# Course catalogue

## Computer Science Engineer

Data and Information Sciences mention (Metz),

Metz Campus of CentraleSupélec

*last update: September 24, 2025*

## Semester 09

| SD9 | | **SD9** | **8 ECTS** |
|---|---|---|---|
| 3MD1520 | 1 | C++ programming | 21.0 h |
| 3MD1540 | 1 | Machine learning 1 | 27.5 h |
| 3MD1510 | 1 | Software Application Engineering | 22.5 h |
| 3MD1530 | 1 | Statistical Models 1 | 29.0 h |

| SM10 | | **SM10** | **8 ECTS** |
|---|---|---|---|
| 3MD4040 | 1 | Deep learning | 28.5 h |
| 3MD4010 | 1 | Machine learning 2 | 21.5 h |
| 3MD4050 | 1 | Statistical models 2 | 33.0 h |
| 3MD4030 | *1* | *HPC-HPDA on GPU with CUDA* | *22.5 h* |
| 3MD4020 | *1* | *Sparse models* | *18.5 h* |

## Semester 10

| SM11 |  | **SM11** | **8 ECTS** |
|---|---|---|---|
| 3MD4120 | 1 | Reinforcement learning | 23.0 h |
| 3MD4150 | 1 | Natural language processing | 29.0 h |
| 3MD4140 | 1 | Statistical learning | 23.0 h |
| 3MD4130 | 1 | Big Data computation models | 21.0 h |
| 3MD4110 | 1 | Perspectives in Learning and Artificial Intelligence | 23.5 h |

## C++ PROGRAMMING

**Course supervisor:** Hervé Frezza-Buet, Frédéric Pennerath

**Total:** 21.0 h **TP:** 18.0 h

3MD1520 *back*

**Description:** Knowing how to code an algorithm effectively in a given programming language requires a prior understanding of the associated calculation model and how the instructions in that language are translated into machine instructions. Too many students still approach programming in a superficial and risky way, lacking the basic knowledge necessary to write elegant and effective code. The unique strength of the C++ language is to allow the production of compiled codes close to the optimal machine code while offering different high-level programming approaches such as strong typing, object programming, functional programming and meta-programming (automatic code generation at compilation). For this reason, C++ has become the essential language for developing optimized algorithms. Its only disadvantage is its richness, which has continued to grow in its most recent versions (C++11/14/17/20) and which makes it difficult to understand the language in its entirety without adequate training. This course is intended for students, including beginners, who want to master the different aspects of C++ programming in order to be able to write code that combines performance and elegance. The course adopts a bottom-up approach starting from the mechanisms of elementary program execution and gradually moving towards the most advanced language functionalities.

**Content:** 6 practical sessions (3h each), and an individual exam on the machine (2h). The program covers the compilation chain, memory management (stack and heap), types, inheritance, genericity (templates) and system programming.

**Learning outcomes:** Know how to write a program in C++ using different programming paradigms such as object programming, functional programming and generic programming. To know certain aspects of the C++ language that have a decisive influence on the performance of programs during their execution. Be familiar with the functionalities offered by the most recent specifications of the C++ language (C++11, C++14, C++17, C++20). Know how to use a C++ compilation and debugging environment

**Teaching methods:** The objective is to transmit to students a real programming know-how, on the one hand by illustrating the concepts through relevant examples of use, and on the other hand by devoting a significant part of the hourly volume to laboratory work.

**Means:** Practical works under Linux PC / g++ Max. 40 students 2 students per machine max.

**Evaluation methods:** 3-hour individual computer test, can be retaken.

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world

## Machine learning 1

**Course supervisor:** Hervé Frezza-Buet

**Total:** 27.5 h **CM:** 13.5 h, **TP:** 12.0 h

3MD1540 *back*

**Description:** This course sets out the general framework of machine learning, allowing you to situate the different approaches in the field. It covers the notions of data pre-processing, an introduction to statistical learning theory (risks, overlearning, convex proxies, regularization), the difference between frequentist and Bayesian approaches, supervised, unsupervised, semi-supervised and reinforcement learning paradigms. Some approaches are detailed (Kernel methods, SVM, Boosting, Bagging, Decision trees...).

**Learning outcomes:** At the end of this course, students will be able to recognize the different classes of algorithms in the landscape of the many methods available on the shelf. They will also have the statistical notions that will enable them to make reasoned use of these methods, thus avoiding a black-box approach with blind parameter testing.

