Plan d’études

Master en Cybersécurité

2024-2025

arrêté par la vice-présidence académique de l'EPFL le 20 juin 2024

Directeur de section	**Prof. M. Payer**
Adjointe du directeur de section	**Mme E. Hazboun**
Coordinatrice des stages en entreprise	**Mme P. Genet**
Spécialiste administrative	**Mme J. Locatelli**

Aux cycles bachelor et master, selon les besoins pédagogiques, les heures d’exercices mentionnées dans le plan d’études pourront être intégrées dans les heures de cours ; les scolarités indiquées représentent les nombres moyens d’heures de cours et d’exercices hebdomadaires sur le semestre

2024-2025 INFORMATIQUE - CYBERSÉCURITÉ - Obligatoire — Cycle Master

Code	Matières	Enseignants sous réserve de modification	depth requirement**	Semestres MA1/MA3 c	e	p	MA2/MA4 c	e	p	Crédits	Période des épreuves *	Type examen **
Groupe "breadth requirement et depth requirement and options"										**72**		
Groupe 1 "breadth requirement"										**min. 32**		
CS-470	Advanced computer architecture	Ienne					**3**		**2**	8	E	écrit
CS-523	Advanced topics on privacy enhancing technologies	Troncoso	x				**3**	**1**	**2**	8	E	écrit
CS-450	Algorithms II	Kapralov/Svensson		**4**	**2**					8	H	écrit
COM-401	Cryptography and security	Vaudenay	x	**4**	**2**					8	H	écrit
CS-438	Decentralized systems engineering	Borso/Ford		**2**	**2**	**2**				8	H	écrit
CS-451	Distributed algorithms	Guerraoui		**2**	**1**	**3**				8	H	écrit
CS-452	Foundations of software	Bourgeat					3	2	2	8	E	écrit
COM-402	Information security and privacy	Payer	x	**3**	**1**	**2**				8	H	écrit
CS-433	Machine learning	Jaggi/Flammarion		**4**	**2**	**2**				8	H	écrit
CS-460	Systems for data management and data science	Ailamaki/Kermarrec					**2**	**2**	**2**	8	E	écrit
COM-407	TCP/IP networking	Nikolopoulos		**2**	**2**	**2**				8	H	écrit
ETHZ	ETHZ courses counting as breadth requirement											
Groupe 2 "depth requirement and options" (la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)												
Bloc "Projet et SHS" :										**18**		
CS-496	Research project in Cyber security	Divers enseignants		← **12** →						12	sem A ou P	
HUM-nnn	SHS : introduction au projet	Divers enseignants		**2**		**1**				3	sem A	
HUM-nnn	SHS : projet	Divers enseignants							**3**	3	sem P	
	Total des crédits du cycle master :									**90**		

Remarque :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** Cf. à l'art. 9 al. 2 du règlement d'application

Stage d'ingénieur :

Voir les modalités dans le règlement d'application

INFORMATIQUE - CYBERSÉCURITÉ - Options

Code	Matières	Enseignants sous réserve de modification	depth requirement***	Semestres MA1/MA3			MA2/MA4			Crédits	Nbre places	Période des épreuves *	Type examen**
				c	e	p	c	e	p				
Groupe 2 "depth requirement and options"													
CS-500	AI product management	Kaboli/Zamir		2	2	3				6		sem A	sans retrait
CS-420	Advanced compiler construction	Schinz					2		2	6		sem P	
CS-440	Advanced computer graphics (pas donné en 24-25)	Jakob					2	1	1	6		sem P	
COM-501	Advanced cryptography	Vaudenay	x				2	2		6		E	écrit
CS-471	Advanced multiprocessor architecture	Falsafi		4		8				8		sem A	
CS-477	Advanced operating systems	Kashyap		2	1	2				6		H	écrit
COM-417	Advanced probability and applications	Shkel		4	2					8		H	écrit
EE-512	Applied biomedical signal processing	Lemay		2		2				4		H	écrit
MATH-493	Applied biostatistics	Goldstein					2	2		5		sem P	
CS-401	Applied data analysis	Brbic		2		2				8		H	écrit
EE-554	Automatic speech processing	Magimai Doss		2	2					4		H	écrit
MICRO-452	Basics of mobile robotics	Mondada		1		3				4		H	écrit
MGT-416	Causal inference ****	Kiyavash					2	1		4		sem P	
MATH-352	Causal thinking	Stensrud		2	2					5		H	écrit
BIO-105	Cellular biology and biochemistry for engineers	Zufferey		2	2					4		H	écrit
CS-524	Computational complexity	Göös		3	1					6		H	écrit
NX-465	Computational neuroscience : neural dynamics	Gerstner					2	2		5		E	écrit
CS-413	Computational photography	Süsstrunk					2		2	6		sem P	
CS-442	Computer vision	Fua					2	1		6		E	écrit
CS-453	Concurrent computing	Guerraoui		2	1	2				6		H	écrit
COM-480	Data visualization	Vuillon					2		2	6		sem P	
EE-559	Deep learning	Cavallaro					2	2		4	150	sem P	sans retrait
CS-456	Deep reinforcement learning	Gulcehre					2	1	1	6		E	écrit
CS-472	Design technologies for integrated systems	De Micheli		3		2				6		sem A	
CS-411	Digital education	Dillenbourg/Jermann					2		2	6		E	écrit
CS-423	Distributed information systems	Aberer		2	1	1				6		H	écrit
ENG-466	Distributed intelligent systems	Martinoli		2	3					5	40	H	oral sans retrait
COM-502	Dynamical system theory for engineers ****	Thiran P.					3	1		6		E	écrit
CS-476	Embedded systems design	Kluter					2		2	6		sem P	
DH-415	Ethics and law of AI	Rochel		2	2					4	100	sem A	sans retrait
CS-489	Experience design (pas donné en 24-25)	Huang		2		4				6		sem A	
CS-550	Formal verification	Kuncak	x	2	2	2				6		sem A	
CS-459	Foundations of probabilistic proofs	Chiesa	x	4	1					6		sem A	
MATH-483	Gödel and recursivity ****	Duparc		2	2					5		H	écrit
MICRO-511	Image processing I	Unser/Van De Ville		3						3		H	écrit
MICRO-512	Image processing II	Unser/Van de Ville/ Liebling/Sage					3			3		E	écrit
CS-487	Industrial automation	Tournier/Sommer					2		1	3	36	E	oral sans retrait
COM-404	Information theory and coding	Telatar		4	2					8		H	écrit
CS-428	Interactive theorem proving	Barrière/Pit-Claudel					2	1	2	6		sem P	
CS-491	Introduction to IT consulting	Regev					6			6		E	oral
COM-406	Foundations of Data Science	Gastpar/Urbanke		4	2					8		H	écrit
CS-430	Intelligent agents	Faltings		3	3					6		sem A	
CS-486	Interaction design	Pu					2	1	1	6		sem P	
CS-431	Introduction to natural language processing	Rajman/Chappelier/Bosselut		2	2					6		H	écrit
CS-479	Learning in neural networks	Gerstner					2	1	1	6		E	oral
CS-526	Learning theory	Macris					2	2		6		E	écrit
CS-421	Machine learning for behavioral data ****	Käser					2		2	6		E	écrit
MGT-427	Management de projet et analyse du risque	Wieser		2	1					4		sem A	sans retrait
COM-516	Markov chains and algorithmic applications (pas donné en 24-25) ****	Lévêque/Macris		2	1	1				6		H	écrit
COM-405	Mobile networks	Al Hassanieh	x	3	2	2				8		A	écrit
COM-430	Modern digital communications: a hands-on approach	Chiurtu		2		2				8		sem A	
EE-452	Network machine learning	Frossard/Thanou					2		2	4		sem P	
COM-512	Networks out of control (pas donné en 24-25) ****	Thiran P./ Grossglauser					2	1		6		E	écrit
MATH-489	Number theory II.c - Cryptography	Jetchev	x				2	2		5		E	écrit
CS-439	Optimization for machine learning	Flammarion					2	2	1	8		E	écrit
CS-596	Optional research project in computer science II	Divers enseignants		2						8		sem A ou P	
CS-522	Principles of computer systems	Argyraki/Candea		4						8		sem A	écrit
CS-412	Software security	Payer	x				3	2	1	8		sem P	
PHYS-512	Statistical physics of computation	Erba		2	2					4		H	écrit
COM-500	Statistical signal and data processing through applications	Ridolfi					3	2		8		E	écrit
COM-506	Student seminar: security protocols and applications***	Vaudenay	x				2			3		sem P	
CS-448	Sublinear algorithms for big data analysis ****	Kapralov					2	1		6		sem P	
CS-473	System programming for Systems-on-Chip	Kluter		2		2				6		sem A	
CS-510	Topics in software security***	Payer	x	1	1					3		sem A	
CS-455	Topics in theoretical computer science (pas donné en 24-25) ****	Kapralov		3	1					6		sem A	
CS-444	Virtual reality	Boulic					2		1	6		sem P	
ETHZ	ETHZ courses counting as options												
ETHZ	ETHZ courses counting as depth requirement		x										

Remarque :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** sans retrait = pas de retrait possible après le délai d'inscription

*** Cf. à l'art. 9 al. 2 du règlement d'application

RÈGLEMENT D’APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION D’INFORMATIQUE POUR LE MASTER EN INFORMATIQUE - CYBERSÉCURITÉ

Année académique 2024-2025

du 20 juin 2024

La Vice-présidence académique,

vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004,

vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015,

vu le plan d’études de la section d’Informatique pour le master en Informatique - Cybersécurité,

arrête:

Art. 1 – Champ d'application

Le présent règlement fixe les règles d’application du contrôle des études de master en Informatique - Cybersécurité, pour l’année académique 2024-2025.

Art. 2 – Étapes de formation

Le master en Informatique - Cybersécurité est composé de deux étapes successives de formation:

- le cycle master, d’une durée de 3 semestres, dont la réussite implique l’acquisition de 90 crédits, condition pour effectuer le projet de master,

- le projet de master, d’une durée de 17 ou 25 semaines, dont la réussite se traduit par l’acquisition de 30 crédits.

Art. 3 – Sessions d’examen

1 Les branches de session sont examinées pendant les sessions d’hiver ou d’été (mention H ou E dans le plan d’études).

2 Les branches de semestre sont examinées pendant le semestre d’automne ou le semestre de printemps (mention sem A ou sem P).

3 Pour les branches de session, l’examen indiqué pour la session peut être complété par des contrôles de connaissances durant le semestre, selon les indications de la personne enseignante.

Art. 4 – Prérequis

Certains enseignements peuvent exiger des prérequis qui sont mentionnés dans la fiche de cours concerné. Le cours prérequis est validé si les crédits correspondants ont été acquis pour le cours ou par moyenne du bloc.

Art. 5 – Conditions d’admission

1 Les titulaires du bachelor EPFL en Informatique ou en Systèmes de communication, et les titulaires du bachelor ETHZ en Informatique, sont admis automatiquement.

2 Pour les titulaires d’autres bachelors, l’admission s’effectue sur dossier.

3 Une candidate ou un candidat refusé à l’ETHZ ne peut pas être admis à l’EPFL.

Art. 6 - Organisation du programme

1 Les étudiantes et les étudiants restent soumis aux plan et règlement d’études en vigueur lors de leur entrée au Master.

2 Les enseignements sont dispensés par l'EPFL et l'ETHZ. Le premier semestre du cycle master se déroule à l'EPFL et un des autres semestres se déroule à l'ETHZ. Durant le semestre passé à l'ETHZ, l'étudiante ou l'étudiant doit obtenir entre 20 et 35 ECTS parmi une liste de cours ETHZ établie par la section d'Informatique. Les crédits acquis sont validés par la section d'Informatique.

3 L'étudiante ou l'étudiant n'ayant pas acquis au moins 20 crédits selon l'alinéa 2, devra se réinscrire à des cours ETHZ et valider les crédits manquants avant la fin de son cycle master.

4 L'étudiante ou l'étudiant ayant acquis son bachelor à l'EPFL peut se réinscrire au programme de master du domaine du bachelor, en cas d'échec dans l'acquisition des crédits à l'ETHZ.

Art. 7 - Organisation du cycle master

1 Les enseignements du cycle master sont répartis en deux groupes et un bloc :

- le groupe 1 « breadth requirement » est composé des cours de la liste du groupe 1 du plan d'études;
- le groupe 2 « depth requirement and options » est composé:
  - des cours de la liste du groupe 2 du plan d'études ;
  - des crédits surnuméraires acquis dans le groupe 1 ;
  - d'un projet de recherche optionnel de 8 crédits;
  - de cours hors plan d'études suivant l'alinéa 5.

2 Le bloc « Projets et SHS » est composé d'un projet de 12 crédits et de l'enseignement SHS (art. 8).

3 Le projet de recherche du bloc « Projets et SHS » et le projet de recherche optionnel du groupe 2 ne peuvent être effectués durant le même semestre.

4 Sur l'ensemble des cours pris à l'EPFL et à l'ETHZ, seuls un cours séminaire et un laboratoire peuvent être comptabilisés comme "depth requirement".

5 Les notes obtenues à l'ETHZ en dehors du semestre d'échange obligatoire, ou du minimum de 20 crédits à obtenir, sont reportées avec un "R" dans le relevé de notes de l'EPFL et ne sont pas prises en compte dans le calcul de la moyenne générale du Master.

6 Des cours comptant pour un maximum de 15 crédits au total peuvent être choisis en dehors de la liste des cours du plan d'études. Le choix de ces cours doit être accepté préalablement par la direction de la section qui peut augmenter le maximum de 15 crédits si la demande est justifiée.

Art. 8 - Enseignement SHS

1 L'enseignement SHS du semestre d'automne constitue l'introduction à la réalisation du projet SHS du semestre de printemps.

2 Pour autant qu'il considère que le cursus d'un cas individuel le justifie, le Collège des Humanités peut, d'entente avec l'équipe enseignante, déroger à cette organisation en autorisant que le projet soit réalisé au même semestre que le cours d'introduction ou soit réalisé à un semestre ultérieur.

Art. 9 - Examens du cycle master

1 Le groupe 1 « breadth requirement » est réussi lorsque **32 crédits** sont acquis.

2 La condition « depth requirement » est remplie lorsque **30 crédits** sont acquis dans la liste des cours avalisés « depth requirement ».

3 Le groupe « breadth requirement et depth requirement and options », composé du groupe 1 et du groupe 2, est réussi lorsque **72 crédits** sont acquis.

4 Le bloc « Projets et SHS » est réussi lorsque **18 crédits** sont acquis.

Art. 10 – Stage d'ingénierie

1 Un stage d'ingénierie doit être effectué dès la fin du 2e semestre et avant le projet de master. Sur demande, la section peut autoriser les titulaires d'un bachelor EPFL en Informatique ou Systèmes de communication, à réaliser le stage plus tôt.

2 Le stage prend l’une des formes suivantes:

- soit un stage d’été de minimum 8 semaines
- soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre)
- soit un projet de master de 25 semaines en entreprise (art. 11).

3 Effectuer des cours/projet en parallèle au stage n’est pas autorisé.

4 La personne responsable des stages de la section évalue le stage, par l’appréciation « réussi » ou « échoué ». Sa réussite est une condition pour l’admission au projet de master. En cas d’échec, il peut être répété une fois, en règle générale dans une autre entreprise.

5 Il est validé avec les 30 crédits du projet de master.

6 Les modalités d’organisation et les critères de validation du stage font l’objet d’instructions internes à la section.

Art. 11 – Projet de master

1 Le projet de master doit être réalisé dans le domaine de la Cybersécurité.

2 Il est placé sous la supervision d’une professeure ou d'un professeur ou MER, affilié à la section d’Informatique ou de Systèmes de communication.

3 Le projet de master s’étend sur une durée de 17 s’il est effectué à l’EPFL ou de 25 semaines s’il est effectué hors EPFL.

Au nom de l'EPFL

Le Vice-président académique, J. S. Hesthaven

Lausanne, le 20 juin 2024

Code	Matière	Enseignant·es sous réserve de modification	Orientation					Semestre						Crédits	Période des épreuves	Type examen
								AUT			PRI					
			A	B	C	D	E	c	e	p	c	e	p			
Branches de bases :														**28**		
CS-250	Algorithms I	Chiesa/Svensson									**4**	**2**		8	E	écrit
MATH-203(d)	Analysis III	Licht						**3**	**2**					6	H	écrit
CS-202	Computer systems	Argyraki/Bugnion/ Chappelier									**4**	**2**	**2**	8	E	écrit
MATH-232	Probability and statistics (for IC)	Abbé						**4**	**2**					6	H	écrit
Branches spécifiques (une orientation A, B, C, D ou E doit être choisie) :														**30 à 32**		
MATH-310	Algebra	Lachowska	A			D		**2**	**2**					4	H	écrit
CS-200	Computer architecture	Ienne	A		C			**4**	**2**	**2**				8	H	écrit
COM-301	Computer security and privacy	Bugnion/Troncoso	A		C			**2**	**2**					6	H	écrit
CS-300	Data-Intensive systems	Ailamaki/Kashyap	A	B	C						**2**	**1**	**2**	6	E	écrit
PHYS-114	Physique générale : électromagnétisme	Avino				D	E	**2**	**2**					4	H	écrit
CS-233	Introduction to machine learning	Fua/Salzmann		B			E				**2**	**2**	**2**	6	E	écrit
COM-300	Modèles stochastiques pour les communications	Thiran P.				D		**4**	**2**					6	H	écrit
BIOENG-310	Neuroscience foundations for engineers	Schrimpf/Zenk					E				**3**	**3**		6	E	écrit
COM-302	Principles of digital communications	Telatar				D	E				**4**	**1**	**1**	6	E	écrit
CS-290	Responsible software	Hardebolle		B	C	D		**2**	**2**					4	sem A	
COM-202	Signal processing	Prandoni/Shkel		B		D	E				**4**	**2**	**2**	8	E	écrit
CS-214	Software construction	Kuncak/Odersky/Pit-Claudel	A	B	C			**3**	**2**	**3**				8	H	écrit
	Totaux													**entre 57 et 60**		

Orientations :

A : Cybersecurité
B : Data science
C : Informatique
D : Systèmes de communication
E : Neuro-X

Le choix de l'orientation se fait en concertation avec le directeur de section ou son adjointe en tenant compte du Master visé

Légende :

c : cours e : exercices p : branches pratiques / : enseignement partagé
colonnes c/e/p : nb d'heures par semaine
1 semestre comprend 14 semaines.
type examination : voir règlement d'application

RÈGLEMENT D’APPLICATION DU CONTRÔLE DES ÉTUDES CONCERNANT LA PASSERELLE HES-EPFL

Année académique 2024-2025
du 20 juin 2024

La Vice-présidence académique,

vu l’art. 9 de l’ordonnance du Conseil des hautes écoles sur la coordination de l’enseignement du 29 novembre 2019,
vu l’art. 11 de l’ordonnance concernant l’admission à l’EPFL du 8 mai 1995,
vu l’art. 8 de la directive sur les programmes de master et les mineurs, du 17 octobre 2005,
vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004,
vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015,

arrête :

Art. 1 - Passerelle HES-EPFL

1 Le présent règlement fixe les règles spécifiques à l’admission à la formation de master de l’EPFL sur la base d’un titre de bachelor HES (passerelle HES-EPFL ; ci-après la passerelle), pour l’année académique 2024-2025.

