructor: Prof. Fabio Vandin

e-mail: fabio.vandin@unipd.it

Important note: Not offered in a.a. 2022/23

IE_CSC 3. Advanced topics in scientific and parallel programming with practical application to the CAPRI HPC infrastructure

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Giacomo Baruzzo, Department of Information Engineering, University of Padova

e-mail: giacomo.baruzzo@unipd.it

Aim: Provide basic skills for working on remote servers, using/developing parallel software and deploying it on a containerized computer server. The course gives basic introduction to modern computer architecture and to the most important parallel programming paradigms: Multi-threading, OpenMP, MPI and CUDA with examples (mostly Python and C++). The course covers basic tools to access and to interact with remote servers, to manage remote resources, and to manage jobs. The course introduces principles of software containerization from the perspective of users, providing practical examples of Singularity/Docker. The concepts discussed are applied to simple case of studies involving writing and/or running parallel programs using the CAPRI HPC infrastructure (256 cores, 6TB shared RAM and 2 GPU Nvidia P100) recently acquired by the University of Padova for research activities.

Topics:

1. How to use a computing server (application to CAPRI)
    a. Introduction to High Performance Computing (HPC hardware and architectures, HPC software, supercomputers)
    b. Job scheduling (slurm; writing a job; running, stopping and querying status of a job)
    c. The CAPRI queuing system and policy (CAPRI hardware and architecture; access to CAPRI and projects; execution queue; how to choose queue)
2. Containerization (Singularity)
    a. Overview of containerization (definition of containers and container daemon; Singularity and Docker software; containers vs virtual machines; advantages: re-usability and reproducibility, flexibility, efficiency; disadvantages: learning curve)
    b. Using container that have already been defined (running, stopping, and resuming containers; containers options and flags)
    c. Defining new containers (new containers from scratch; extending existing containers)
    d. Sharing containers and the container repository (browsing and adding a container to the repository; guidelines for creating and documenting containers to be shared)
3. Version control (git)
    a. Basic operations (create a git repository, staging and committing changes, repository status and history, work with branches)
    b. Advanced operations and remote repository (clone a remote repository, work with a remote repository, GUI for git, git web-based hosting services)

[back to **Summary**]

4. Parallel architectures and multi-process/parallel programming
   a. Introduction to parallel programming models and architectures (basic definitions; shared vs distributed memory architectures; threading: CPU and GPU; shared memory programming; message passing programming; performance metrics)
   b. Parallel programming languages and frameworks (multi-threading; OpenMP; MPI; CUDA)
5. Hands on example (a simple parallel software for data analysis / machine learning / numerical analysis; students' proposals)

References:

- Eijkhout, V. (2013). Introduction to High Performance Scientific Computing. Lulu. com.
- Grama, A., Kumar, V., Gupta, A., & Karypis, G. (2003). Introduction to parallel computing. Pearson Education.
- Parhami, B. (2006). Introduction to parallel processing: algorithms and architectures. Springer Science & Business Media.
- Ad-hoc material by Lecturer

Schedule and room: please, see Class Schedule

Course requirements: Basics usage of tools for run/develop of scientific software (preferable unix/linux platforms)

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Examination and grading: Each student must produce a small parallel and containerized software (either predefined or custom built container) related to her/his research field. Each student can either a) write a simple parallel software with one of the programming paradigm presented during the course using a language of choice or b) choose a (possibly parallel) software typically used in the research activity. Containerized software must run on the CAPRI server.

IE_CSC 4. Domain-Specific Accelerators

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Carlo Fantozzi

e-mail: carlo.fantozzi@unipd.it

Aim: The hitting of fundamental limits in computer hardware technology, at a time when the needs for computing performance and efficiency are increasing, is driving a shift from general-purpose processors to domain-specific accelerators (DSAs). This course will examine how the phenomenon is occurring by analyzing several DSAs at different levels: architectural, algorithmic, and software. The course will also explore whether DSAs can be effective for computations not originally intended by their designers. The course will give students, regardless of their research specialization, an elemental understanding of a key ongoing event in computer design, and will provide heavy users of computer systems with valuable elements to decide how to accelerate their scientific computations.