**Means:** The courses and practical work are given by Hervé Frezza-Buet, Arthur Hoarau, Jérémy Fix. The courses present theoretical aspects, mathematical proofs, but are also illustrated by demonstrations of algorithms. The practical work will be done in Python, using sickit-learn, in pairs.

**Evaluation methods:** 2h written test, can be retaken.

**Evaluated skills:**

- Analyze, design, and build complex systems with scientific, technological, human, and economic components
- Develop in-depth skills in an engineering field and a family of professions

## Software Application Engineering

**Course supervisor:** Virginie Galtier

**Total:** 22.5 h **CM:** 6.0 h, **TD:** 7.5 h, **TP:** 9.0 h

3MD1510 *back*

**Description:** The application architecture of a system defines its software components and the exchanges between them, as well as their implementation on the hardware architecture. It naturally stands at the crossroads of development and operations missions. This course provides a typology of the main application architectures and presents the concepts of "devops", a set of practices that facilitate the automation of software delivery. Some concepts will be implemented on an illustrative architecture.

**Content:** Introduction to devops (software life cycle, agility, CI CD pipeline, build, versioning, tests, containers, infrastructure as code): principles and overview of some tools Typology of application architectures and associated technologies, middleware (focus on MOM)

**Prerequisites:** Students are expected to review their 1rt year SIP course before taking this one. They must also be comfortable in Java (self studies are provided), and with Git and Linux.

**Learning outcomes:** At the end of this course, students will be able to choose the application architecture best suited to their project and will be able to implement a software solution based on message queuing. They will be able to quickly integrate into an application development and deployment chain and will be familiar with the use of a few tools.

**Teaching methods:** General concepts will be presented in class. Practical, guided tutorials will then enable students to familiarize themselves with the associated approaches and tools. Finally, they will apply their new knowledge and skills to a project that will be enriched over the course of the sessions, and which will serve as an example for the evaluation project carried out independently.

**Means:** The lab works use free software that can be installed directly on students' personal computers. A prepackaged container is also provided. Students take advantage of the clusters of the campus for some deployments.

**Evaluation methods:** Knowledge and skills will be assessed on the basis of a few individual written tests taking place during the lab sessions and a project

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world

## Statistical Models 1

**Course supervisor:** Frédéric Pennerath

**Total:** 29.0 h **CM:** 16.5 h, **TD:** 4.5 h, **TP:** 6.0 h

3MD1530 *back*

**Description:** The ”Statistical Modeling” courses ModStat1 and ModStat2 deal with the modeling of systems for which the outputs are sufficiently uncertain that they need to be modeled by random variables. The course begins with a review of statistics and the introduction of elementary models (e.g. naive Bayes, linear regression, etc.), moving progressively towards more complex models. While the courses present the most useful elementary models and methods in this modeling context, they are not intended to be an exhaustive catalog. The aim is rather to present, within a consistent theory, the concepts and tools common to all these models and methods, and to show how, starting from modeling hypotheses specific to each concrete problem, these concepts are logically assembled before leading to an operational method. From a practical point of view, the aim of this course is not only to give students the means to understand and make good use of existing model implementations, but also to design their own implementations to take into account the specificities of a given problem. The course focuses on linking theory to practice: first, the hypotheses associated with a given class of problems are identified in class, followed by theoretical modeling work, leading to the definition of a model and its estimation algorithms. These results are then applied to a case study in tutorial sessions, before being implemented (in Python) and evaluated on data in practical exercises. The ModStat1 course will introduce the basic tools of statistical modeling, while the ModStat2 course will focus on hidden variable models.

**Prerequisites:** - Basic knowledge of probability theory, statistics and machine learning - Beginner level in Python / Numpy programming

**Learning outcomes:** - Be able to choose a statistical model/method adapted to the problem under consideration and implement it appropriately - Be able to understand the theoretical concepts underlying a statistical inference method presented in a scientific article. - Be able to implement a model / statistical method in a language such as Python. - Be able to adapt a model/method to take into account the specificities of the problem being addressed.