Il fixe également les règles d’études de la passerelle.

2 Le titre de bachelor HES avec une moyenne minimale correspondant à la notation de 5.00 de l’EPFL permet l’admission au programme de master EPFL dans la discipline correspondante, avec la condition de réussir la passerelle.

3 La passerelle complète la formation HES par l’acquisition à l’EPFL de 57 à 60 crédits en sciences de base, en ingénierie ou en architecture, suivant le master suivi, suivant le plan d’études de passerelle correspondant à la discipline.

4 Les branches composant la passerelle sont divisées en un bloc de branches de base et un bloc de branches d’approfondissement. Chacun des deux blocs comprend entre 25 et 35 crédits.

5 Aux branches permettant d’acquérir les crédits de la passerelle, peuvent s’ajouter les éventuelles branches prérequises pour les branches du master suivi, conformément à la fiche de cours correspondante.

6 La réussite de la passerelle permet l’admission définitive au programme de master. Elle ne donne lieu à aucun titre.

Art. 2 - Inscription anticipée au master

Pour s’inscrire aux branches du programme de master, au moins 30 crédits doivent être acquis dans les branches de la passerelle (branches comptées individuellement).

Art. 3 - Conditions de réussite de la passerelle

1 La passerelle est réussie lorsque l’ensemble de ses crédits sont acquis dans un délai de deux ans au maximum. Ces crédits sont acquis par l’obtention d’une moyenne des branches égale ou supérieure à 4,00 pour chacun des deux blocs de la passerelle.

2 L’acquisition de moins de 30 crédits dans les branches de la passerelle (branches prises individuellement) au terme des examens de la première année entraîne un échec définitif.

Art. 4 - Règles applicables en deuxième année

1 La personne qui doit acquérir des crédits manquants sur une deuxième année demeure soumise au règlement et au plan d’études de passerelle qui se rapporte à sa première année (année d’admission à la passerelle).

2 La répétition d’une branche se déroule conformément aux règles de la branche pour l’année de la répétition.

Art. 5 - Période des cours et épreuves

1 Les cours de la passerelle débutent avec le semestre d'automne. L'entrée en cours d'année est exclue.

2 Conformément aux règles de l'EPFL,

- les branches de session sont examinées aux sessions d'examens d'hiver ou d'été correspondantes avec d'éventuelles épreuves de semestre (mention H ou E dans le plan d'études),
- les branches de semestre sont examinées pendant le semestre correspondant (mention sem A ou sem P dans le plan d'études).

Au nom de l'EPFL

Le Vice-président académique, J. S. Hesthaven

Lausanne, le 20 juin 2024

CS-500

AI product management

Kaboli Amin, Zamir Amir

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**7 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	3 weekly
Number of positions	

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course focuses on the development of real-word AI/ML products. It is intended for students who have acquired a theoretical background in AI/ML and are interested in applying that toward developing AI/ML-oriented products.

Content

AI is set to transform several industry sectors, and there is high demand for AI product managers. AI product management is a complex role that requires an understanding of both AI and product management. This course will enable students to identify opportunities for developing new AI products, understand when they should use AI in an existing product/process, manage the development of AI products, and launch AI products successfully. The lectures will introduce general product management to the students, and the guest lectures, by leading figures in AI industries, explain how the general product management skills are applied to the development of AI products.

Module 1: Introduction to AI product management

- Why is this needed?
- Product strategy
- Setting product objectives & identifying opportunities

Module 2: AI Product Discovery

- Understanding customers and problems
- Creating and testing hypothesis
- Defining product requirements
- Establishing product-market fit
- Setting AI product vision, strategy, roadmap

Module 3: AI Product development

- Mastering agile methodologies
- AI product design, tests, and development
- Managing team dynamics and effective communication with stakeholders

Module 4: AI Product Delivery

- Planning and executing a successful AI product launch
- Marketing the AI product
- Release and performance metrics for continuous improvement of AI products

Keywords

Artificial Intelligence (AI), AI product managers, Innovation

Learning Prerequisites

Required courses

CS-233 Introduction to machine learning or CS-433 Machinie learning or equivalent course on the basics of machine learning and deep learning

Important concepts to start the course

- Python programming
- Bascis of deep learning and machine learning
- Basics of probability and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- and understand opportunities for an AI product or using AI within an existing product
- the development of AI features
- Launch AI products successfully

Transversal skills

- Demonstrate the capacity for critical thinking
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Communicate effectively, being understood, including across different languages and cultures.
- Set objectives and design an action plan to reach those objectives.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Resolve conflicts in ways that are productive for the task and the people concerned.
- Make an oral presentation.
- Take account of the social and human dimensions of the engineering profession.

Teaching methods

- Formal lectures
- Group activities
- Class discussions
- Simulation games
- Hands-on exercises
- Project-based learning
- Real-world case studies

- Guest lectures by leading academic and industry figuers

Expected student activities

- **Individual** : Case evaluations, self-study, class discussions
- **In-group** : In-class exercises, projects, simulations games
- **Presentation** : Weekly presentations of assignments in coaching sessions

Assessment methods

Continuous evaluation of case reports, proejcts, individual and group presentations, class discussions, during the semester. More precisely :
25% Weekly in-class work and engagement
45% Class assignments, presentations, projects, and case reports
30% Final (final report and presentation and understanding of the case)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Cagan, M. (2017). *How to Create Tech Products Customers Love.* Wiley
- Kahneman, D., Sibony, O., & Sunstein, C. R. (2021). *Noise: A flaw in human judgment*. Little, Brown.
- Iansiti, M., & Lakhani, K. R. (2020). *Competing in the age of AI: strategy and leadership when algorithms and networks run the world*. Harvard Business Press.

Ressources en bibliothèque

- How to Create Tech Products Customers Love / Cagan
- Noise / Kahneman
- Competing in the age of AI / Iansiti

Moodle Link

- https://go.epfl.ch/CS-500

CS-420

Advanced compiler construction

Schinz Michel

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Students learn several implementation techniques for modern functional and object-oriented programming languages. They put some of them into practice by developing key parts of a compiler and run time system for a simple functional programming language.

Content

Part 1: implementation of high-level concepts

- functional languages: closures, continuations, tail call elimination,
- object-oriented languages: object layout, method dispatch, membership test.

Part 2: optimizations

- compiler intermediate representations (RTL, SSA, CPS),
- inlining and simple optimizations,
- register allocation.

Part 3: run time support

- interpreters and virtual machines,
- memory management (including garbage collection).

Keywords

compilation, programming languages, functional programming languages, object-oriented programming languages, code optimization, register allocation, garbage collection, virtual machines, interpreters, Scala.

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Excellent knowledge of Scala and C programming languages

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the quality of a compiler intermediate representation
- Design compilers and run time systems for object-oriented and functional programming languages
- Implement rewriting-based compiler optimizations
- Implement efficient virtual machines and interpreters
- Implement mark and sweep or copying garbage collectors

Teaching methods

Ex Cathedra, mini-project

Assessment methods

Continuous control (mini-project 80%, final exam 20%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

The Garbage Collection Handbook - The Art of Automatic Memory Management, second edition, Richard Jones, Antony Hosking, Eliot Moss (ISBN 9781032218038).

Ressources en bibliothèque

- The Garbage Collection Handbook / Jones

Websites

- https://cs420.epfl.ch/

CS-470

Advanced computer architecture

Ienne Paolo

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Labs	2 weekly
Number of positions	

Summary

The course studies techniques to exploit Instruction-Level Parallelism (ILP) statically and dynamically. It also addresses some aspects of the design of domain-specific accelerators. Finally, it explores security challenges based on microarchitectural features and hardware isolation techniques.

Content

Pushing processor performance to its limits:

- Principles of Instruction Level Parallelism (ILP)
- Register renaming techniques
- Prediction and speculation
- Simultaneous multithreading
- VLIW and compiler techniques for ILP
- Dynamic binary translation

Domain specific architectures and accelerators:

- Specificities of embedded vs. general computing processors
- Overview of DSPs and related compilation challenges
- High-Level Synthesis and accelerators

Hardware security:

- Information leakage through the microarchitecture
- Trusted Execution Environments
- Physical side-channel attacks

Keywords

Processors, Instruction Level Parallelism, Systems-on-Chip, Embedded Systems, High-Level Synthesis, Hardware Security.

Learning Prerequisites

Required courses

- CS-208 Computer Architecture I

Recommended courses

- CS-209 Computer Architecture II

Important concepts to start the course

Undergraduate knowledge of digital circuit design and of computer architecture

Learning Outcomes

By the end of the course, the student must be able to:

- Design strategies to exploit instruction level parallelism in processors.
- Contrast static and dynamic techniques for instruction level parallelism.
- Design effective processor (micro-)architectures for which efficient compilers can be written.
- Develop hardware accelerators competitive to best commercial processors
- Defend against security threats based on microarchitectural processor features

Teaching methods

Courses, labs, and compulsory homeworks.

Assessment methods

Homeworks (30%)
Final exam (70%)

Supervision

Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufman, 6th edition, 2017.

Ressources en bibliothèque

- Computer Architecture / Hennessy

Moodle Link

- https://go.epfl.ch/CS-470

Prerequisite for

CS-471 Advanced Multiprocessor Architecture

CS-440

Advanced computer graphics

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	1 weekly
Project	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

This course covers advanced 3D graphics techniques for realistic image synthesis. Students will learn how light interacts with objects in our world, and how to recreate these phenomena in a computer simulation to create synthetic images that are indistinguishable from photographs.

Content

This is a project-based course: students will initially receive a basic software package that lacks most rendering-related functionality.
Over the course of the semester, we will discuss a variety of concepts and tools including the basic physical quantities, how light interacts with surfaces, and how to solve the resulting mathematical problem numerically to create realistic images. Advanced topics include participating media, material models for sub-surface light transport, and Markov Chain Monte Carlo Methods.
Each major topic is accompanied by an assignment so that students can implement solution algorithms and obtain practical experience with these techniques within their own software framework.
Towards the end of the course, students will realize a self-directed final project that extends their rendering software with additional features of their own choosing. The objective of the final project is to create a single image of both technical and artistic merit that is entered into a rendering competition and judged by an independent panel of computer graphics experts.

Learning Prerequisites

Required courses

Nothing

Recommended courses

Introduction to Computer Graphics

Important concepts to start the course

We will rely on calculus, linear algebra and use basic concepts of algorithms and data structures. Students are expected to be familiar with the C++ programming language that is used in the programming assignments.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize and understand the physical quantities of light transport and be able to perform basic computations using pencil+paper
- Explain a range of surface and subsurface material models
- Explain the rendering and radiative transfer equation and show how to construct Monte Carlo methods to solve them
- Design and implement an advanced rendering system based on Monte Carlo integration
- Assess / Evaluate the performance and conceptual limits of the implemented simulation code

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project.

Assessment methods

Intermediate assignments (40%), final project (60%)

Resources

Bibliography

A list of books will be provided at the beginning of the class

Notes/Handbook

Slides and online resources will be provided in class

Websites

- https://rgl.epfl.ch/courses/ACG22

Moodle Link

- https://go.epfl.ch/CS-440

EPFL

COM-501

Advanced cryptography

Vaudenay Serge

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course reviews some failure cases in public-key cryptography. It introduces some cryptanalysis techniques. It also presents fundamentals in cryptography such as interactive proofs. Finally, it presents some techniques to validate the security of cryptographic primitives.

Content

1. **The cryptograhic zoo**: definitions, cryptographic primitives, math, algorithms, complexity
2. **Cryptographic security models:** security notions for encryption and authentication, game reduction techniques, RSA and Diffie-Hellman security notions
3. **Public-key cryptanalysis:** side channels, low RSA exponents, discrete logarithm, ElGamal signature
4. **Interactive proofs:** NP-completeness, interactive systems, zero-knowledge
5. **Symmetric-key cryptanalysis:** differential and linear cryptanalysis, hypothesis testing, decorrelation
6. **Proof techniques:** random oracles, leftover-hash lemma, Fujisaki-Okamoto transform

Keywords

cryptography, cryptanalysis, interactive proof, security proof

Learning Prerequisites

Required courses

- Cryptography and security (COM-401)

Important concepts to start the course

- Cryptography
- Mathematical reasoning
- Number theory and probability theory
- Algorithmics
- Complexity

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the security deployed by cryptographic schemes

- Prove or disprove security
- Justify the elements of cryptographic schemes
- Analyze cryptographic schemes
- Implement attack methods
- Model security notions

Teaching methods

ex-cathedra

Expected student activities

- active participation during the course
- take notes during the course
- do the exercises during the exercise sessions
- complete the regular tests and homework
- read the material from the course
- self-train using the provided material
- do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

- homework (30%)
- regular graded tests (30%)
- midterm exam (40%)

Final exam averaged (same weight) with the contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

- Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
- A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.
- Algorithmic cryptanalysis. Antoine Joux. CRC 2009.

Ressources en bibliothèque

- Algorithmic cryptanalysis / Joux
- Communication security / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-501

Videos

- https://mediaspace.epfl.ch/channel/COM-501+Advanced+Cryptography

EPFL

CS-471 Advanced multiprocessor architecture

Falsafi Babak

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**12 weekly**
Lecture	4 weekly
Project	8 weekly
Number of positions	

Summary

Multiprocessors are basic building blocks for all computer systems. This course covers the architecture and organization of modern multiprocessors, prevalent accelerators (e.g., GPU, TPU), and datacenters. It includes a research project on multiprocessors and post-Moore era datacenters.

Content

- Introduction
- Metrics and methodologies
- Parallel programming models
- Communication models
- Applications and Workloads
- Cache hierarchies & memory models
- Memory & storage hierarchies
- Interconnects
- GPUs
- AI/ML/Analytic accelerators
- Near-memory computing
- Datacenters & cloud Computing
- Cloud-native CPU
- Cloud-native memory hierarchies
- Sustainable architecture

Learning Prerequisites

Recommended courses

- Advanced computer architecture
- Systems for data management and data science

Learning Outcomes

- Explore the development trend of computation systems and datacenters
- Establish the basic model to analyze the performance and characteristics of foundational workloads operating in cloud environments.
- Classify and describe the components of modern parallel systems, including multiple processors, cache hierarchies, memory systems, interconnects, and accelerators, and their roles in handling emerging workloads
- Define and clarify research questions and opportunities
- Interpret and critique research papers and extract insights for research questions
- Plan and conduct a research project
- Present research contributions

Teaching methods

Lecture, research paper retrieval, and a research project

Assessment methods

- Homework: 10%
- Research project (in group): 40%
- Midterm exam: 20%
- Final exam: 30%

Resources

Websites

- https://parsa.epfl.ch/course-info/cs471/

EPFL

CS-477

Advanced operating systems

Kashyap Sanidhya

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

This course teaches advanced OS system design. Using an extensive hands-on approach, the course focuses on traditional and new concepts proposed in the past two decades in the area of operating systems.

Content

- Concurrent execution
- Memory management using things like virtual memory and memory allocations
- Scalability, such as lock-free data structures
- File systems
- Operating system architecture
- Virtualization
- Security such as data security, integrity, and authentification

Learning Prerequisites

Required courses

CS-200 Computer architecture
CS-202 Computer systems (or COM-208 Computer networks / CS-323 Introduction to operating systems)
CS-214 Software construction (or CS-210 Functional programming)
CS-300 Data-Intensive Systems (or CS-322 Introduction to database systems)

Important concepts to start the course

This course is a hands-on course that required solid background in operating systems, databases, programming language, and computer architecture.

Learning Outcomes

By the end of the course, the student must be able to:

- Identify and manage key components of operating systems
- Choose or critique design choices for OS
- Create and design an OS with practical choices
- Report performance and possible optimizations for applications

Teaching methods

- In-class lectures
- Homework assignments
- Weekly preparatory exercises
- Programming exercise

Expected student activities

- Attend lectures and tutorial sessions
- Complete the homework assignments
- Participate actively in the course
- Complete programming exercises

Assessment methods

- Preparatory exercise 10%
- Labs 40%
- Final exam 50%

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-477

EPFL

COM-417

Advanced probability and applications

Shkel Yanina

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Obl.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Electrical Engineering		Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course, various aspects of probability theory are considered. The first part is devoted to the main theorems in the field (law of large numbers, central limit theorem, concentration inequalities), while the second part focuses on the theory of martingales in discrete time.