Topics:

- Introduction to DSAs. Applications. Advantages and challenges.
- Survey of DSAs (e.g., NVIDIA's Tensor Cores v1-v4, Google's Tensor Processing Units v1-v4, Intel's VNNI and AMX). DSAs implemented on FPGAs.
- Towards the design of algorithms and software for DSAs: computational models, software APIs.
- Algorithms for DSAs: dense linear algebra, integral transforms, dynamic programming. Reduced precision and mixed precision. Managing sparsity. Exploiting DSAs for computations not intended by their designers.
- From algorithms to software: case studies taken from the algorithms presented in previous lectures (e.g., matrix multiplication for linear algebra, convolution for integral transforms).

References:

No comprehensive book on DSAs has been published yet. The following references provide an overview of the topics covered in the course. Full references will be made available to students during lectures.

[1] Abdul Dakkak, et al. "Accelerating reduction and scan using tensor core units." Proceedings of the ACM International Conference on Supercomputing. 2019.

[2] Albert Reuther, et al. "Survey of machine learning accelerators." 2020 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2020.

[3] Asit Mishra, et al. "Accelerating sparse deep neural networks." arXiv preprint arXiv:2104.08378 (2021).

[back to **Summary**]

[4] Jens Domke, et al. “Matrix engines for high performance computing: A paragon of performance or grasping at straws?.” 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2021.

[5] John L. Hennessy, and David A. Patterson. Computer Architecture: A Quantitative Approach (6th Edition); Chapter 7. Morgan Kaufmann, 2017.

[6] Rezaul Chowdhury, Francesco Silvestri, and Flavio Vella. “A computational model for Tensor Core Units.” Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures. 2020.

[7] Suejb Memeti, et al. “Benchmarking OpenCL, OpenACC, OpenMP, and CUDA: programming productivity, performance, and energy consumption.” Proceedings of the 2017 Workshop on Adaptive Resource Management and Scheduling for Cloud Computing. 2017.

[8] William J. Dally, Yatish Turakhia, and Song Han. “Domain-specific hardware accelerators.” Communications of the ACM 63.7 (2020): 48-57.

Course requirements:

- Computer Architecture (intermediate knowledge).
- C/C++ programming (basic knowledge).
- Design and Analysis of Algorithms (basic knowledge).
- Microelectronics (a basic knowledge helps, but it is not mandatory).

Examination and grading: Each student must propose and turn in a survey report, or a software project, related to the material covered in class and exploring some architecture, algorithm, or application of DSAs. The final mark will be based on the grading of the report/project, with a small modifier to take class participation into account. Each student is encouraged to propose a report/project related to her/his research area.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

IE_OPT 1. Quantum Communication: methods and implementations

Credits: 5 (20 hours)

Instructors: Dr. Marco Avesani, Dr. Francesco Vedovato, University of Padova

e-mail: marco.avesani@unipd.it, francesco.vedovato@unipd.it

Aim: The course aims at giving an introduction to the methods and experimental techniques used in quantum communication. The main topic of the course will be Quantum Key Distribution since it offers the possibility to present a modern perspective on both theoretical (protocols, security proofs) and practical tools (source and detectors technologies, implementation schemes, and realizations) using the framework of photonic quantum communication technologies.

Topics:

- Elements of quantum communication and Quantum Key Distribution
- Discrete-variable (DV) Quantum Key Distribution: methods and protocols
- Experimental DV-QKD
- Continuous-variable Quantum Key Distribution: methods and protocols
- Experimental CV-QKD
- Advanced Protocols

References:

[1]S. Pirandola *et al.*, «Advances in quantum cryptography», *Adv. Opt. Photonics*, vol. 12, n. 4, pagg. 1012–1236, dic. 2020, doi: 10.1364/AOP.361502.

[2]N. Gisin, G. Ribordy, W. Tittel, H. Zbinden, e N. Gisin, «Quantum cryptography», *Rev Mod Phys*, vol. 74, n. 1, pagg. 145–195, mar. 2002, doi: 10.1103/RevModPhys.74.145.

[3]V. Scarani *et al.*, «The security of practical quantum key distribution», *Rev. Mod. Phys.*, vol. 81, n. 3, pagg. 1301–1350, 2009, doi: 10.1103/RevModPhys.81.1301.

Schedule and room: please, see Class Schedule

Enrollment: add the course to the list of courses you plan to attend using the Course Enrollment Form (requires SSO authentication) and, if you are taking the course for credits, to the Study and Research Plan.

Course requirements: Linear Algebra. Basics of Quantum Information and Quantum Optics may help

Examination and grading: Oral test on the contents of the course, with the possibility of presenting an essay on a topic, agreed with the teachers.

Alphabetical List of Course Instructors

[back to **Summary**]