**Evaluation methods:** 2h written test without documents, can be retaken.

**Evaluated skills:**

- Analyze, design, and build complex systems with scientific, technological, human, and economic components
- Develop in-depth skills in an engineering field and a family of professions

## Deep learning

**Course supervisor:** Jérémy Fix

**Total:** 28.5 h **CM:** 13.5 h, **TP:** 15.0 h

3MD4040 *back*

**Description:** Deep learning is a technology that is booming, thanks in particular to the use of GPUs (Graphical Processing Units), the availability of large amounts of data and the understanding of theoretical elements that make it possible to better define neural network architectures that are more easily trainable. In this course, students will be introduced to the basics of neural networks and also to the different architectural elements that make it possible to design a neural network according to the prediction problem considered. The course is divided into modules in which questions of optimization algorithms, their initialization, regularization techniques, fully connected architectures, convolutional networks, recurrent networks, introspection techniques are addressed. Practical works on GPUs are associated with the courses.

**Content:** The lectures will be discussing :

- Historical introduction to neural networks, linear classifier/regressor (1.5 HPE) - Computational graph and gradient descent, Fully connected networks, RBFs, Auto-encoders, Optimisation methods, initialisation, regularisation (3 HPE) - Convolutional networks: architectures (1.5 HPE) - Convolutional networks: classification, object detection, semantic segmentation (1.5 HPE) - Recurrent networks: architectures and training (1.5 HPE) - Recurrent networks: applications (1.5 HPE) - Transformer approaches (1.5 HPE) - Generative models: autoregressive models, VAE, GANs and diffusion (1.5 HPE)

The praticals will be on:

- Introduction to pytorch on classification with linear predictors, fully connected networkjs and convolutional networks (3 HPE) - Convolutional neural networks for semantic segmentation (3 HPE) - Recurrent neural networks applied to sequence to sequence translation (3 HPE) - Adversial neural networks (3 HPE) - Self-supervized learning (3 HPE)

**Learning outcomes:** Being able to implement and deploy a deep learning algorithm Being able to choose the right architecture that suits a particular machine learning problem Being able to diagnose the training of a neural network (what is it learning ? how is it learning ? is it learning ? will it be able to generalize ?)

**Teaching methods:** The course is structured into lectures during which we introduce the theoretical and experimental notions are introduced and illustrated with various examples. Practicals allow the students to put into practice the notions we discuss during the lectures.

**Means:** We will be using the Pytorch framework. The students could work in pairs and will make use of GPUs of the Data Center d"Enseignement of the Metz campus for running their codes. A page will be dedicated on edunao. Forums will be opened, allowing the students to ask questions on the lectures or the tutorials, having the possibility to interact with the teaching staff but between them as well.

**Evaluation methods:** The assessment will be based on two elements: assessments on paper questionnaires at the beginning of each practical session and participation to a dedicated challenge in a team. The grade depends both on their submission, and a recorded video explaining their approach and results.

**Evaluated skills:**

- Act, initiate, innovate in science and technology environment
- Be operational, responsible, and innovative in the digital world
- Demonstrate ethical, responsible, and honest engineering practice

**External resources:**

- Site du cours
- Sujets des TPs

## Machine learning 2

**Course supervisor:** Arthur Hoarau

**Total:** 21.5 h **CM:** 10.5 h, **TP:** 9.0 h

3MD4010 *back*

**Description:** This course complements Machine Learning 1 with notions of data processing (dimension reduction, etc.), unsupervised learning, active and semi-supervised learning, explicability issues.

**Learning outcomes:** By the end of this course, students will have completed their breadth approach to machine learning.

**Evaluation methods:** 2h written test, can be retaken.

**Evaluated skills:**

- Analyze, design, and build complex systems with scientific, technological, human, and economic components
- Develop in-depth skills in an engineering field and a family of professions

## Statistical models 2

**Course supervisor:** Frédéric Pennerath

**Total:** 33.0 h **CM:** 12.0 h, **TD:** 6.0 h, **TP:** 12.0 h

3MD4050 *back*

**Description:** This course is an extension of the ModStat1 course. It is structured around the three fundamental concepts of stochastic processes, latent variables and approximate inference techniques. The first part of the course on processes focuses on three main families of processes: point processes, Markov processes and Gaussian processes. The notion of latent variable is then addressed through mixture models and the EM algorithm. The two notions are then combined to develop hidden Markov models, for both discrete (HMM) and continuous states (Kalman filters). Finally, approximate inference techniques are presented, with sampling techniques (MCMC) and variational inference.

**Prerequisites:** - Having followed the course ”Statistical Models 1” - Beginner level in Python / Numpy programming

**Learning outcomes:** At the end of this course, students will be able to associate the corresponding type of stochastic process with data series and apply the associated estimation methods. They will also be able to specify a model incorporating hidden variables and apply the EM algorithm to estimate its parameters. They will be able to model a clustering problem in the form of a mixture model. They will be able to specify an HMM or a Kalman filter to model the dynamic behavior of a discrete or continuous state system.