Content

- sigma-fields, random variables
- probability measures, distributions
- independence, convolution
- expectation, characteristic function
- random vectors and Gaussian random vectors
- inequalities, convergences of sequences of random variables
- laws of large numbers, applications and extensions
- convergence in distribution, central limit theorem and applications
- moments and Carleman's theorem
- concentration inequalities
- conditional expectation
- martingales, stopping times
- martingale convergence theorems

Keywords

probability theory, measure theory, martingales, convergence theorems

Learning Prerequisites

Required courses

Basic probability course
Calculus courses

Recommended courses

Complex analysis

Important concepts to start the course

This course is NOT an introductory course on probability: the students should have a good understanding and practice of basic probability concepts such as: distribution, expectation, variance, independence, conditional probability.
The students should also be at ease with calculus. Complex analyisis is a plus, but is not required.

On the other hand, no prior background on measure theory is needed for this course: we will go through the basic concepts one by one at the beginning.

Learning Outcomes

By the end of the course, the student must be able to:

- understand the main ideas at the heart of probability theory

Teaching methods

Ex cathedra and flipped lectures + exercise sessions

Expected student activities

active participation to exercise sessions

Assessment methods

graded homeworks 20%
midterm 20%
final exam 60%

Resources

Bibliography

Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability,1st edition, www.ProbabilityBookstore.com, 2007.
Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory,2nd edition, World Scientific, 2006.
Geoffrey R. Grimmett, David R. Stirzaker, Probability and Random Processes,3rd edition, Oxford University Press, 2001.
Richard Durrett, Probability: Theory and Examples, 4th edition, Cambridge University Press, 2010.
Patrick Billingsley, Probability and Measure, 3rd edition, Wiley, 1995.

Ressources en bibliothèque

- Probability and Random Processes
- Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability, 1st ed
- Patrick Billingsley, Probability and Measure, 3rd ed
- Richard Durrett, Probability: Theory and Examples, 4th ed
- Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory, 2nd ed

Notes/Handbook

available on the course website

Websites

- https://moodle.epfl.ch/course/view.php?id=14557

Moodle Link

- https://go.epfl.ch/COM-417

Prerequisite for

Advanced classes requiring a good knowledge of probability

EPFL

CS-523

Advanced topics on privacy enhancing technologies

Troncoso Carmela

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This advanced course will provide students with the knowledge to tackle the design of privacy-preserving ICT systems. Students will learn about existing technologies to prect privacy, and how to evaluate the protection they provide.

Content

The course will cover the following topics :
- Privacy definitions and concepts
- Privacy-preserving cryptographics solutions : anonymous credentials, zero-knowledge proofs, secure multi-party computation, homomorphic encryption, Private information retrieval (PIR), Oblivious RAM (ORAM)
- Anonymization and data hiding : generalization, differential privacy, etc
- Machine learning and privacy
- Protection of metadata : anonymous communications systems, location privacy, censorpship resistance
- Online tracking and countermeasures
- Privacy engineering : design and evaluation (evaluation metrics and notions)
- Legal aspects of privacy

Keywords

Privacy, anonymity, homomorphic encryption, secure multi-paty computation, anonymous credentials, ethics

Learning Prerequisites

Required courses

COM-301 Computer security
COM-402 Information security and privacy

Recommended courses

COM-401 Cryptography and security

Important concepts to start the course

Basic programming skills; basics of probabilities and statistics; basics of cryptography

Learning Outcomes

By the end of the course, the student must be able to:

- Select appropriately appropriately privacy mechanisms

- Develop privacy technologies
- Assess / Evaluate privacy protection
- Reason about privacy concerns

Teaching methods

Lectures and written exercises to deepen understanding of concepts
Programming-oriented assignments to practice use of privacy technologies

Expected student activities

Participation in the lectures. Active participation is encouraged.
Participation in exercise session and complete the exercises regularly
Completion of programming assignments

Supervision

Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-523

EPFL

CS-450 **Algorithms II**

Kapralov Michael, Svensson Ola Nils Anders

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**7 weekly**
Lecture	4 weekly
Exercises	3 weekly
Number of positions	

Summary

A first graduate course in algorithms, this course assumes minimal background, but moves rapidly. The objective is to learn the main techniques of algorithm analysis and design, while building a repertory of basic algorithmic solutions to problems in many domains.

Content

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance.

Learning Prerequisites

Required courses

An undergraduate course in Discrete Structures / Discrete Mathematics, covering formal notation (sets, propositional logic, quantifiers), proof methods (derivation, contradiction, induction), enumeration of choices and other basic combinatorial techniques, graphs and simple results on graphs (cycles, paths, spanning trees, cliques, coloring, etc.).

Recommended courses

An undergraduate course in Data Structures and Algorithms.
An undergraduate course in Probability and Statistics.

Important concepts to start the course

Basic data structures (arrays, lists, stacks, queues,trees) and algorithms (binary search; sorting; graph connectivity); basic discrete mathematics (proof methods, induction, enumeration and counting, graphs); elementary probability and statistics (random variables, distributions, independence, conditional probabilities); data abstraction.

Learning Outcomes

By the end of the course, the student must be able to:

- Use a suitable analysis method for any given algorithm
- Prove correctness and running-time bounds
- Design new algorithms for variations of problems studied in class
- Select appropriately an algorithmic paradigm for the problem at hand

- Define formally an algorithmic problem

Teaching methods

Ex cathedra lecture, reading

Assessment methods

- midterm (30%)
- homework (30%)
- final exam (40%)

Supervision

Forum	Yes
Others	For details, see the course web page

Resources

Websites

- http://theory.epfl.ch/courses/AdvAlg/

Moodle Link

- https://go.epfl.ch/CS-450

EPFL

EE-512

Applied biomedical signal processing

Lemay Mathieu

Cursus	Sem.	Type
Biomedical technologies minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The goal of this course is twofold: (1) to introduce physiological basis, signal acquisition solutions (sensors) and state-of-the-art signal processing techniques, and (2) to propose concrete examples of applications for vital sign monitoring and diagnosis purposes.

Content

• Introduction on the basics in anatomy and physiology of autonomous nervous system, electrical cardiac system, hemodynamic basis, brain and respiratory activities as well as location.

• Digital signal processing basics including sampling, Fourier transform, filtering, stochastic signal correlation and power spectral density. Time-frequency analysis including short-term Fourier and wavelet transforms. Linear modelling including autoregressive models, linear prediction, parametric spectral estimation, and criteria for model selection. Adaptive filtering including adaptive prediction and estimations of transfer functions as well as adaptive interference cancellation.

• Digital signal processing miscellaneous techniques including polynomial models, singular value decomposition and principal component analysis, phase-rectified signal averaging, source separation, support vector regression, and neural network structures such as CNN and RNN.

• Applications and exercises related to cardiac arrythmia detection and classification, central blood pressure estimation, sleep phase classification, heart rate tracking robust against motion artefacts, epilepsy event detection, fall detection, apnoea detection, SpO2 estimation, and respiration tracking and volume estimation. These exercises will be based on biomedical signals such as bio-impedance, electrocardiogram, electroencephalogram, hypnogram, movement (accelerometer, gyroscope, and barometer), photoplethysmography, vocal/audio.

Keywords

signal processing, biomedical engineering, signal modelling, spectral analysis, adaptive filtering, algorithm design

Learning Prerequisites

Recommended courses

Signal processing for telecommunications COM-303
Signal processing EE-350

Important concepts to start the course

basics of discrete-time signal analysis
basics in signal processing programming

Teaching methods

Ex cathedra lectures (approx.. 2h per module) and practical work using Matlab/Python (approx.. 2h per module). The student should provide a separate report for each of the practical work session for evaluation. Grades are based on the practicals and a final exam.

Expected student activities

- Attending lectures
- Processing and analysing human data
- Testing signal processing techniques

Assessment methods

1.75 points in total for the lab/exercise sessions reports during the semester (35% of the final total grade)
3.25 points for the final exam during the examination period (65% of the final total grade)

Supervision

Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/EE-512

MATH-493

Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Ing.-math	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course covers topics in applied biostatistics, with an emphasis on practical aspects of data analysis using R statistical software. Topics include types of studies and their design and analysis, high dimensional data analysis (genetic/genomic) and other topics as time and interest permit.

Content

Types of studies
Design and analysis of studies
R statistical software
Reproducible research techniques and tools
Report writing
Exploratory data analysis
Liniear modeling (regression, anova)
Generalized linear modeling (logistic, Poission)
Survival analysis
Discrete data analysis
Meta-analysis
High dimensional data analysis (genetics/genomics applications)
Additional topics as time and interest permit

Keywords

Data analysis, reproducible research, statistical methods, R, biostatistical data analysis, statistical data analysis

Learning Prerequisites

Required courses

This course will be very difficult for students with no previous course or experience with statistics. Previous experience with R is neither assumed nor required.

Recommended courses

Undergraduate statistics course

Important concepts to start the course

It is useful to review statistical hypothesis testing.

Learning Outcomes

By the end of the course, the student must be able to:
- Interpret analysis results
- Justify analysis plan
- Plan analysis for a given dataset
- Analyze various types of biostatistical data
- Synthesize analysis into a written report
- Report plan of analysis and results obtained
- Synthesize analysis into a written report
- Report plan of analysis and results obtained
- Justify analysis plan
- Plan analysis for a given dataset
- Interpret analysis results
- Analyze various types of biostatistical data

Transversal skills

- Write a scientific or technical report.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Take feedback (critique) and respond in an appropriate manner.
- Use a work methodology appropriate to the task.

Teaching methods

Lectures and practical exercises using R. Typically, each week covers an analysis method in the lecture and then the corresponding exercise session consists of an R practical showing how to implement the methods using R. In each practical, students use R to carry out analyses of the relevant data type for that week.

Expected student activities

Students are expected to participate in their learning by attending lectures and practical exercise sessions, posing questions, proposing topics of interest, peer reviewing of preliminary reports, and interacting with teaching staff regarding their understanding of course material. In addition, there will be a number of short activities in class aimed at improving English for report writing.

Assessment methods

Evaluation is based on written reports of projects analyzing biostatistical data.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

To be provided during the course.
Pre-recorded lectures (videos) will also be provided.

Moodle Link

- https://go.epfl.ch/MATH-493

CS-401

Applied data analysis

Brbic Maria

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data and Internet of Things minor	H	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Obl.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Statistics	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course teaches the basic techniques, methodologies, and practical skills required to draw meaningful insights from a variety of data, with the help of the most acclaimed software tools in the data science world (pandas, scikit-learn, Spark, etc.)

Content

Thanks to modern software tools that allow to easily process and analyze data at scale, we are now able to extract invaluable insights from the vast amount of data generated daily. As a result, both the business and scientific world are undergoing a revolution which is fueled by one of the most sought after job profiles: the data scientist.
This course covers the fundamental steps of the data science pipeline:
Data wrangling

- Data acqusition (scraping, crawling, parsing, etc.)
- Data manipulation, array programming, dataframes

- The many sources of data problems (and how to fix them): missing data, incorrect data, inconsistent representations
- Data quality testing with crowdsourcing

Data interpretation

- Statistics in practice (distribution fitting, statistical significance, etc.)
- Working with "found data" (design of observational studies, regression analysis)
- Machine learning in practice (supervised and unsupervised, feature engineering, evaluation, etc.)
- Text mining: preprocessing steps, vector space model, topic models
- Social network analysis (properties of real networks, working graph data, etc.)

Data visualization

- Introduction to different plot types (1, 2, and 3 variables), layout best practices, network and geographical data
- Visualization to diagnose data problems, scaling visualization to large datasets, visualizing uncertain data

Reporting

- Results reporting, infographics
- How to publish reproducible results

The students will learn the techniques during the ex-cathedra lectures and will be introduced, in the lab sessions, to the software tools required to complete the homework assignments.
In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. In the project, students propose and execute meaningful analyses of a real-world dataset, which will require creativity and the application of the tools encountered in the course. The outcome of this team effort will be a project portfolio that will be made public (and available as open source).
At the end of the semester, students will take a 3-hour final exam in a classroom with their own computer, where they will be asked to complete a data analysis pipeline (both with code and extensive comments) on a dataset they have never worked with before.

Keywords

data science, data analysis, data mining, machine learning

Learning Prerequisites

Required courses

The student must have passed an introduction to databases course, OR a course in probability & statistics, OR two separate courses that include programming projects. Programming skills are required (in class we will use mostly Python).

Recommended courses

- CS-423 Distributed Information Systems
- CS-433 Machine Learning

Important concepts to start the course

programming, algorithms, probability and statistics, databases

Learning Outcomes

By the end of the course, the student must be able to:

- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the Data Science pipeline
- Perform data acquisition (data formats, dataset fusion, Web scrapers, REST APIs, open data, big data platforms, etc.)
- Perform data wrangling (fixing missing and incorrect data, data reconciliation, data quality assessments, etc.)
- Perform data interpretation (statistics, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline
- Perform data interpretation (statistics, correlation vs. causality, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline

Transversal skills

- Give feedback (critique) in an appropriate fashion.
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

- Physical in-class recitations and lab sessions
- Homework assignments
- Course project

Expected student activities

Students are expected to:

- Attend the lectures and lab sessions
- Complete 2-3 homework assignments
- Conduct the class project
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- Homework
- Project
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

- http://ada.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-401

EE-554

Automatic speech processing

Magimai Doss Mathew

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The goal of this course is to provide the students with the main formalisms, models and algorithms required for the implementation of advanced speech processing applications (involving, among others, speech coding, speech analysis/synthesis, and speech recognition, speaker recognition).

Content

1. Introduction: Speech processing tasks, Speech science, language engineering applications.

2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming, speech representation learning

3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.

4. Speech Synthesis: speech synthesis models, concatenative synthesis, hidden Markov models (HMMs) based speech synthesis, Neural speech synthesis.

5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on HMMs, Neural networks-based speech recognition.

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, Text-dependent and Text-independent speaker verification, Gaussian mixture models-/HMM-based speaker verification, speaker embeddings based speaker verification, presentation attack detection (antispoofing).

7. Paralinguistic speech processing: fundamentals and applications (e.g., emotion recognition, pathological speech detection, depression detection), hand-crafted feature based approaches, neural approaches.

Keywords

speech processing, speech coding, speech analysis/synthesis, automatic speech recognition, speaker identification, text-to-speech

Learning Prerequisites

Required courses

Basis in linear algebra, signal processing (FFT), and statistics.

Important concepts to start the course

Basic knowledge in signal processing, linear algebra, statistics and stochastic processes. Basic knowledge in machine learning/statistical pattern recognition is not a must but would be helpful in .

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze speech signal properties
- Exploit those properties for speech coding, speech synthesis, speech recognition, speaker recognition and paralinguistic speech processing
- Formulate speech processing problems
- Choose appropriate methods for target speech processing tasks

Transversal skills

- Use a work methodology appropriate to the task.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Teaching methods

Lecture + lab exercises

Expected student activities

Attending courses and lab exercises. Read additional papers and continue lab exercises at home if necessary. Regulary answer list of questions for feedback.

Assessment methods

Written exam without notes

Resources

Bibliography

Fundamentals of Speech Recognition / Rabiner and Juang
Speech and Language Processing / Dan Jurafsky and Daniel Martin (2nd Edition)
Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Ressources en bibliothèque

- Speech and Language Processing / Dan Jurafsky and Daniel Martin
- Fundamentals of Speech Recognition / Rabiner and Juang
- Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
- Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
- Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Moodle Link

- https://go.epfl.ch/EE-554

EPFL

MICRO-452 **Basics of mobile robotics**

Mondada Francesco

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	1 weekly
Practical work	1 weekly
Project	2 weekly
Number of positions	

Summary

The course teaches the basics of autonomous mobile robots. Both hardware (energy, locomotion, sensors) and software (signal processing, control, localization, trajectory planning, high-level control) will be tackled. The students will apply the knowledge to program and control a real mobile robot.

Content

- Sensors
- Perception, feature extraction
- Modeling
- Markov localization: Bayesian filter, Monte Carlo localization, extended Kalman filter
- Navigation: path planning, obstacle avoidance
- Control architectures and robotic frameworks
- Locomotion principles and control
- Sustainability

Keywords

mobile robots, sensing, perception, localisation, navigation, locomotion.

Learning Prerequisites

Required courses

Introduction to automatic control (catching up possible with extra effort)
Introduction to signal processing

Recommended courses

Microinformatique (SMT)

Important concepts to start the course

Embedded system programming
Basics of automatic control

Basics of signal processing

Learning Outcomes

By the end of the course, the student must be able to:

- Choose the right methods to design and control a mobile robot for a particular task.
- Integrate approriate methods for sensing, cognition and actuation
- Justify design choices for a robotic system
- Implement perception, localisation/navigation and control methods on a mobile robot
- Choose the right methods to design and control a mobile robot for a particular task.

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Set objectives and design an action plan to reach those objectives.
- Use a work methodology appropriate to the task.
- Assess progress against the plan, and adapt the plan as appropriate.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra, case studies, exercises, work on mobile robots, group project

Expected student activities

- weekly lectures
- studying provided additional materials
- attend case study discussions
- lab exercises with practical components
- project at the end of the semester

Assessment methods

Project during the semester (60% of the grade). The project takes place during the semester and the report and presentation are done before the end of the semester, following the specific planning given by the teacher at the beginning of the semester.
Written exam (40% of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004

Autonomous Robots: From Biological Inspiration to Implementation and Control G.A. Bekey, MIT Press, 2005
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005
Handbook of Robotics (chapter 35) B. Sicilian, and O. Khatib (Eds.), Springer, 2008
Elements of Robotics M. ben-Ari and F. Mondada, Spinger, 2017.
additional literature provided on Moodle

Ressources en bibliothèque

- Handbook of Robotics / Sicilian
- Elements of Robotics / Ben-Ari
- Probabilistic Robotics / Thrun
- Introduction to Autonomous Mobile Robots / Siegwart
- Autonomous Robots / Bekey

Notes/Handbook

Lecture slides are continously provided on Moodle during the course.
Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005

Moodle Link

- https://go.epfl.ch/MICRO-452

EPFL

MGT-416

Causal inference

Kiyavash Negar

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Management, Technology and Entrepreneurship minor	E	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Students will learn the core concepts and techniques of network analysis with emphasis on causal inference. Theory and application will be balanced, with students working directly with network data throughout the course.

Content

- Introduction: What is causal inference?
- Review of Useful Probability concepts
- Random variable, predictors, divergences
- Introduction to Applications
- Computational neuroscience
- Financial markets
- Social networks
- Pearl Causality
- Causal Bayesian Networks (CBNs)
- Learning CBNS: Faithfulness and identifiably
- Algorithms
- Potential Outcome Model
- Counterfactuals and identification problems
- Graphical causal models
- Randomized Experiments
- Identification of causes in randomized experiments
- Effect modification
- Causality in Times Series
- Granger causality
- More general linear predictors
- Beyond linear models and Granger causality
- Directed information graphs
- Efficient algorithms
- Concrete Applications
- Computational neuroscience
- Financial markets
- Social networks

Keywords

Causality, structure learning, network inference

Learning Prerequisites

Required courses

This course attempts to be as self contained as possible, but it does approach the topic from a quantitative point of view and, as such, students should be comfortable with the basics of (i.e. have taken at least one course in) the following topics before enrolling:
- Statistics
- Probability Theory
- Linear Algebra
- Calculus (integral and differential)
- Programing in Pythor and Matlab
As course work will be largely computational, experience with at least one programming language is also required.

Recommended courses

Knowledge of probability and calculus as well as programming is a must.

Learning Outcomes

By the end of the course, the student must be able to:

- Identify situations in which a problem/data can be thought of as a network.
- Analyze data appropriately using a variety of network causal inference techniques
- Interpret the results of applying causal discovery or inference algorithms

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

In class with supporing problem solving sessions.

Expected student activities

TA problem solving sessions, homework, exams, projects

Assessment methods

Regular individual assignments: 30%
Midterm project: 30%
Final project: 40%

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

course notes

Moodle Link

- https://go.epfl.ch/MGT-416

MATH-352

Causal thinking

Stensrud Mats Julius

Cursus	Sem.	Type
Chemistry	BA5	Opt.
Computational and Quantitative Biology		Obl.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course will give a unified presentation of modern methods for causal inference. We focus on concepts, and we will present examples and ideas from various scientific disciplines, including medicine, computer science, engineering, economics and epidemiology.

Content

Association vs. causation
Definitions of causal effects
- Causal models
- Counterfactuals and potential outcomes
- Individual level causal effects vs. average causal effects
- Population causal effects
Study design
- Randomisation and experiments
- Observational studies
Causal graphs
- Causal Directed Acyclic Graphs
- Single World Intervention Graphs
Identification of causal effects
- Identifiability assumptions
- SWIGs
Causal mechanisms
- Mediation and path specific effects
- Instrumental variables
Applications
- Medical interventions, including pharmaceuticals
- Experiments in technology industry and engineering
- Experiments in life sciences
- Causal effects and mechanisms in the social sciences.
Estimation of causal effects
- Estimation using classical statistical models
- Estimation using machine learning

Keywords

Causality; Causal inference; Randomisation; Design of experiments; Observational studies; Causal Graphs

Learning Prerequisites

Required courses

The course is intended for students from a range of different disciplines, including computer science, engineering, life science and physics. The students are expected to know the basics of statistical theory and probability theory (such as the second year courses in probability and statistics for engineers).

Recommended courses

Courses in statistical inference.

Important concepts to start the course

Familiarity with basic concepts in probability and statistics.

Learning Outcomes

By the end of the course, the student must be able to:

- Design experiments that can answer causal questions.
- Describe the fundamental theory of causal models.
- Critique assess causal assumptions and axioms.
- Distinguish between interpretation, identification and estimation.
- Describe when and how causal effects can be identified and estimated from non- experimental data.
- Estimate causal parameters from observational data

Teaching methods

Classroom lectures, where I will use a digital blackboard and slides.

Assessment methods

Final written exam. 1-2 graded homeworks.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Hernan, M.A. and Robins, J.M., 2020. Causal inference: What if?
Imbens, G.W. and Rubin, D.B., 2015. Causal inference in statistics, social, and biomedical sciences. Cambridge University Press.
Pearl, J., 2009. Causality. Cambridge university press.

Ressources en bibliothèque

- Causal inference in statistics, social, and biomedical sciences / Imbens
- Causality / Pearl
- Causal Inference / Hernan

Moodle Link

- https://go.epfl.ch/MATH-352

EPFL

BIO-105 **Cellular biology and biochemistry for engineers**

Zufferey Romain

Cursus	Sem.	Type
Biomedical technologies minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Minor in life sciences engineering	H	Opt.
Physics of living systems minor	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Basic course in biochemistry as well as cellular and molecular biology for non-life science students enrolling at the Master or PhD thesis level from various engineering disciplines. It reviews essential notions necessary for a training in biology-related engineering fields.

Content

The course gives basic knowledge on various phenoma taking place within a cell, and among cells within tissues and organs. The course gives an integrated view of various molecular mechanisms (rather in the second half of the class). It should therefore allow engineering students involved in future projects touching on biomedical problems to better integrate the constraints of a biological system and to enable them to communicate with specialists in both fields.
This course is not available to students who had already taken basic cell biology or biochemistry classes during their Bachelor studies at EPFL or elsewhere. This applies for example to the course BIO-109 "Introduction to Life Sciences for Information Sciences" and MSE 212 "Biology for engineers"

Keywords

The course contains chapters on the following subjects:
1.Cells and Organs
2.Chemical components of cells
3.Proteins, Enzymes
4.Energy, Metabolism
5.DNA, Chromosomes, Replication
6.Gene expression
7.Recombinant techniques
8.Membrane and Transport
9.Intracellular trafficking
10.Cytoskeleton
11.Cell division, Mitosis
12.Genetics, Meiosis
13.Cell communication, Signaling
14.Tissue, Tissue regeneration

Learning Prerequisites

Required courses

Bachelor degree in engineering or other non-life science discipline

Recommended courses

Some basic knowledge in chemistry can help, but not required

Important concepts to start the course

Curiosity about how biological systems work, willingness to acquire a certain amount of facts and details necessary to understand and discuss the various molecular mechanisms present in cells or related to modern biology

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the basic components and functions found in cells
- Draw schemes explaining essential cellular phenomena
- Explain which are the important metabolic pathways
- Translate information from genetic code
- Verify statements about specific cellular mechanisms
- Integrate knowledge from different cellular mechanisms

Transversal skills

- Access and evaluate appropriate sources of information.

Teaching methods

2 hours of ex cathedra-type of lecture
2 hours of exercises: the instructor gives out appr. 10 questions out (through Moodle and in the beginning of the session). The questions have different formats, and can in some cases just retrieve the acquired facts, in others have a more integrative problem-based learning approach.

Expected student activities

- review regularly the presented lectures.
- participate actively in the exercise sessions when the questions and problems are discussed altogether

Assessment methods

- a written exam at the winter exam session

Supervision

Office hours	Yes
Assistants	Yes
Forum	No
Others	- the teacher can always be reached through Email or phone to fix a one-to-one discussion about specific subjects

Resources

Bibliography

The lecture is aligned to selected chapters in the following book (recommended although not required):
"Essential Cell Bioogy" by B Alberts et al. , 3rd edition, Garland Science Taylor & Francis Group

Notes/Handbook

- Essential Cell Biology / Alberts

Moodle Link

- https://go.epfl.ch/BIO-105

EPFL

CS-524 **Computational complexity**

Göös Mika

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

In computational complexity we study the computational resources needed to solve problems and understand the relation between different types of computation. This course advances the students knowledge of computational complexity, and develop an understanding of fundamental open questions.

Content

- Complexity classes (time, space, nondeterminism)
- Space complexity (Logspace, L vs NL)
- Boolean circuits and nonuniform computation
- Power of randomness
- Lower bounds for concrete models of computation: Decision trees, communication protocols, propositional proofs.

Keywords

theoretical computer science
computational complexity

Learning Prerequisites

Recommended courses

Theory of computation (CS-251)
Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Demonstrate an understanding of computational complexity and the P vs NP problem
- Formalize and analyze abstractions of complex scenarios/problems
- Express a good understanding of different concepts of proofs
- Prove statements that are similar to those taught in the course
- Use and understand the role of randomness in computation

- Illustrate a basic understanding of probabilistically checkable proofs and their characterization of the class NP (the PCP-Theorem)
- Explain recent exciting developments in theoretical computer science
- Compare different models of computation

Transversal skills

- Demonstrate the capacity for critical thinking
- Summarize an article or a technical report.

Teaching methods

Lecturing and exercises

Expected student activities

Actively attending lectures and exercise sessions. Also homeworks and exam.

Assessment methods

Three homeworks and final exam

Resources

Bibliography

Sanjeev Arora and Boaz Barak: *Computational Complexity: A Modern Approach, Cambridge University Press.*
Stasys Jukna: *Boolean Function Complexity, Springer*

Ressources en bibliothèque

- Computational Complexity: A Modern Approach / Arora
- Boolean Function Complexity / Jukna

Moodle Link

- https://go.epfl.ch/CS-524

NX-465

Computational neurosciences: neuronal dynamics

Gerstner Wulfram

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
Biomedical technologies minor	E	Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Neuroscience		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course we study mathematical models of neurons and neuronal networks in the context of biology and establish links to models of cognition. The focus is on brain dynamics approximated by deterministic or stochastic differential equations.

Content

I. Models of single neurons
1. Introduction: brain, computers, and a first simple neuron model
2. Models on the level of ion current (Hodgkin-Huxley model)
3./4. Two-dimensional models and phase space analysis
II. Neuronal Dynamics of Cognition
5. Associative Memory and Attractor Dynamics (Hopfield Model)
6. Neuronal Populations and mean-field methods
7. Continuum models and perception
8. Competition and models of Decision making
III. Noise and the neural code
9. Noise and variability of spike trains (point processes, renewal process, interval distribution)
10: Variance of membrane potentials and Spike Response Models
11. Population dynamics: Fokker-Planck equation
12. Fitting Neural Models to Data
13. Neural Manifolds and low-dimensional dynamics
14. Summary

Keywords

neural networks, neuronal dynamics, computational neuroscience, mathematical modeling in biology, applied mathematics, brain, cognition, neurons, memory, learning, plasticity

Learning Prerequisites

Required courses

undergraduate math at the level of electrical engineering or physics majors
undergraduate physics.

Recommended courses

Analysis I-III, linear algebra, probability and statistics
For SSV students: Dynamical Systems Theory for Engineers or "Mathematical and Computational Models in Biology"

Important concepts to start the course

Differential equations, Linear equations,

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze two-dimensional models in the phase plane
- Solve linear one-dimensional differential equations
- Develop a simplified model by separation of time scales
- Analyze connected networks in the mean-field limit
- Predict outcome of dynamics
- Prove stability and convergence
- Describe neuronal phenomena
- Test model concepts in simulations

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Write a scientific or technical report.

Teaching methods

- Video lectures, combined with classroom teaching, classroom discussions, exercises and miniproject. One of the two exercise hours is integrated into the lectures.
- Short mooc-style videos are available as support
- Textbook available as support

Expected student activities

-participate in ALL in-class exercises.
- do all homework exercises (paper-and-pencil)
- study video lectures if you miss a class
- study suggested textbook sections for in-depth understanding of material
- submit miniprojects

Assessment methods

Written exam (70%) & miniproject (30%)
The miniproject is done in teams of 2 students.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	The teacher is available during the breaks of the class. Some exercises are integrated in class in the presence of the teacher and the teaching assistants.

Resources

Bibliography

Gerstner, Kistler, Naud, Pansinski : Neuronal Dynamics, Cambridge Univ. Press 2014

Ressources en bibliothèque

- Neuronal dynamics: from single neurons to networks and models of cognition / Wulfram Gerstner, Werner M. Kistler, Richard Naud, Liam Paninski

Websites

- https://neuronaldynamics.epfl.ch/
- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

Moodle Link

- https://go.epfl.ch/NX-465

Videos

- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

EPFL

CS-413

Computational photography

Süsstrunk Sabine

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The students will gain the theoretical knowledge in computational photography, which allows recording and processing a richer visual experience than traditional digital imaging. They will also execute practical group projects to develop their own computational photography application.

Content

Computational photography is the art, science, and engineering of creating a great (still or moving) image. Information is recorded in space, time, across visible and invisible radiation and from other sources, and then post-processed to produce the final - visually pleasing - result.
Basics: Human vision system, Light and illumination, Geometric optics, Color science, Sensors, Digital camera systems.
Generalized illumination: Structured light, High dynamic range (HDR) imaging, Time-of-flight.
Generalized optics: Coded Image Sensing, Coded aperture, Focal stacks.
Generalized sensing: Low light imaging, Depth imaging, Plenoptic imaging, Light field cameras.
Generalized processing: Super-resolution, In-painting, Compositing, Photomontages, Panoramas, HDR imaging, Multi-wavelength imaging, Dynamic imaging.
Generalized display: Stereoscopic displays, HDR displays, 3D displays, Mobile displays.
Deep Learning for image resoration and image enhancement.

Keywords

Computational Photography, Coded Image Sensing, Non-classical image capture, Multi-Image & Sensor Fusion, Mobile Imaging, Machine Learning

Learning Prerequisites

Required courses

- A basic Signal Processing, Image Processing, and/or Computer Vision course.
- Linear Algebra.

Recommended courses

- Computer Vision.
- Signal Processing
- Machine Learning.

Important concepts to start the course

- Basic signal/image processing.
- Basic computer vision.
- Basic programming (Python, iOS, Android).

Learning Outcomes

By the end of the course, the student must be able to:

- Identify the main components of a computational photography system.
- Contextualise the main trends in computational optics, sensing, processing, and displays.
- Create a computational photography application on a mobile platform.
- Design a computational photography solution to solve a particular imaging task.
- Assess / Evaluate hardware and software combinations for their imaging performance.
- Formulate computational photography challenges that still need to be resolved.
- Create a computational photography application.

Transversal skills

- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

The course consists of 2 hours of lectures per week that will cover the theoretical basics. An additional 2 hours per week are dedicated to a group project designing, developing, and programming a compuational photography application on a mobile plateform (iOS, Android).

Expected student activities

The studens is expected to attend the class and actively participate in the practical group project, which requires coding on either Android or iOS plateform. The student is also required to read the assigned reading material (book chapters, scientific articles).

Assessment methods

The theoretical part will be evaluated with an oral exam at the end of the semester, and the practical part based on the students' group projects

Resources

Bibliography

- Selected book chapters
- Course notes (on moodle)
- Links to relevant scientific articles and on-line resources will be given on moodle.

Moodle Link

- https://go.epfl.ch/CS-413

EPFL

CS-442

Computer vision

Fua Pascal

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Computer Vision aims at modeling the world from digital images acquired using video or infrared cameras, and other imaging sensors. We will focus on images acquired using digital cameras. We will introduce basic processing techniques and discuss their field of applicability.

Content

Introduction

- History of Computer Vision
- Human vs Machine Vision
- Image formation

Extracting 2D Features

- Contours
- Texture
- Regions

3D Shape Recovery

- From one single image
- From multiple images

Learning Outcomes

By the end of the course, the student must be able to:

- Choose relevant algorithms in specific situations
- Perform simple image-understanding tasks

Teaching methods

Ex cathedra lectures and programming exercises using Python.

Assessment methods

With continuous control

Resources

Bibliography

• R. Szeliski, Computer Vision: Computer Vision: Algorithms and Applications, 2010.
• A. Zisserman and R. Hartley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

Ressources en bibliothèque

• Multiple View Geometry in Computer Vision / Zisserman
• Computer Vision: Algorithms and Applications / Szeliski

Moodle Link

• https://go.epfl.ch/CS-442

EPFL

CS-453 **Concurrent computing**

Guerraoui Rachid

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

With the advent of modern architectures, it becomes crucial to master the underlying algorithmics of concurrency. The objective of this course is to study the foundations of concurrent algorithms and in particular the techniques that enable the construction of robust such algorithms.

Content

Model of a parallel system
Multicore and multiprocessors architecture

Processes and objects

Safety and liveness

Parallel programming
Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

Register Implementations
Safe, regular and atomic registers
Counters General and limited operations
Atomic counters and snapshots

Hierarchy of objects
The FLP impossibility
The consensus number
Universal constructions

Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Anonymous computing

Fault-tolerant shared-memory computing

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses

ICC, Operating systems

Recommended courses

This course is complementary to the Distributed Algorithms course

Important concepts to start the course

Processes, threads, datas structures

Learning Outcomes

By the end of the course, the student must be able to:

- Reason in a precise manner about concurrency
- Design a concurrent algorithm
- Prove a concurrent algorithm
- Implement a concurrent system

Teaching methods

Lectures, exercises and practical work

Expected student activities

Final exam
Project

Assessment methods

Final exam (theory) and project (practice)

Resources

Notes/Handbook

Algorithms for Concurrent Systems, R. Guerraoui and P. Kouznetsov

Moodle Link

- https://go.epfl.ch/CS-453

COM-401 Cryptography and security

Vaudenay Serge

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

This course introduces the basics of cryptography. We review several types of cryptographic primitives, when it is safe to use them and how to select the appropriate security parameters. We detail how they work and sketch how they can be implemented.