**Evaluation methods:** 3h written test with documents, can be retaken.

**Evaluated skills:**

- Analyze, design, and build complex systems with scientific, technological, human, and economic components
- Develop in-depth skills in an engineering field and a family of professions

## HPC-HPDA on GPU with CUDA

**Course supervisor:** Stéphane Vialle

**Total:** 22.5 h (optional) **CM:** 6.0 h, **TD:** 4.5 h, **TP:** 12.0 h

3MD4030 *back*

**Description:** This course aims to introduce high performance algorithmics and programming on GPU, with experiments on Machine Learning algorithms run on GPU servers.

**Content:** GPU architecture Algorithmic principles of fine grained GPU parallelism (SIMD and SIMT models) CUDA programming Usage of CUBLAS library Optimization of GPU and CPU-GPU CUDA codes Design and experiment of K-means algorithms on GPU

**Prerequisites:** 1st year common course: ”Systèmes d'Information et Programmation” (1CC1000) 1st year common course: ”Algorithmique & Complexité” (1CC2000) C++ Advanced programming course (3MD1020) of SDI mention at Metz Automatic Learning course (3MD1040) of SDI mention at Metz

**Learning outcomes:** At the end of this course, students will be able: Learning Outcome AA1: to analyse the adequacy of a mathematical solution with an implementation and execution on GPU, Learning Outcome AA2: to design a GPU algorithm, or to adapt an algorithm to increase its efficiency on GPU, Learning Outcome AA3: to design hybrid algorithms for CPU-GPU systems, overlapping data transfers and computations, Learning Outcome AA4: to implement algorithms and to debug codes on GPU, Learning Outcome AA5: to analyse and to summarize GPU software.

**Means:** Development and execution platform: GPU servers of the Data Center for Education of CentraleSupélec Metz Campus. NVIDIA CUDA development environment.

**Evaluation methods:** Evaluation of Lab results about parts 2 and 3, and final and individual exam Reports of the Lab about parts 2 and 3 (the content and the number of pages of the reports are constrained, in order to force the students to an effort of synthesis and clarity) In the event of unjustified absence from a practical work, the mark of 0 will be applied, in the event of justified absence the average mark of other labs will be applied. The remedial exam will be a 1 hour oral exam, which will constitute 100

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world
- Know how to convince

## Sparse models

**Course supervisor:** Laurent Duval

**Total:** 18.5 h (optional) **CM:** 6.0 h, **TP:** 12.0 h

3MD4020 *back*

**Description:** The course introduces principles behind data transformation and optimization methods at the heart of automatic learning and data science, from the perspective of sparsity and robustness, applied to digital data compression (mp3, jpg) and representations by predictive models, making extensive use of algorithmic experimentation, intuition and history of science.

**Content:** The course presents a travel through data analysis and learning, via different sparse tools and methods, aimed at explaining observations by a reduced number of parameters: data metrics, descriptors and transformations (norms, vector bases and frames, wavelets); implementation through data compression algorithms (audio, image, video, text); extension to prediction models (statistical moments, linear and polynomial regressions, parsimonious or robust models).

**Learning outcomes:** To understand practical and theoretical motivations of optimization algorithms used in automatic learning and data science. To implement related algorithms in an adapted manner by understanding their meaning in relation to the problem at hand. To link different methods and implement them in a data processing flow.

**Teaching methods:** On each theme, students are first confronted with a ”toy” problem for which they must mobilize their knowledge, ask themselves questions and implement first algorithms (in pairs). In a second step, after an exchange on this first phase, theoretical aspects, mathematical proofs and algorithmic tools are presented. Finally, in a third part, students apply these skills to a more complex problem.

**Means:** Courses and practical work are given by Laurent Duval (ESIEE-Paris, Université Paris-Est Marne-la-Vallée and IFP Energies nouvelles). Courses and practical work are intertwined, using signals, images or experimental data ranging from simple simulations to real-world data.

**Evaluation methods:** The module will be evaluated by an oral examination in groups of two or three students, with a report provided in advance, on an integrative topic, designed to mobilize different skills and methods acquired during the course. If the number of students allows it, a project-type structure, allowing groups to collaborate, will be proposed.

**Evaluated skills:**

- Act, initiate, innovate in science and technology environment
- Demonstrate ethical, responsible, and honest engineering practice
- Lead a project, a team

## Reinforcement learning

**Course supervisor:** Hervé Frezza-Buet

**Total:** 23.0 h **CM:** 9.0 h, **TD:** 3.0 h, **TP:** 9.0 h

3MD4120 *back*

**Description:** The course presents the theoretical foundations of reinforcement learning as well as the principles of the most common algorithms. Through practical work, these elements will be extended to more complex situations, making it possible to introduce the most recent algorithms that have, for example, enabled computers to master the game of Go.

**Content:** Reinforcement learning is introduced using the formal framework of the Markov Decision Processes. After having shown the existence and uniqueness of a solution in the form of the value function, we will discuss the classical algorithms used to calculate this function. We will then see how approximate methods (linear approximation, monte carlo estimation, bandits, deep learning) can be used to tackle more complex contexts.

**Learning outcomes:** Understand the theoretical foundations of reinforcement learning. Implement these methods in a way that is adapted to the problems to be solved. Sharpen your critical thinking skills.

**Teaching methods:** Taking into account the context (group size), lectures will be as interactive as possible andd will aim to present the theoretical and algorithmic concepts underlying reinforcement learning. The purpose of the practical work is to really confront the methods by implementing and testing the algorithms to better understand how they work and their limitations.

**Means:** Courses and practical work are provided by Alain DUTECH, Hervé FREZZA-BUET and Jérémy FIX. The practical work will be based on the Python language and its scientific libraries.

**Evaluation methods:** The module will be evaluated by a written exam, where the idea is to test the student's ability to use methods in a clever way, to analyze the results of an algorithm, etc.

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world

## Natural language processing

**Course supervisor:** Joël Legrand

**Total:** 29.0 h **CM:** 9.0 h, **TP:** 18.0 h

3MD4150 *back*

**Description:** This course explores the fundamentals of automatic natural language processing (ANNP), covering topics such as word embeddings, language models, recurrent and recursive neural networks and transformers, enabling students to master text analysis and generation.

**Content:** This course introduces the main linguistic theories used to model natural language (e.g. formal grammars, dependency grammars, etc.). It presents the various natural language processing (NLP) tools available and the statistical models on which they are based. Particular emphasis will be placed on the deep learning methods that constitute the state of the art for most NLP tasks.

**Learning outcomes:** By the end of this course, participants will have acquired a thorough understanding of the fundamental concepts of NLP. They will be able to apply text preprocessing techniques to clean and organize linguistic data, as well as use pre-trained language models for various tasks such as text classification, text generation, machine translation. Learners will be proficient in the use of popular natural language processing libraries such as NLTK, SpaCy, Transformers.

**Teaching methods:** Each session will include a lecture part during which new concepts will be introduced, followed by a practical work session. Practical work sessions will be direct applications concepts seen in lectures. All teaching materials will be provided to students.

**Evaluation methods:** 2h written test, can be retaken.

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world

## Statistical learning

**Course supervisor:** Michel Barret

**Total:** 23.0 h **CM:** 10.5 h, **TD:** 10.5 h

3MD4140 *back*

**Description:** The objective of supervised learning is to propose methods that, based on a training set of examples, make a decision on a parameter based on observations, the decision being the best possible on average. For example, classify images according to their content, i.e. decide if an image represents a cat, a dog, or something else. We will formally present the problem and study the guarantees of generalization of supervised learning algorithms, i.e. th3e quality of prediction of the output associated with an entry not present in the training set. To achieve this objective, we will introduce the concepts of hypothesis space with PAC (probably approximately correct) learning capacity , Vapnik-Chervonenkis dimension of a hypothesis space. We will state and prove two fundamental theorems of supervised learning theory giving a lower bound and an upper bound of the real risk to the binary classification problem.

**Content:** Formalization of supervised learning problems PAC learning capacity and uniform convergence The bias-complexity trade-off The VC (Vapnik-Chervonenkis) dimension of a hypothesis space Two fundamental theorems of PAC learning

**Learning outcomes:** At the end of this course, students will be able: -to understand elements of the theory of supervised learning; -to understand the bias-complexity trade-off of an hypothesis class; -to understand and use PAC bayesian bounds of supervised learning (in particular those of binary classification problem).