Content

1. **Ancient cryptography**: Vigenère, Enigma, Vernam cipher, Shannon theory
2. **Diffie-Hellman cryptography**: algebra, Diffie-Hellman, ElGamal
3. **RSA cryptography**: number theory, RSA, factoring
4. **Elliptic curve cryptography**: elliptic curves over a finite field, ECDH, ECIES, pairing
5. **Symmetric encryption**: block ciphers, stream ciphers, exhaustive search
6. **Integrity and authentication**: hashing, MAC, birthday paradox
7. **Public-key cryptography**: cryptosystem, digital signature, post-quantum cryptography
8. **Trust establishment**: password-based cryptography, secure communication, trust setups
9. **Case studies**: WiFI, bitcoin, mobile telephony, WhatsApp, EMV, Bluetooth, biometric passport, TLS

Keywords

cryptography, encryption, secure communication

Learning Prerequisites

Required courses

MATH-310 Algebra
MATH-232 Probability and statistics for IC
CS-250 Algorithms I

Recommended courses

COM-301 Computer security and privacy

Important concepts to start the course

- Mathematical reasoning
- Probabilities

• Algebra, arithmetics
• Algorithmics

Learning Outcomes

By the end of the course, the student must be able to:

• Choose the appropriate cryptographic primitive in a security infrastructure
• Judge the strength of existing standards
• Assess / Evaluate the security based on key length
• Implement algorithms manipulating big numbers and use number theory
• Use algebra and probability theory to analyze cryptographic algorithms
• Identify the techniques to secure the communication and establish trust

Teaching methods

ex-cathedra

Expected student activities

• active participation during the course
• take notes during the course
• do the exercises during the exercise sessions
• complete the regular tests and homework
• read the material from the course
• self-train using the provided material
• do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

• homework (30%)
• regular graded tests (30%)
• midterm exam (40%)

Final exam averaged (same weight) with the contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

• Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
• A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.

Ressources en bibliothèque

- A Classical Introduction to Cryptography / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-401

Videos

- https://mediaspace.epfl.ch/channel/COM-401+Cryptography+and+security

Prerequisite for

- Advanced cryptography (COM-501)
- Student seminar: security protocols and applications (COM-506)

EPFL

COM-480 **Data visualization**

Vuillon Laurent Gilles Marie

Cursus	Sem.	Type
Computational and Quantitative Biology		Obl.
Computational biology minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Understanding why and how to present complex data interactively in an effective manner has become a crucial skill for any data scientist. In this course, you will learn how to design, judge, build and present your own interactive data visualizations.

Content

Tentative course schedule
Week 1: Introduction to Data visualization Web development
Week 2: Javascript
Week 3: More Javascript
Week 4: Data Data driven documents (D3.js)
Week 5: Interaction, filtering, aggregation (UI /UX). Advanced D3 / javascript libs
Week 6: Perception, cognition, color Marks and channels
Week 7: Designing visualizations (UI/UX) Project introduction Dos and don’ts for data-viz
Week 8: Maps (theory) Maps (practice)
Week 9: Text visualization
Week 10: Graphs
Week 11: Tabular data viz Music viz
Week 12: Introduction to scientific visualisation
Week 13: Storytelling with data / data journalism Creative coding
Week 14: Wrap-Up

Keywords

Data viz, visualization, data science

Learning Prerequisites

Required courses

CS-250 Algorithms I (BA)
CS-401 Applied data analysis (MA)

Recommended courses

CS-486 Interaction design (MA)
CS-214 Software construction (BA)

Important concepts to start the course

Being autonomous is a prerequisite, we don't offer office hours and we won't have enough teaching assistants (you've been warned!).
Knowledge of one of the following progrmaming language such as C++, Python, Scala.
Familiarity with web-development (you already have a blog, host a webiste). Experience with HTML5, Javascript is a strong plus for the course.

Learning Outcomes

By the end of the course, the student must be able to:

- Judge visualization in a critical manner and suggest improvements.
- Design and implement visualizations from the idea to the final product according to human perception and cognition
- Know the common data-viz techniques for each data domain (multivariate data, networks, texts, cartography, etc) with their technical limitations
- Create interactive visualizations int he browser using HTM5 and Javascript

Transversal skills

- Communicate effectively, being understood, including across different languages and cultures.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra lectures, exercises, and group projects

Expected student activities

- Follow lectures
- Read lectures notes and textbooks
- Create an advanced data-viz in groups of 3.
- Answer questions assessing the evolution of the project.
- Create a 2min screencast presentation of the viz.
- Create a process book for the final data viz.

Assessment methods

- Data-viz (35%)
- Technical implementation (15%)
- Website, presentation, screencast (25%)
- Process book (25%)

Resources

Bibliography

Visualization Analysis and Design by Tamara Munzner, CRC Press (2014). Free online version at EPFL.
Interactive Data Visualization for the Web by Scott Murray O'Reilly (2013) - D3 - Free online version.
The Truthful Art: Data, Charts, and Maps for Communication by Cairo, Alberto. Royaume-Uni, New Riders, (2016).
Data Visualisation: A Handbook for Data Driven Design by Kirk, Andy. Royaume-Uni, SAGE Publications, (2019).

Ressources en bibliothèque

- Data Visualisation / Kirk
- Visualization Analysis and Design / Munzner
- The Truthful Art / Cairo
- Interactive Data Visualization for the Web / Murray [2 ed. 2017]

Notes/Handbook

Lecture notes

Moodle Link

- https://go.epfl.ch/COM-480

EPFL

CS-438 **Decentralized systems engineering**

Borsò-Tan Pierluca, Ford Bryan Alexander

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

A decentralized system is one that works when no single party is in charge or fully trusted. This course teaches decentralized systems principles while guiding students through the engineering of their own decentralized system featuring messaging, file sharing, encryption, and blockchain concepts.

Content

- Addressing, Forwarding, Routing. Peer-to-peer communication.
- Information gossip. UseNet: technical, security, and social lessons. Randomized rumor-mongering and anti-entropy algorithms.
- Trust and Reputation. Authorities, trust networks. Sybil attacks and defenses.
- Naming and search. Request flooding. Hierarchical directories, landmark routing. Self-certifying identities. Distributed hash tables.
- Distributed consensus, distributed ledgers (blockchains), and cryptocurrencies.
- Anonymous Communication. Onion routing, mix networks. Dining cryptographers. Voting, verifiable shuffles, homomorphic encryption. Anonymous disruption.
- Fireproofing Alexandria: Decentralized Storage. Replication. Parity, erasure coding. Renewal. Digital preservation.
- Content Distribution. Opportunistic caching (FreeNet). Content integrity: hash trees, hash file systems. Convergent encryption. Swarming downloads: BitTorrent. Free-riding, incentives.
- Gaining perspective. Spam, malicious content. Review/moderation and reputation systems. Leveraging social networks (Peerspective). Balancing local and global viewpoints.
- Decentralized Collaboration. Network file systems, version management. Consistency.
- Consistency Models. Disconnected operation, eventual consistency, conflict resolution.
- Distributed Consensus. Paxos. Accountability (PeerReview). Byzantine fault tolerance.
- Mobile Code. Smart contract systems. Privacy: trusted computing, fully homomorphic encryption. Decentralized virtual organizations.
- Securing Decentralized Systems: threat modelling, use of threshold cryptosystems.
- Going into production: Quality assurance, chaos engineering and operations.

Learning Prerequisites

Required courses

- CS-202 Computer systems or COM-208 Computer networks

• COM-301 Computer security and privacy

Learning Outcomes

By the end of the course, the student must be able to:

- Implement decentralized systems via hands-on coding, debugging, and testing
- Design effective testing strategies for distributed and decentralized systems
- Design practical distributed and decentralized systems

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Assess progress against the plan, and adapt the plan as appropriate.
- Take account of the social and human dimensions of the engineering profession.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Lectures: The course's lectures will present and discuss challenges, known techniques, and open questions in decentralized system design and implementation. Lectures will often be driven by examination of real decentralized systems with various purposes in widespread use the past or present, such as UseNet, IRC, FreeNet, Tor, BitTorrent, and Bitcoin. Throughout the course we will explore fundamental security and usability challenges such as decentralized identification and authentication, denial-of-service and Sybil attacks, and maintenance of decentralized structures undergoing rapid changes (churn).
Labs: During the semester, students will develop a small but usable peer-to-peer communication application that reflects a few of the important design principles and techniques to be explored in the course, such as gossip, distributed hash tables, consensus algorithms, and cryptocurrencies. The labs will designed so that solutions can initially be tested individually on private, virtual networks running on one machine, then tested collectively by attempting to make different students' solutions interoperate on a real network.

Warning: This course is extremely programming-intensive. Students should be strong and confident in their programming skills in general, and be willing to spend substantial time outside of class debugging difficult distributed concurrency bugs and other challenges. TAs will be available to help at the exercise sessions, but *they will not solve your problems or debug your code for you*.

Expected student activities

Students will be expected to attend lectures to understand the concepts needed for the course, but the main workload will be actual hands-on programming assignments, which the students will perform on their own during the first part of the course and in teams during the final project-oriented part of the course.

Assessment methods

- Programming assignment grading (evaluating function, performance, correctness and implementation quality): 40%
- Team project grading (evaluating for scope, implementation quality, testing, evaluation, documentation and report): 30%
- Written exam: 30%

Resources

Virtual desktop infrastructure (VDI)

No

Moodle Link

- https://go.epfl.ch/CS-438

EPFL

EE-559

Deep learning

Cavallaro Andrea

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in Quantum Science and Engineering	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	**150**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This course explores how to design reliable discriminative and generative neural networks, the ethics of data acquisition and model deployment, as well as modern multi-modal models.

Content

This course equips students with a comprehensive foundation of modern deep learning, enabling students to design and train discriminative and generative neural networks for a wide range of tasks. Topics include:

- Deep learning applications (natural language processing, computer vision, audio processing, biology, robotics, science), principles and regulations
- Loss functions, data and labels, data provenance
- Training models: gradients and initialization
- Generalization and performance
- Transformers
- Graph neural networks
- Generative adversarial networks
- Variational autoencoders
- Diffusion models
- Multi-modal models
- Interpretability, explanations, bias and fairness

Keywords

machine learning, neural networks, deep learning, python

Learning Prerequisites

Required courses

- Basics in probabilities and statistics
- Linear algebra
- Differential calculus
- Python programming

Recommended courses

- Basics in optimization
- Basics in algorithmic
- Basics in signal processing

Important concepts to start the course

Discrete and continuous distributions, normal density, law of large numbers, conditional probabilities, Bayes, PCA, vector, matrix operations, Euclidean spaces, Jacobian, Hessian, chain rule, notion of minima, gradient descent, computational costs, Fourier transform, convolution.

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret the performance of a deep learning model
- Analyze the limitations of a deep learning model
- Justify the choices for training and testing a deep learning model
- Propose new solutions for a given application

Transversal skills

- Respect relevant legal guidelines and ethical codes for the profession.
- Take account of the social and human dimensions of the engineering profession.
- Design and present a poster.
- Make an oral presentation.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex-cathedra lectures, class discussion, exercises (using python), group project.

Expected student activities

Attendance to lectures, participation in discussions, completing exercises, completing a project, reading written material (scientific papers and books).

Assessment methods

Excercises and group project.

Resources

Références suggérées par la bibliothèque

- Deep learning, Goodfellow, MIT Press, 2016

Notes/Handbook

Not mandatory: http://www.deeplearningbook.org/

Moodle Link

- https://go.epfl.ch/EE-559

EPFL

CS-456 **Deep reinforcement learning**

Gulcehre Caglar

Cursus	Sem.	Type
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Obl.
Financial engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course provides an overview and introduces modern methods for reinforcement learning (RL.) The course starts with the fundamentals of RL, such as Q-learning, and delves into commonly used approaches, like PPO and DQN. The course will introduce students to practical applications of RL.

Content

- *Introduction and Overview (What is RL?)*

- *An overview of neural networks and deep learning approaches*

- *Deep learning frameworks*

- *Supervised learning of behaviors (behavior cloning)*

- *Value function methods and related theory*
- *Policy gradient methods and related theory*
- *Actor-Critic Algorithms (A2C, A3C)*
- *Deep RL with Q functions (DQN, R2D2)*
- *Deep Policy Gradient and Optimization methods (PPO, TRPO, Impala, MPO)*
- *Model-based RL and Planning (Alphago, Alphazero, Dreamer)*
- *Exploration and credit assignment in Deep RL*
- *Offline RL (BVE, CQL, CRR, ...)*
- *Deep Imitation learning and Learning from demonstrations (DAGGER, DQFD, R2D3, Learning from play, Third*

person imitation)

- *RL from human feedback and alignment (InstructGPT, DPO, ReST, etc.)*
- *Advanced continuous control approaches (DDPG, D4PG, SAC)*
- *A selection of extra topics from:*

- MPO, IMPALA
- Distributional RL
- Multi-agent RL (Centralized Training, Decentralized Execution)

Keywords

Deep learning, reinforcement learning, TD learning, SARSA, Actor-Critic Networks, policy gradients, alphago, alphastar, planning, alignment, RLHF, PPO

Learning Prerequisites

Required courses

- Analysis I, II
- Linear Algebra
- Probability and statistics (MATH-232)
- Algorithms I (CS-250)

Recommended courses

- Introduction to machine learning (CS-233)
- Machine learning (CS-433)

Important concepts to start the course

- *Regularization in machine learning,*
- *Gradient descent. Stochastic gradient descent.*
- *Expectation, statistics*
- *Linear algebra and probabilities*
- *programming*

Learning Outcomes

By the end of the course, the student must be able to:

- Apply Understand and define basic problems and tasks in reinforcement learning (like Markov decision process, model-based and model-free RL, on-policy vs off-policy RL)
- Assess / Evaluate Formulate a real-world problem as an RL setting to apply the approaches taught in the class.
- Elaborate Implement standard deep RL algorithms.
- Judge Understand the failure modes of these models and learning algorithms.
- Propose Read and review academic papers to understand their contributions and learn how to evaluate them critically.
- Apply Students gain the skills and knowledge necessary to tackle complex problems in autonomous robotics, game-playing, and other domains through lectures, hands-on coding exercises, practical applications, and course

projects.

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Manage priorities.

Teaching methods

- Lectures
- Lab sessions
- Individual course projects
- Paper reading
- Group projects

Expected student activities

- Work on miniproject
- Solve all exercises
- Attend all lectures and take notes during lecture, participate in quizzes.
- If you cannot attend a lecture, then you must read the recommended book chapters
- Work on a project

Assessment methods

- Written final exam (25%)
- Assignments (25%)
- Course project (50%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	TAs are available during exercise sessions. Every week one of the exercises is run as 'integrated exercise' during the lecture.

Resources

Bibliography

- Textbook: Reinforcement Learning by Sutton and Barto (MIT Press). Pdfs of the preprint version of the book are availble online

Ressources en bibliothèque

- Reinforcement Learning / Sutton

Moodle Link

- https://go.epfl.ch/CS-456

CS-472

Design technologies for integrated systems

De Micheli Giovanni

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
MNIS	MA3	Obl.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Labs	2 weekly
Number of positions	

Summary

Hardware compilation is the process of transforming specialized hardware description languages into circuit descriptions, which are iteratively refined, detailed and optimized. The course presents algorithms, tools and methods for hardware compilation and logic synthesis.

Content

The course will present the most outstanding features of hardware compilation, as well as the techniques for optimizing logic representations and networks. The course gives a novel, uptodate view of digital circuit design. Practical sessions will teach students the use of current design tools.Syllabus1) Modeling languages and specification formalisms;2) High-level synthesis and optimization methods (scheduling, binding, data-path and control synthesis);3) Representation and optimization of combinational logic functions (encoding problems, binary decision diagrams);4) Representation and optimization of multiple-level networks (algebraic and Boolean methods, "don't care" set computation, timing verification and optimization);5) Modeling and optimization of sequential functions and networks (retiming);6) Semicustom libraries and library binding.

Keywords

Hardware, VLSI, Synthesis, Optimization, Algorithms

Learning Prerequisites

Required courses

No specific course

Recommended courses

Good knowledge of digital design, algorithm design and programming.

Important concepts to start the course

Good knowledge of digital design, algorithm design and programming.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize important problems in digital design
- Examine and evaluate available design tools and methods
- Decide upon a design tool flow to perform a digital design

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.

Assessment methods

Continuous control :
Homework : 30 %, Project 10 %, Midterm test : 25 %,
End term test : 35 %

Resources

Bibliography

G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw'Hill.

Ressources en bibliothèque

- Synthesis and Optimization of Digital Circuits / De Micheli

Moodle Link

- https://go.epfl.ch/CS-472

EPFL

CS-411

Digital education

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Winter, Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course addresses the relationship between specific technological features and the learners' cognitive processes. It also covers the methods and results of empirical studies: do student actually learn due to technologies? I will be given twice in 2025 (Spring + Fall)

Content

- *Learning theories and learning processes.*
- *Types of learning technlogies*
- *Instructional design: methods, patterns and principles.*
- *On-line education.*
- *Effectiveness of learning technologies.*
- *Methods for empirical research.*
- *Computational thinkiing skills*
- *Maker spaces*

Keywords

learning, pedagogy, teaching, online education, maker spaces

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the learning processes triggered by a technology-based activity
- Explain how a technology feature influences learning processes
- Elaborate a study that measures the learning effects of a digital environment
- Select appropriately a learning technology given the target audience and the expected learning outcomes

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Take account of the social and human dimensions of the engineering profession.

- Write a scientific or technical report.

Teaching methods

The course will combine participatory lectures with a project on designing a learning environment, using it in a controlled experiment, analysing the learning effects with statistical methods and writing a report.

Expected student activities

The project will include a few milestones to be delivered along the semester.

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-411

EPFL

CS-451 Distributed algorithms

Guerraoui Rachid

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	3 weekly
Number of positions	

Summary

Computing is nowadays distributed over several machines, in a local IP-like network, a cloud or a P2P network. Failures are common and computations need to proceed despite partial failures of machines or communication links. This course will study the foundations of reliable distributed computing.