**Teaching methods:** 10,5h of courses + 10,5h of tutorials + written exam of 2h

**Means:** The tutorials (TDs), consisting of exercises, will allow the concepts seen in class to be used.

**Evaluation methods:** Written exam of 2h with documents

**Evaluated skills:**

- Analyze, design, and build complex systems with scientific, technological, human, and economic components

## Big Data computation models

**Course supervisor:** Stéphane Vialle

**Total:** 21.0 h **CM:** 10.5 h, **TD:** 1.5 h, **TP:** 9.0 h

3MD4130 *back*

**Description:** The goal of this course is to teach students how to develop high-performance data analysis applications in the Spark environment on distributed platforms (clusters and clouds). Distributed file system mechanisms such as HDFS will be studied, as well as Spark's extended map-reduce programming model and algorithmic on top of Spark ”RDD”, followed by higher-level programming models on top of Spark ”Data Frames”, and finally programming models on Clouds. Scaling criteria and metrics will also be studied. Throughout the course, implementations will take place on clusters and in a Cloud, and the developed solutions will be evaluated by the performance obtained on test cases, and by their ability to scale.

**Content:** Emergence of Big Data technologies: motivations, industrial needs, main players. Hadoop software stack, architecture and operation of its distributed file system (HDFS) Spark distributed computing architecture and deployment mechanism Spark ”RDD” programming model and algorithmics of Spark extended map-reduce Spark ”Data Frames” programming model applied to graph analysis (GraphX module) Architecture et environnement d'analyse de données sur Cloud Experiments and performance measures Performance criteria and metrics

**Learning outcomes:** After this course, students will be able:

Learning Outcome AA1: to design and implement extended map-reduce algorithms, powerful and scaling on distributed platforms, Learning Outcome AA2: to analyse the scaling capabilities of an application, Learning Outcome AA3: to use a cluster or a cloud to achieve large scale data analysis, Learning Outcome AA4: to synthetically present a data analysis solution designed on top of a ”map-reduce” model.

**Teaching methods:** This course links 3 parts relating to ”Big Data” computing models: the first on PC clusters, the second in the Cloud, and the third which assesses ”scaling-up” solutions.

Course plan in 4 parts:

Part 1: Software architecture and development with Spark RDD on top of HDFS and PC clusters.

Part 2: Criteria and metrics for performance and scaling.

Part 3: Large scale computation and data analysis on Cloud.

Part 4: Development with Spark Data Frames on top of HDFS and PC clusters.

**Means:** Teaching team: Stéphane Vialle and Gianluca Quercini (CentraleSupelec), Wilfried Kirschemann (ANEO) Development and execution plateform: computing clusters of the Data Center for Education (DCE) of CentraleSupelec Metz campus access to a professional cloud Development environment: Spark+HDFS on DCE machines other environment on Cloud ressources

**Evaluation methods:** Evaluation from Labs:

The reports of the Labs will be evaluated (the content and the number of pages of the reports will be constrained, in order to force the students to an effort of synthesis and clarity) In case of unjustified absence from a practical course, a mark of 0 will be applied; in case of justified absence, the practical course will not be included in the final mark. The remedial exam will be a 1 hour written exam, which will constitute 100

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world
- Know how to convince

## Perspectives in Learning and Artificial Intelligence

**Course supervisor:** Joël Legrand

**Total:** 23.5 h **CM:** 8.5 h, **TP:** 15.0 h

3MD4110 *back*

**Description:** This module offers insight into real-world applications of machine learning through talks by researchers and industry professionals. Each session highlights a specific application domain (healthcare, finance, energy, robotics, etc.), illustrating how classical or advanced techniques are used to address real-world problems. These talks complement academic instruction by exposing students to more advanced approaches at times, thereby fostering a broader understanding of current challenges and practices in professional or research settings.

**Learning outcomes:** At the end of this module, students will have gained a broad perspective on real-world applications of machine learning across various sectors. They will be able to analyze real-world problems through the lens of AI, understand the methodological choices made by experts, and identify the specific constraints related to implementing solutions in industrial or research contexts. This module will also enhance their ability to engage in dialogue with professionals in the field and to envision themselves contributing to interdisciplinary projects involving machine learning.

**Means:** Each module (5 in total) is given by an industrialist or a researcher. It consists of a 1h30 lecture followed by a 3h practical session.

**Evaluation methods:** At the end of each practical session, a submission will be required by the instructor. One submission will then be randomly selected for evaluation by the instructor. The grade given will serve as the assessment for the entire module.

**Evaluated skills:**

- Be operational, responsible, and innovative in the digital world