Content

Reliable broadcast
Causal Broadcast
Total Order Broadcast
Consensus
Non-Blocking Atomic Commit
Group Membership, View Synchrony
Terminating Reliable Broadcast
Shared Memory in Message Passing Systems
Byzantine Fault Tolerance
Self Stabilization
Population protocols (models of mobile networks)
Bitcoin, Blockchain
Distributed Machine Learning
Gossip

Keywords

Distributed algorithms, checkpointing, replication, consensus, atomic broadcast, ditributed transactions, atomic commitment, 2PC, Machine Learning

Learning Prerequisites

Required courses

Basics of Algorithms, networking and operating systems

Recommended courses

The lecture is orthogonal to the one on concurrent algorithms: it makes a lot of sense to take them in parallel.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose an appropriate abstraction to model a distributed computing problem
- Specify the abstraction
- Present and implement it
- Analyze its complexity
- Prove a distributed algorithm
- Implement a distributed system

Teaching methods

Ex cathedera
Lectures, exercises and practical work

Assessment methods

Final exam (theory)
Project (practice)

Resources

Ressources en bibliothèque

- Introduction to reliable and secure distributed programming / Cachin

Notes/Handbook

Reliable and Secure Distributed Programming
Springer Verlag
C. Cachin, R. Guerraoui, L. Rodrigues

Moodle Link

- https://go.epfl.ch/CS-451

EPFL

CS-423

Distributed information systems

Aberer Karl

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course introduces the foundations of information retrieval, data mining and knowledge bases, which constitute the foundations of today's Web-based distributed information systems.

Content

Information Retrieval

1. Information Retrieval - Introduction
2. Text-Based Information Retrieval (Boolean, Vector space, probabilistic)
3. Inverted Files
4. Distributed Retrieval
5. Query Expansion
6. Embedding models (LSI, word2vec)
7. Link-Based Ranking

Mining Unstructured Data

1. Document Classification (knn, Naive Bayes, Fasttext, Transformer models)
2. Recommender Systems (collaborative filtering, matrix factorization)
3. Mining Social Graphs (modularity clustering, Girvan-Newman)

Knowledge Bases

1. Semantic Web
2. Keyphrase extraction
3. Named entity recognition
4. Information extraction
5. Taxonomy Induction
6. Entity Disambiguation
7. Label Propagation
8. Link Prediction

Learning Prerequisites

Recommended courses

Introductory courses to databases and machine learning are helpful, but not required.
Programming skills in Python are helpful, but not required.

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the main tasks performed by information systems, namely data, information and knowledge management
- Apply collaborative information management models, like crowd-sourcing, recommender systems, social networks
- Apply knowledge models, their representation through Web standards and algorithms for storing and processing semi-structured data
- Apply fundamental models and techniques of text retrieval and their use in Web search engines
- Apply main categories of data mining techniques, local rules, predictive and descriptive models, and master representative algorithms for each of the categories

Teaching methods

Ex cathedra + programming projects (Python)

Assessment methods

60% Continuous evaluations of projects with bonus system during the semester
40% Final written exam (180 min) during exam session

Resources

Moodle Link

- https://go.epfl.ch/CS-423

EPFL

ENG-466

Distributed intelligent systems

Martinoli Alcherio

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	Oral
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Project	1 weekly
Labs	2 weekly
Number of positions	**40**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The goal of this course is to provide methods and tools for modeling distributed intelligent systems as well as designing and optimizing coordination strategies. The course is a well-balanced mixture of theory and practical activities.

Content

- Introduction to key concepts such as self-organization and tools used in the course
- Examples of natural, artificial and hybrid distributed intelligent systems
- Modeling methods: sub-microscopic, microscopic, macroscopic, multi-level; spatial and non-spatial; mean field, approximated and exact approaches
- Machine-learning methods: single- and multi-agent techniques; expensive optimization problems and noise resistance
- Coordination strategies and distributed control: direct and indirect schemes; algorithms and methods; performance evaluation
- Application examples in distributed sensing and action

Keywords

Swarm intelligence, artificial intelligence, machine-learning, distributed robotics, swarm robotics, multi-robot systems, sensor networks, modeling, control, optimization

Learning Prerequisites

Required courses

Fundamentals in analysis, probability, and programming for both compiled and interpreted languages

Recommended courses

Basic knowledge in statistics, programming language used in the course (C, Matlab, Python), and signals and systems

Learning Outcomes

By the end of the course, the student must be able to:

- Design control algorithms
- Formulate a model at different level of abstraction for a distributed intelligent system
- Analyze a model of a distributed intelligent system
- Analyze a distributed coordination strategy/algorithm
- Design a distributed coordination strategy/algorithm
- Implement code for single robot and multi-robot systems
- Carry out systematic performance evaluation of a distributed intelligent system
- Apply modeling and design methods to specific problems requiring distributed sensing and action
- Optimize a controller or a set of possibly coordinated controllers using model-based or data-driven methods

Transversal skills

- Demonstrate a capacity for creativity.
- Access and evaluate appropriate sources of information.
- Collect data.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

Ex-cathedra lectures, assisted exercises, and course project in teams

Expected student activities

Attending lectures, carrying out exercises and the course project, and reading handouts.

Assessment methods

Oral exam (50%) with continuous assessment during the semester based on course project (50%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

Lecture notes, selected papers and book chapters distributed at each lecture.

Websites

- https://disal.epfl.ch/teaching/distributed_intelligent_systems/

Moodle Link

- https://go.epfl.ch/ENG-466

Videos

- https://mediaspace.epfl.ch/channel/ENG-466+Distributed+intelligent+systems+-+Spring+2022/29956

Prerequisite for

R&D activities in engineering

COM-502

Dynamical system theory for engineers

Thiran Patrick

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Mechanical engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal

Summary

Linear and nonlinear dynamical systems are found in all fields of science and engineering. After a short review of linear system theory, the class will explain and develop the main tools for the qualitative analysis of nonlinear systems, both in discrete-time and continuous-time.

Content

- **Introduction:** Dynamics of linear and non linear systems. Definitions; Unicity of a solution; Limit Sets, Attractors.
- **Linear Systems:** Solutions; Stability of autonomous systems, Geometrical analysis, connection with frequency domain analysis.
- **Nonlinear Systems:** Solutions; Examples. Large-scale notions of stability (Lyapunov functions). Hamiltonian systems, gradient systems. Small-scale notions of stability (Linearization; stability and basin of attraction of an equilibrium point, stability of periodic solutions, Floquet Multipliers). Graphical methods for the analysis of low-dimensional systems. Introduction to structural stability, Bifurcation theory. Introduction to chaotic systems (Lyapunov exponents); time permitting: a review of tools of measure theory to compute Lyapunov exponents.
- The class is methodology-driven. It may present some limited examples of applications, but it is not application-driven.

Keywords

Dynamical Systems, Attractors, Equilibrium point, Limit Cycles, Stability, Lyapunov Functions, Bifurcations, Lyapunov exponents.

Learning Prerequisites

Required courses

- Linear algebra (MATH 111 or equivalent).
- Analysis I, II, III (MATH 101, 106, 203 or equivalent).
- Circuits & Systems II (EE 205 or equivalent) or a Systems & Signals class (MICRO 310/311 or

equivalent).

Recommended courses

• A first-year Probabilty class, such as MATH-232, MATH-231, MATH-234(b), MATH-234(c), or equivalent.
• Analysis IV (MATH 207 or equivalent)

Important concepts to start the course

• Linear Algebra (vector spaces, matrix operations, including inversion and eigendecomposition).
• Calculus (linear ordinary differential equations; Fourier, Laplace and z-Transforms).
• Basic notions of topology.
• Basic notions of probability.

Learning Outcomes

By the end of the course, the student must be able to:

• Analyze a linear or nonlinear dynamical system.
• Anticipate the asymptotic behavior of a dynamical system.
• Assess / Evaluate the stability of a dynamical system.
• Identify the type of solutions of a dynamical sytem.
• Analyze a linear or nonlinear dynamical system
• Anticipate the asymptotic behavior of a dynamical system
• Assess / Evaluate the stability of a dynamical system
• Identify the type of solutions of a dynamical system

Teaching methods

• Lectures (blackboard), 2h per week
• Exercise session, 1h per week

Expected student activities

Exercises in class and at home (paper and pencil, and Matlab)

Assessment methods

1. Mid-term 20% (conditionally on the Covid situation allowing for it to be taken at EPFL).
2. Final exam 80%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Course notes; textbooks given as reference on the moodle page of the course.

Notes/Handbook

Course notes, exercises and solutions provided on the moodle page of the course.

Moodle Link

- https://go.epfl.ch/COM-502

Prerequisite for

Classes using methods from dynamical systems.

EPFL

CS-476

Embedded system design

Kluter Ties Jan Henderikus

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Labs	2 weekly
Number of positions	

Summary

Hardware-software co-design is a well known concept in embedded system design.It is also a concept required in designing FPGA-accelerators in data-centers.This course teaches how to transform algorithms in smart hardware-software solutions.

Content

High-level architectures:

- FIFO's, LIFO's, ring-buffers, and ping-pong buffers.
- FSM-D (finite state machine data-path) structures.
- Stream processing.

Acceleration methods:

- Custom instruction set extensions.
- Hardware accelerators
- Compiler optimizations.

Implementation methods:

- Hardware-software co-design.
- Timing closure.
- Virtual prototyping.
- Bare metal versus usage of an RTOS.

Learning Prerequisites

Required courses

•

Recommended courses

- Architecture-aware programming
- CS-200 Computer Architecture or CS-208 Computer Architecture I and CS-209 Computer Architecture II

Important concepts to start the course

- C/C++ programming skills

- Verilog/VHDL description skills

Learning Outcomes

By the end of the course, the student must be able to:

- Design buffers to account for different read and write behaviors
- Understand the concept of FSM-D's and use this concept to design accelerators
- Understand the concept of stream processing and be able to implement a stream processor
- Design and optimize an embedded system on FPGA given a set of prerequisites

Teaching methods

- First 9 weeks : theory with small projects and reports
- Last 5 weeks : mini project in groups of 2 students with final demonstration and presentation

Expected student activities

- Perform in groups of two students small projects to put the theory into practice
- Optimize a real world problem in a final project, explain the choices made, and present the results

Assessment methods

- Lab reports : 50%
- Final mini-project : 50%

Supervision

Office hours	No
Assistants	Yes
Others	Electronic forum and Moodle

Resources

Moodle Link

- https://go.epfl.ch/CS-476

EPFL

DH-415

Ethics and law of AI

Rochel Johan Robert

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Minor in digital humanities, media and society	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - HEC	H	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	**100**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This master course enables students to sharpen their proficiency in tackling ethical and legal challenges linked to Artificial Intelligence (AI). Students acquire the competence to define AI and identify ethical and legal questions linked to its conception and increased use in society.

Content

AI is used as shortcut-concept to identify a number of computational systems producing intelligent behavior, i.e., complex behavior conducive to reaching goals. AI systems are increasingly used across society. They raise conceptual issues (how to define AI?), technological-ethical issues (how should AI systems be conceived?), legal issues (how to define the responsibility of an AI system? how to regulate AI?) and social-political issues (which justice questions does the deployment of AI raise?)
The following issues will be dealt with:
- What is ethics?
- What is an AI system?
- Who is responsible for the actions of an AI system?
- What are the most pressing ethical questions in the phase of conception of AI systems?
- How should we design AI system in order to overcome ethical-legal challenges?
- Should we regulate AI?
- How should we address the consequences of the wide deployment of AI systems?

Keywords

artificial intelligence, ethics, law, data, regulation, responsibility

Learning Prerequisites

Required courses

No pre-requirement

Learning Outcomes

By the end of the course, the student must be able to:

- Define the concept of AI
- Assess / Evaluate the contexts in which AI is deployed
- Systematize general principles (law and ethics)
- Analyze the different senses/conceptions/interpretations of agency, autonomy and responsibility
- Develop principles for the conception of AI system
- Distinguish legal and ethical arguments

Transversal skills

- Demonstrate the capacity for critical thinking
- Take account of the social and human dimensions of the engineering profession.
- Respect relevant legal guidelines and ethical codes for the profession.
- Use a work methodology appropriate to the task.

Teaching methods

The course will be organized as an interactive and participative course. For the weekly course: students have to read texts and to be ready for critical discussion. For the weekly exercise: students have to engage in group discussions. The course requires reading complex texts in English.

Expected student activities

Weekly reading of preparatory texts
Active participation in class, both course and exercise

Assessment methods

Students will be assessed in the following way :
- Mid-term: students will have to answer 2 questions during class (compulsory, no grading)
- Open book written exam during the term (100% of the grade)

Supervision

Office hours	No
Assistants	Yes
Others	Upon appointment with Dr Rochel

Resources

Bibliography

All resources will be made available on moodle.
To start with: *AI Ethics* (Mark Coekelbergh, MIT 2020)

Ressources en bibliothèque

- AI Ethics / Mark Coeckelbergh

Moodle Link

- https://go.epfl.ch/DH-415

EPFL

CS-489 **Experience design**

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Project	4 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

As we move towards a design economy, the success of new products, systems and services depend increasingly on the excellence of personal experience. This course introduces students to the notion and practice of experience design following a hands-on, studio-based approach.

Content

Experience design in practice encompasses the collection, analysis and design of users experiences based on a deep understanding of the context concerned. We will examine these processes using a series of mini-workshops, to rapidly iterate on multiple design experience options. The goal is to create a meaningful, interactive, data-driven (and possibly AI-assisted) digital interface and physical prototypes for new experiences.
We explore questions at the intersection of physical and digital architecture through an experience design approach, involving: (1) a mapping of the social dynamics surrounding an experience; (2) a critical analysis of the geographical and temporal flows (experience journeys); and (3) a detailed evaluation of the experience touch points. Based on this experience diagnosis, we propose alternative designs of experience blueprints that combine physical and digital touch points which in turn will constitute the elements of future typologies.
Our particular focus will be on information intensive typologies in the contemporary city, such as museums, libraries, airports, banks, boutiques, governments, hospitals and homes. Each year, we will investigate a different typology. Digital interfaces and augmented artifacts will be considered as possible alternatives to reconfigure the senses of perception, redistribute time, and reorchestrate the configuration of social, emotional and spatial experiences.
The seminar will combine students from both IC and ENAC to work together in a real interdisciplinary process.

Keywords

User Experience (UX) Design, Design Thinking, Journey Mapping, Optioneering, Critical Prototyping, Value Proposition

Learning Prerequisites

Required courses

Bachelor in Computer Science or equivalent

Learning Outcomes

By the end of the course, the student must be able to:

- Identify issues of experience design in relation to an actual design project

- Perform rigorous analysis of the problem space and map the design opportunities
- Develop alternative design concepts for future artifacts
- Translate design concepts into meaningful experiences through iterative prototyping at appropriate scales and levels of granularity
- Create convincing arguments for the design propositions and persuasive visual and tangible evidence

Teaching methods

Workshop, Design reviews, Presentations, Group projects

Expected student activities

Group discussion, Case studies, Design Reviews, Pin-Up, Desk Crits

Assessment methods

Grading will be based upon the quality of the projects in the preliminary workshops (30%), intermediary reviews (20%) and in the final review (50%). Projects will be reviewed and assessed based on their conceptual strength, the coherence of their translation into prototypes, their narrative clarity and experiential power, and the persuasiveness of their communication, both orally and through the presented artifacts.

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-489

EPFL

CS-550 **Formal verification**

Kuncak Viktor

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

We introduce formal verification as an approach for developing highly reliable systems. Formal verification finds proofs that computer systems work under all relevant scenarios. We will learn how to use formal verification tools and explain the theory and the practice behind them.

Content

Topics may include (among others) some of the following:

- Importance of Reliable Systems. Methodology of Formal Verification. Soundness and Completeness in Modeling and Tools. Successful Tools and Flagship Case Studies
- Review of Sets, Relations, Computability, Propositional and First-Order Logic Syntax, Semantics, Sequent Calculus.
- Completeness and Semi-Decidability for First-Order Logic. Inductive Definitions and Proof Trees. Higher-Order Logic and LCF Approach.
- State Machines. Transition Formulas. Traces. Strongest Postconditions and Weakest Preconditions.
- Hoare Logic. Inductive Invariants. Well-Founded Relations and Termination Measures
- Linear Temporal Logic. System Verilog Assertions. Monitors
- SAT Solvers and Bounded Model Checking
- Model Checking using Binary Decision Diagrams
- Loop Invariants. Hoare Logic. Statically Checked Function Contracts. Relational Semantics and Fixed-Point Semantics
- Symbolic Execution. Satisfiability Modulo Theories
- Abstract Interpretation
- Set theory for verification

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Discrete Mathematics (e.g. Kenneth Rosen: Discrete Mathematics and Its Applications)

Learning Outcomes

By the end of the course, the student must be able to:

- Formalize specifications
- Synthesize loop invariants
- Specify software functionality
- Generalize inductive hypothesis
- Critique current software development practices

Teaching methods

Instructors will present lectures and exercises and supervise labs on student laptops.

Expected student activities

Follow the course materials, take mid-term, and complete and explain projects during the semester.

Assessment methods

The grade is based on the written mid-term, as well as code, documentation, and explanation of projects during the semester. Specific percentages will be communicated in the first class.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- **Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511576430**
- **Aaron Bradley and Zohar Manna: The Calculus of Computation - Decision Procedures with Applications to Verification, Springer 2007.**
- Michael Huth and Mark Rayan: Logic in Computer Science - Modelling and Reasoning about Systems. Cambridge University Press 2004.
- Handbook of Model Checking, https://www.springer.com/de/book/9783319105741 Springer 2018. Including Chapter Model Checking Security Protocols by David Basin.
- Tobias Nipkow, Gerwin Klein: Concrete Semantics with Isabelle/HOL. http://concrete-semantics.org/concrete-semantics.pdf
- Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: Principles of Program Analysis. ISBN 978-3-662-03811-6. Springer 1999.
- Peter B. Andrews: An Introduction to Mathematical Logic and Type Theory (To Truth Through Proof), Springer 2002.
- http://logitext.mit.edu/tutorial

Ressources en bibliothèque

- Handbook of Practical Logic and Automated Reasoning / Harrison
- Handbook of model checking / Clarke
- [chapter] Model Checking Security Protocols / Bassin
- Principles of Program Analysis / Flemming
- The Calculus of Computation / Bradley
- Logic in Computer Science / Huth
- Introduction to mathematical logic and type theory / Andrews

Notes/Handbook

https://lara.epfl.ch/w/fv

Websites

- https://lara.epfl.ch/w/fv

Moodle Link

- https://go.epfl.ch/CS-550

EPFL

COM-406 **Foundations of Data Science**

Gastpar Michael C., Urbanke Rüdiger

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

We discuss a set of topics that are important for the understanding of modern data science but that are typically not taught in an introductory ML course. In particular we discuss fundamental ideas and techniques that come from probability, information theory as well as signal processing.

Content

This class presents basic concepts of Information Theory and Signal Processing and their relevance to emerging problems in Data Science and Machine Learning.
A tentative list of topics covered is:

1. Information Measures
2. Signal Representations
3. Detection and Estimation
4. Multi-arm Bandits
5. Distribution Estimation, Property Testing, and Property Estimation
6. Exponential Families
7. Compression and Dimensionality Reduction
8. Information Measures and Generalization Error

Keywords

Information Theory, Signal Processing, Statistical Signal Processing, Machine Learning, Data Science.

Learning Prerequisites

Required courses

COM-300 Modèles stochastiques pour les communications

Recommended courses

Statistics

Important concepts to start the course

Solid understanding of linear algebra and probability as well as real and complex analysis.

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the fundamental concepts of signal processing such as basis representations and sampling
- Formulate the fundamental concepts of information theory such as entropy and mutual information
- Analyze problems in statistical settings using fundamental bounds from information theory
- Formulate problems using robust and universal techniques

Teaching methods

Ex cathedra lectures, exercises, and small projects.

Expected student activities

Follow lectures; independent work on problems (homework and small projects).

Assessment methods

Written final exam during the exam session.
Homework Problem Sets during the semester.
10% homework, 30% midterm, 60% final exam; (if for some reason the course has to be given over zoom then we will skip the midterm and the course will be evaluated by 10% homework and 90% final)

Resources

Bibliography

Cover and Thomas, Elements of Information Theory (Second Edition), Wiley, 2006.

Ressources en bibliothèque

- Elements of Information Theory / Cover

Notes/Handbook

Lectures notes will be available on the course web page.

Moodle Link

- https://go.epfl.ch/COM-406

EPFL

CS-459 **Foundations of probabilistic proofs**

Chiesa Alessandro

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	4 weekly
Exercises	1 weekly
Number of positions	

Summary

Probabilistic proof systems (eg PCPs and IPs) have had a tremendous impact on theoretical computer science, as well as on real-world secure systems. They underlie delegation of computation protocols and hardness of approximation. This course covers the foundations of probabilistic proof systems.

Content

Proofs are at the foundations of mathematics, and verifying the correctness of a mathematical proof is a fundamental computational task. (The P versus NP problem, which deals precisely with the complexity of proof verification, is one of the most important open problems in all of mathematics.) The complexity-theoretic study of proof verification has led to new notions of mathematical proofs, such as Interactive Proofs, Probabilistically Checkable Proofs, and others.
Probabilistic proofs are a powerful tool for proving hardness of approximation results, and are an essential building block to achieve delegation of computation (protocols that enable super fast verification of long computations, such as SNARKs). Via these applications, probabilistic proofs have had a tremendous impact on theoretical computer science and, more recently, are playing an exciting role in applied cryptography, computer security, and blockchain technology (e.g., they help secure billions of dollars in transactionds per day).
This course provides an introduction to probabilistic proofs and the beautiful mathematics underlying them.
Covered topics include arithmetization, the sumcheck protocol, zero knowledge, doubly-efficient interactive proofs, linearity testing, low-degree testing, proof composition, succinct verification, and more.
This courses assumes basic familiarity with algorithms (asymptotic notation and analysis of algorithms), complexity theory (computation models and simple complexity classes), and some algebra (finite fields and their properties).

Learning Prerequisites

Important concepts to start the course

- Basic knowledge of discrete probability.
- Basic knowledge of finite fields and their properties.
- Basic knowledge of algorithms (asymptotic notation and analysis of algorithms).
- Basic knowledge of computational complexity (Turing machines; boolean circuits; complexity classes; reductions, familiarity with the classes P and NP; probabilistic computation and the class BPP).

Teaching methods

Two weekly lectures that include definitions, theorems, and proofs. One weekly recitation to guide students through exercises. Weekly problem sets to reinforce the material.

Resources

Moodle Link

- https://go.epfl.ch/CS-459

CS-452

Foundations of software

Bourgeat Thomas

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

The course introduces the foundations on which programs and programming languages are built. It introduces syntax, types and semantics as building blocks that together define the properties of a program part or a language. Students will learn how to apply these concepts in their reasoning.

Content

- simple types, lambda-calculus
- normalization, references, exceptions
- recursive types
- polymorphism
- proposition as types
- Tentatively: combinatory logic, calculus of inductive constructions

Keywords

lambda-calculus, type systems

Learning Prerequisites

Recommended courses

Advanced topics in programming, Compiler construction

Important concepts to start the course

Functional programming
Basic knowledge of formal languages

Learning Outcomes

By the end of the course, the student must be able to:

- Argue design decisions of programming languages
- Assess / Evaluate soundness of type systems
- Design programming language semantics and typing rules
- Verify progress and preservation in type systems
- Work out / Determine operational equivalences
- Carry out projects of 2-3 weeks duration

- Distinguish valid from invalid proofs
- Implement type systems and operational semantics

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Manage priorities.

Teaching methods

Ex cathedra, practical exercises

Assessment methods

Final written exam during the exam session (60%)
Projects and homeworks by groups during the semester (40%)

Resources

Moodle Link

- https://go.epfl.ch/CS-452

EPFL

Gödel and recursivity

MATH-483

Duparc Jacques

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Cours donné en alternance tous les deux ans

Summary

Gödel incompleteness theorems and mathematical foundations of computer science

Content

Recursivity :
Turing Machines and variants. The Church-Turing Thesis. Universal Turing Machine. Undecidable problems (the halting and the Post-Correspondance problems). Reducibility. The arithmetical hierarchy. Relations to Turing machines. Turing degrees.
Gödel's theorems:
Peano and Robinson Arithmetics. Representable functions. Incompleteness, and undecidability theorems in arithmetics.

If time permits, the completeness of the first order theory of the field of real numbers.

Keywords

Gödel, incompleteness theorems, Peano arithmetic, Robinson arithmetic, decidability, recursively enumarable, arithmetical hierarchy, Turing machine, Turing degrees, jump operator, primitive recursive functions, recursive functions, automata, pushdown automata, regular languages, context-free languages, recursive languages, halting problem, universal Turing machine, Church thesis.

Learning Prerequisites

Recommended courses

Mathematical logic (or equivalent). A good understanding of 1st order logic is required - in particular the relation between syntax and semantics.

Important concepts to start the course

1st order logic: syntax, semantics, proof theory, completeness theorem, compactness theorem, Löwenheim-Skolem theorem.

Learning Outcomes

By the end of the course, the student must be able to:

- Estimate whether a given theory, function, language is recursive or no
- Decide the class that a language belongs to (regular, context-free, recursive,...)
- Elaborate an automaton
- Design a Turing machine
- Formalize a proof in Peano arithmetic
- Sketch the incompleteness theorems
- Propose a non-standard model
- Argue why Hilbert program failed

Teaching methods

Ex cathedra lecture and exercises

Assessment methods

Written: 3 hours
Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Set Theory:

- Thomas Jech: Set theory, Springer 2006
- Kenneth Kunen: Set theory, Springer, 1983
- Jean-Louis Krivine: Theory des ensembles, 2007
- Patrick Dehornoy: Logique et théorie des ensembles; Notes de cours, FIMFA ENS: http://www.math.unicaen.fr/~dehornoy/surveys.html
- Yiannis Moschovakis: Notes on set theory, Springer 2006
- Karel Hrbacek and Thomas Jech: Introduction to Set theory, (3d edition), 1999

Recursion Theory :

- Micheal Sipser: Introduction to the Theory of Computation, Thomson Course Technology Boston, 2006
- Piergiorgio Odifreddi: Classical recursion theory, vol. 1 and 2, Springer, 1999
- Robert I. Soare: Recursively Enumerable Sets and Degres, A Study of Computable Functions and Computably Generated Sets, Springer-Verlag 1987
- Nigel Cutland: Computability, an introduction to recursive function theory, 1980
- Raymond M. Smullyan: recursion theory for methamathematics, Oxford, 1993

Proof theory :

- Wolfram Pohlers: Proof Theory, the first step into impredicativity, Springer, 2008
- A. S. Troelstra, H. Schwichtenberg, and Anne S. Troelstra: Basic proof theory, Cambridge, 2000

• S.R. Buss: Handbook of proof theory, Springer, 1998

Gödel's results :
• Raymond M. Smullyan: Gödel's incompleteness theorems, Oxford, 1992
• Peter Smith: An introduction to Gödel's theorems, Cambridge, 2008
• Torkel Franzen: Inexhaustibility, a non exhaustive treatment, AK Peteres, 2002
• Melvin Fitting: Incompleteness in the land of sets, King's College, 2007
• Torkel Franzen: Gödel's theorem: an incomplete guide to its use and abuse, AK Peters, 2005

Ressources en bibliothèque

- Théorie des ensembles / Krivine
- Inexhaustibility, a non exhaustive treatment / Franzen
- Proof Theory / Pohlers
- Notes on theory / Moschovakis
- Basic proof theory / Troelstra
- Introduction to the Theory of Computation / Sipser
- Handbook of proof theory / Buss
- Set theory / Jech
- Classical recursion theory / Odifreddi
- Recursion theory for methamathematics / Smullyan
- Set theory / Kunen
- Incompleteness in the land of sets / Fitting
- Recursively Enumerable Sets and Degres / Soare
- Gödel's theorem / Franzen
- Computability, an introduction to recursive function theory / Cutland
- Logique et théorie des ensembles / Dehornoy
- Gödel's incompleteness theorems / Smullyan
- An introduction to Gödel's theorems / Smith
- Introduction to Set theory / Hrbacek

Websites

- http://www.hec.unil.ch/logique/enseignement/recursivity

Moodle Link

- https://go.epfl.ch/MATH-483

EPFL

MICRO-511 **Image processing I**

Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Minor in Imaging	H	Opt.
Minor in life sciences engineering	H	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Nuclear engineering	MA1	Opt.
Photonics minor	H	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly
Number of positions	

Summary

Introduction to the basic techniques of image processing. Introduction to the development of image-processing software and to prototyping using Jupyter notebooks. Application to real-world examples in industrial vision and biomedical imaging.

Content

- **Introduction**. Image processing versus image analysis. Applications. System components.
- **Characterization of continuous images**. Image classes. 2D Fourier transform. Shift-invariant systems.
- **Image acquisition**. Sampling theory. Acquisition systems. Histogram and simple statistics. Max-Lloyd quantization (K-means).
- **Characterization of discrete images and linear filtering**. z-transform. Convolution. Separability. FIR and IIR filters.
- **Morphological operators**. Binary morphology (opening, closing, etc.). Gray-level morphology.
- **Image-processing tasks**. Preprocessing. Matching and detection. Feature extraction. Segmentation.
- **Convolutional neural networks**. Basic components. Operator-based formalism. CNN in practice: denoising and segmentation.

Learning Prerequisites

Required courses

Signals and Systems I & II (or equivalent)

Important concepts to start the course

1-D signal processing: convolution, Fourier transform, z-transform

Learning Outcomes

By the end of the course, the student must be able to:

- Exploit the multidimensional Fourier transform
- Select appropriately Hilbert spaces and inner-products
- Optimize 2-D sampling to avoid aliasing
- Formalize convolution and optical systems
- Design digital filters in 2-D
- Analyze multidimensional linear shift-invariant systems
- Apply image-analysis techniques
- Construct image-processing software
- Elaborate morphological filters
- Exploit the multidimensional Fourier transform
- Select appropriately Hilbert spaces and inner-products
- Optimize 2-D sampling to avoid aliasing
- Formalize convolution and optical systems
- Design digital filters in 2-D
- Analyze multidimensional linear shift-invariant systems
- Apply image-analysis techniques
- Construct image-processing software

Transversal skills

- Use a work methodology appropriate to the task.
- Manage priorities.
- Use both general and domain specific IT resources and tools

Assessment methods

- 70% final exam
- 30% IP labs during semester

Resources

Moodle Link

- https://go.epfl.ch/MICRO-511

EPFL

MICRO-512 **Image processing II**

Liebling Michael, Sage Daniel, Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Environmental Sciences and Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Photonics minor	E	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly
Number of positions	

Summary

Study of advanced image processing; mathematical imaging. Development of image-processing software and prototyping in Jupyter Notebooks; application to real-world examples in industrial vision and biomedical imaging.

Content

- **Directional image analysis.** Mathematical foundations. Structure tensor. Steerable filters.
- **Continuous representation of discrete data.** Splines. Interpolation. Geometric transformations. Multi-scale decomposition (pyramids and wavelets).
- **Image transforms.** Karhunen-Loève transform (KLT). Discrete cosine transform (DCT). JPEG coding. Image pyramids. Wavelet decomposition.
- **Reconstruction in the continuum.** Wiener filter. Radon transform. Fourier slice theorem. Filtered backprojection.
- **Computational imaging.** Imaging as an inverse problem. Iterative reconstruction methods. Elements of convex analysis. Regularization & sparsity constraints.

Learning Prerequisites

Required courses

Image Processing I
Recommended courses
Signals and Systems I & II, linear algebra, analysis
Important concepts to start the course
Basic image processing and related analytical tools (Fourier transform, z-tranform, etc.)

Recommended courses

Signals and Systems I & II, linear algebra, analysis

Important concepts to start the course

Basic image processing and related analytical tools (Fourier transform, z-tranform, etc.)

Learning Outcomes

- Construct interpolation models and continuous-discrete representations
- Analyze image transforms
- Design image-reconstruction algorithms
- Formalize multiresolution representations using wavelets
- Design deconvolution algorithms
- Perform image analysis and feature extraction
- Design image-processing software (plugins)
- Synthesize steerable filters
- Construct interpolation models and continuous-discrete representations
- Analyze image transforms
- Design image-reconstruction algorithms
- Formalize multiresolution representations using wavelets
- Perform image analysis and feature extraction
- Design image-processing software
- Design image reconstruction algorithms

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Manage priorities.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Assessment methods

The objectives of the course will be assessed as follows:

- 70% final exam
- 30% IP labs

Resources

Moodle Link

- https://go.epfl.ch/MICRO-512

EPFL

CS-487

Industrial automation

Sommer Philipp Alexander, Tournier Jean-Charles

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Energy Science and Technology	MA2, MA4	Opt.
Mechanical engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	**36**

It is not allowed to withdraw from this subject after the registration deadline.

Remark

This course can be taken by students of all engineering sections.

Summary

This course consists of two parts: 1) architecture of automation systems, hands-on lab 2) dependable systems and handling of faults and failures in real-time systems, including fault-tolerant computing

Content

Trends like digitalization and internet of things affect the way industrial plants are designed, deployed and operated. Industrial Automation comprises the control, communication and software architecture in (real-time) automation systems: factories, energy production and distribution, vehicles and other embedded systems.

Keywords

1. Processes and plants, automation system architecture
2. Instrumentation, Programmable Logic Controllers and embedded computers
3. Industrial communication networks, field busses
4. Field device access protocols and application program interfaces
5. Human interface and supervision
6. Dependability (Reliability, Availability, Safety, ...)

Learning Prerequisites

Recommended courses

Communication networks

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the (software) architecture of a automation system
- Apply methods and trade-offs in real-time systems
- Analyze a plant
- Propose suitable automation solutions meeting the requirements

- Analyze the reliability, availability, safety of a system

Transversal skills

- Write a scientific or technical report.
- Use both general and domain specific IT resources and tools
- Communicate effectively with professionals from other disciplines.
- Keep appropriate documentation for group meetings.
- Access and evaluate appropriate sources of information.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.

Teaching methods

Oral presentation aided by slides, exercises as part of the lecture, practical work (workshop at Siemens and group assignment).

Expected student activities

- Understand material presented during lectures by asking questions and/or independent (online) searches
- Attend Siemens workshop (one full day on Siemens premises in Renens)
- Work on group assignment
- Hand-in artifacts for assignment on time

Assessment methods

Assignment 25% and final oral exam 75%

Resources

Bibliography

Introduction to Industrial Automation, Stamatios Manesis & George Nikolakopoulos, CRC Press, 2018

Ressources en bibliothèque

- Introduction to Industrial Automation / Manesis

Moodle Link

- https://go.epfl.ch/CS-487

COM-402

Information security and privacy

Payer Mathias

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This course provides an overview of information security and privacy topics. It introduces students to the knowledge and tools they will need to deal with the security/privacy challenges they are likely to encounter in today's world. The tools are illustrated with relevant applications.

Content

- Overview of cyberthreats
- Basic exploitation of vulnerabilities
- Authentication, access control, compartmentalization
- Basic applied cryptography
- Operational security practices and failures
- Machine learning and privacy
- Data anonymization and de-anonymization techniques
- Privacy enhancing technologies
- Blockchain and decentralization

Keywords

security, privacy, protection, intrusion, anonymization, cryptography

Learning Prerequisites

Required courses

COM-301 Computer security
Basic systems programming (in C/C++) or better
Basic networking knowledge
Good scripting knowledge (Python)

Learning Outcomes

By the end of the course, the student must be able to:

- Understand the most important classes of information security/privacy risks in today's "Big Data" environment
- Exercise a basic, critical set of "best practices" for handling sensitive information
- Exercise competent operational security practices in their home and professional lives
- Understand at overview level the key technical tools available for security/privacy protection
- Understand the key technical tools available for security/privacy protection
- Exercise competent operational security practices

Expected student activities

Attending lectures, solving assigned problems and "hands-on" exercises, reading and demonstrating understanding of provided materials.

Assessment methods

- continuous control : 30% of the grade
- final exam : 70% of the grade

Resources

Moodle Link

- https://go.epfl.ch/COM-402

EPFL

COM-404

Information theory and coding

Telatar Emre

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

The mathematical principles of communication that govern the compression and transmission of data and the design of efficient methods of doing so.

Content

1. Mathematical definition of information and the study of its properties.
2. Source coding: efficient representation of message sources.
3. Communication channels and their capacity.
4. Coding for reliable communication over noisy channels.
5. Multi-user communications: multi access and broadcast channels.
6. Lossy source coding : approximate representation of message sources.
7. Information Theory and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the fundamenal concepts of information theory such as entropy, mutual information, channel capacity
- Elaborate the principles of source coding and data transmission
- Analyze source codes and channel codes
- Apply information theoretic methods to novel settings

Teaching methods

Ex cathedra + exercises

Assessment methods

With continuous control

Resources

Websites

- http://moodle.epfl.ch/enrol/index.php?id=14593

Moodle Link

- https://go.epfl.ch/COM-404

EPFL

CS-430

Intelligent agents

Faltings Boi

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	3 weekly
Number of positions	

Summary

Software agents are widely used to control physical, economic and financial processes. The course presents practical methods for implementing software agents and multi-agent systems, supported by programming exercises, and the theoretical underpinnings including computational game theory.

Content

The course contains 4 main subject areas:

1) Basic models and algorithms for individual agents:
Models and algorithms for rational, goal-oriented behavior in agents: reactive agents, reinforcement learning, exploration-exploitation tradeoff, AI planning methods.
2) Multi-agent systems:
multi-agent planning, coordination techniques for multi-agent systems, distributed algorithms for constraint satisfaction.
3) Self-interested agents:
Models and algorithms for implementing self-interested agents motivated by economic principles: elements of computational game theory, models and algorithms for automated negotiation, social choice, mechanism design, electronic auctions and marketplaces.
4) Implementing multi-agent systems:
Agent platforms, ontologies and markup languages, web services and standards for their definition and indexing.

Learning Prerequisites

Recommended courses

Intelligence Artificielle or another introductory course to AI

Learning Outcomes

By the end of the course, the student must be able to:

- Choose and implement methods for rational decision making in software agents, based on decision processes and AI planning techniques
- Choose and implement methods for efficient rational decision making in teams of multiple software agents
- Model scenarios with multiple self-interested agents in the language of game theory
- Evaluate the feasibility of achieving goals with self-interested agents using game theory

- Design, choose and implement mechanisms for self-interested agents using game theory
- Implement systems of software agents using agent platforms

Teaching methods

Ex cathedra, practical programming exercises

Expected student activities

Lectures: 3 hours
Reading: 3 hours
Assignments/programming: 4 hours

Assessment methods

Midterm and quizzes 30%, final exam 70%

Resources

Bibliography

Michael Wooldridge : An Introduction to MultiAgent Systems - Second Edition, John Wiley & Sons, 2009
Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd/3rd/4th Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009/2015.

Ressources en bibliothèque

- An Introduction to MultiAgent Systems / Wooldridge
- Artificial Intelligence: A Modern Approach / Russell

Moodle Link

- https://go.epfl.ch/CS-430

CS-486

Interaction design

Pu Pearl

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course focuses on goal-directed design and interaction design, two subjects treated in depth in the Cooper book (see reference below). To practice these two methods, we propose a design challenge, which is further divided into mini-projects evenly spaced throughout the semester.

Content

Design methods for HCI
What is HCI: its aims and goals
Design thinking
Goal-directed Design
Mental model and different types of users
Qualitative research and user interviews
User modeling: persona and empathy diagram
Scenarios, requirements and framework design
Visual design
Information Visualization design
Basic prototyping methods for HCI
Storyboarding
Context scenario
Interactive prototype
Video prototype
Human computer interaction evaluation methods
Cognitive walkthrough
Heuristic evaluation
Evaluation with users

Keywords

Interaction design, design thinking, user interviews, ideation, storyboard, context scenarios, digital mockup, user evaluation, video prototyping.

Learning Prerequisites

Required courses

Interaction personne-système

Recommended courses

Open to students enrolled in the Master and PhD programs in IC.

Important concepts to start the course

Goal-directed design, design thinking, user needs assessment, user interviews & observation, ideation, prototyping, evaluation

Learning Outcomes

By the end of the course, the student must be able to:

- Interview users and elicit their needs using the goal-directed design method
- Design and implement interfaces and interactions
- Project management : set objectives and device a plan to achieve them
- Group work skills : discuss and identify roles, and assume those roles including leadership
- Communication : writing and presentation skills

Teaching methods

Lectures, flipped classroom lectures, exercises, hands-on practice, case studies

Expected student activities

Participation in lectures, exercises, user interviews, ideation sessions, readings, design project, project presentation

Assessment methods

The assessments consist of five in-class open-book exercises, each lasting one hour. Three of these exercises will be randomly selected for grading. Additionally, there will be two mini-projects that will be graded based on group performance. Furthermore, students' individual engagement in group activities, including user interviews, ideation, prototyping, and peer evaluation, will also be evaluated to determine individual performance.
30% open-book exercises (done in class, open notes, open book) - individual performance
20% individual engagement in group activities such as user interviews - individual performance
50% project - group performance

Resources

Bibliography

About Face 3: The Essentials of Interaction Design by Alan Cooper et al. (available as e-book at NEBIS)

Ressources en bibliothèque

- About Face 3 / Cooper

Moodle Link

- https://go.epfl.ch/CS-486

Videos

- https://mediaspace.epfl.ch/channel/CS-486%2BInteraction%2BDesign/29793?&&

EPFL

CS-428

Interactive theorem proving

Barrière Aurèle, Pit-Claudel Clément

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

A hands-on introduction to interactive theorem proving, computer-checked mathematics, compiler verification, proofs as programs, dependent types, and proof automation. Come learn how to write computer-checked proofs and certified bug-free code!

Content

- Intro to the Coq proof assistant (logic, higher-order functions, tactics)
- Functional programming (inductive types and fixpoints)
- Structural induction (data structures and verified algorithms)
- Interpreter-based program semantics (intro to compiler verification)
- Inductive relations (predicates, rule induction)
- Automation and tactics I (bottom-up reasoning and logic programming)
- Operational program semantics (small-and big-step semantics)
- Program logics (hoare triples)
- Automation and tactics II (top-down reasoning)
- Type systems (Simply-typed lambda calculus)
- Dependent types and equality proofs
- Automation and tactics III (proofs by reflection)
- Real-world theorem proving (various topics)

Learning Prerequisites

Recommended courses

This course assumes no knowledge of programming language theory. The following courses may be useful, but are not required:

- CS-320 Computer language processing (to introduce the concept of interpreter)
- CS-425 Foundations of software (to introduce type systems and the lambda calculus)
- CS-550 Formal verification (for a different perspective on theorem proving)

Important concepts to start the course

- Functional programming

Learning Outcomes

- Implement purely-functional algorithms in the Gallina language
- Translate informal requirements about software into precise mathematical properties
- Plan and carry out mechanized proofs in Coq (e.g. maths, algorithms, compilers, type systems)
- Automate repetitive proof tasks by crafting simple custom decision procedures

Teaching methods

- Lectures
- Live-coding sessions

Expected student activities

- Lectures
- Programming and verification assignments
- Project (proposal, check-in, presentation, report)

Assessment methods

- Take-home programming and verification assignments: 40% of the final grade (3 or 4 labs)
- Formal verification project: 60% of the final grade (~10 weeks, in teams of 1 to 4 students)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-511

EPFL

CS-594 **Internship credited with Master Project (master in Cybersecurity)**

Profs divers *

Cursus	Sem.	Type
Cybersecurity	MA1, MA2, MA3, MA4, PME, PMH	Obl.

Language of teaching	English
Credits	0
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	0h
Weeks	
Project	320 weekly
Number of positions	

Summary

The engineering internship is an integral part of the curriculum for master's students. They join companies in Switzerland or abroad to carry out an internship or their master's project in a field of activity where the skills of the futur Engineer are highlighted.

Content

Internships represent an important experience for students, allowing them to achieve the following goals :
- Immerse themselves in the professional world
- Highlight the importance of teamwork
- Consider the imperatives of a companyin its processes
- Put into practice the knowledge acquired from the study plan

The following three forms of internship are possible as part of the master's study plan :
- 8 weeks internship during the summer only
- 6 months (long) internship. The student is on leave for one semester
- Master project in the industry (25 weeks)

Learning Prerequisites

Required courses

- Have completed one full semester for students who have obtained their Bachelor EPFL
- Have completed two full semesters for students coming from another university

Learning Outcomes

By the end of the course, the student must be able to:

- Be aware of the importance of legal procedures and the ethical code of the profession
- Communicate effectively and be understood
- Self-assess the level of acquired skills and plan the next goals
- Manage the priorities
- Receive and give feedback (criticism) and respond appropriately

Expected student activities

The student agrees to do this internship with the professionalism

Assessment methods

- Short internship (8 weeks) : electronic evaluation at the end of the internship
- Long internship (6 months) : electronic evaluation at the end of the internship
- Master's project in the industry : see the master's project course book

Supervision

Others	Industry supervisor EPFL supervisor

Resources

Websites

- https://www.epfl.ch/schools/ic/internships/
- https://www.epfl.ch/about/recruiting/recruiting/internships/

EPFL

CS-491 Introduction to IT consulting

Regev Gil

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	6 weekly
Number of positions	

Summary

This course is an introduction to the alignment of enterprise needs with the possibilities offered by Information Technology (IT). Using a simulated business case, we explore how to define the requirements for an IT service that matches stakeholders explicit and implicit wishes.

Content

Target Audience

Engineers who want to become
- Business Analysts
- Requirements Engineers
- Project Managers
- Management and IT consultants
- Product Owners

Content

Technological and societal changes are pressuring enterprise IT departments to hire engineers with excellent technical andbusiness skills. Their roles are called business analysts, requirements engineers, or product owners. Their skills enable the bidirectional alignment of business needs and IT capabilities. With IT being the most important enabler of enterprise strategy, these roles are crucial in many organizations, large and small, private or public.

We use experiential learning beginning with concrete experience, followed by reflection and abstraction to encourage collaborative learning by doing. You will be part of a small team that needs to understand and solve a business case through fast-paced role-playing with the teaching staff. This is interspersed with lectures on the nature of organizations, business analysis and the role of enterprise IT. Several external speakers from industry illustrate what we see in class.

We will explore the following subjects:

- Problems and solutions
- Requirements elicitation
- Business process modeling
- Project management
- Change management
- Enterprise and service modeling
- The nature of organizations
- Creating a request for tender

Keywords

Appreciation, business analysis, business process, business service, contextual inquiry,
ethnography, homeostasis, interviews, IT service, motivation modeling, request for tender, requirements engineeing,

resilience, service modeling

Learning Outcomes

By the end of the course, the student must be able to:

- Elicit requirements with business stakeholders
- Analyze business stakeholder perception and motivations
- Assess / Evaluate business processes
- Define requirements for business and IT services
- Present problems and solutions to management

Transversal skills

- Demonstrate a capacity for creativity.
- Communicate effectively with professionals from other disciplines.
- Take feedback (critique) and respond in an appropriate manner.

Teaching methods

Experimental learning and teamwork.

Assessment methods

Group oral exam.

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Beyer, H. and K. Holtzblatt (1999). "Contextual design." interactions 6(1): 32-42.
Markus M.L., Keil M. (1994). If We Build It, They Will Come: Designing Information Systems that People Want to use, Sloan Management Review; Summer 1994; 35, 4; ABI/INFORM Global pg. 11
Regev, G. et al.(2013) What We Can Learn about Business Modeling from Homeostasis, Lecture Notes in Business Information Processing, 142, 1-15, 2003
Zachman, J. A. (1987). "A framework for information systems architecture." IBM Syst. J. 26 (3): 276-292.
Weinberg, G.M., The secrets of consulting, Dorset House, 1985

Ressources en bibliothèque

- A framework for information systems architecture / Zachman
- What We Can Learn about Business Modeling from Homeostasis / Regev
- Contextual design / Holtzblatt

Moodle Link

- https://go.epfl.ch/CS-491

EPFL

CS-431

Introduction to natural language processing

Bosselut Antoine, Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The objective of this course is to present the main models, formalisms and algorithms necessary for the development of applications in the field of natural language information processing. The concepts introduced during the lectures will be applied during practical sessions.

Content

Several models and algorithms for automated textual data processing will be described: morpho-lexical level: n-gram and language models, spell checkers, ...; semantic level: models and formalisms for the representation of meaning, embedings, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computationnal Linguisitics; Part-of-Speech tagging

Learning Outcomes

By the end of the course, the student must be able to:

- Compose key NLP elements to develop higher level processing chains
- Assess / Evaluate NLP based systems
- Choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, ...)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independant components

Teaching methods

Flipped classroom (reviews and supervised "hands-on" in class) ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 16%, final exam 84%

Resources

Bibliography

1. M. Rajman editor, "*Speech and Language Engineering*", EPFL Press, 2006.
2. Daniel Jurafsky and James H, Martin, "*Speech and Language Processing"*, Prentice Hall, 2008 (2nd edition)
3. Christopher D. Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 2000
4. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008
5. Nitin Indurkhya and Fred J. Damerau editors, "*Handbook of Natural Language Processing*", CRC Press, 2010 (2nd edition)

Ressources en bibliothèque

- Handbook of Natural Language Processing / Indurkhya
- Introduction to Information Retrieval / Manning
- Foundations of Statistical Natural Language Processing / Manning
- Speech and Language Engineering / Rajman
- Speech and Language Processing / Jurafsky

Websites

- https://coling.epfl.ch/

Moodle Link

- https://go.epfl.ch/CS-431

EPFL

CS-479

Learning in neural networks

Gerstner Wulfram

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

Artificial Neural Networks are inspired by Biological Neural Networks. One big difference is that optimization in Deep Learning is done with the BackProp Algorithm, whereas in biological neural networks it is not. We show what biologically plausible learning algorithms can do (and what not).

Content

- Why BackProp is biologically not plausible. Biological two-factor rules and neuromorphic hardware
- Hebbian Learning (two-factor rules) for PCA and ICA
- Two-factor rules for dictionary learning (k-means/competitive learning/winner-takes-all)
- Three-factor rules and neuromodulators (theory and neuroscience)
- Three-factor rules for reward-based learning (theory)
- Three-factor rules for TD reinforcement-learning (algorithmic formulations)
- Actor-critic networks
- Reinforcement learning in the brain
- Learning by surprise and novelty: exploration and changing environments (algorithmic)
- Surprise and novelty in the brain
- Learning representations in multi-layer networks (algorithms without backprop)
- Learning to find a goal: a bio-plausible model with place cells and rewards
- Neuromorphic hardware and in-memory computing

Keywords

- Hebbian learning and two-factor rules
- distributed local algorithms,
- Principal Component Analysis/Independent Component Analysis (PCA and ICA)
- Reinforcement Learning (RL)
- surprise and novelty
- three-factor rules
- neuromorphic hardware

Learning Prerequisites

Required courses

Linear Algebra AND Analysis.
Machine learning

Recommended courses

Signal processing

Important concepts to start the course

Optimization, Gradient Descent, Filtering, Loss function, Eigenvalues,

Learning Outcomes

By the end of the course, the student must be able to:

- Translate concepts from machine learning and signal processing into bio-plausible algorithms
- Translate neuroscience of learning into algorithms
- Explain differences between and similarities of various algorithms
- Discriminate imitations and advantages of various learning algorithms for implementation in biology or hardware

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Set objectives and design an action plan to reach those objectives.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Give feedback (critique) in an appropriate fashion.
- Manage priorities.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Ex cathedra, Exercises, and Miniproject

Expected student activities

Participation in Class, Solution of Exercises, Miniproject.

Assessment methods

Oral exam (70 percent) plus miniproject (30 percent). If more than 45 students participate, the oral exam is replaced by a written exam.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-479

CS-526

Learning theory

Macris Nicolas

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Machine learning and data analysis are becoming increasingly central in many sciences and applications. This course concentrates on the theoretical underpinnings of machine learning.

Content

- Basics : statistical learning framework, Probably Approximately Correct (PAC) learning, learning with a finite number of classes, Vapnik-Chervonenkis (VC).
- Bias-variance tradeoff and modern double descent phenomena.
- Neural Nets : representation power of neural nets.
- Stochastic gradient descent, modern aspects: mean field approach, neural tangent kernel.
- Matrix factorization, Tensor decompositions and factorization, Jenrich's tehorem, Alternating least squares, Tucker decompositions.
- Applications: e.g. Learning mixture models, topic modeling.

Learning Prerequisites

Recommended courses

- Analysis I, II, III
- Linear Algebra
- Machine learning
- Probability
- Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the framework of PAC learning
- Explain the importance basic concepts such as VC dimension, bias-variance tradeoff and double descent
- Describe basic facts about representation of functions by neural networks
- Describe recent results on specific topics e.g., matrix and tensor factorization, learning mixture models

Teaching methods

- Lectures
- Exercises

Expected student activities

- Attend lectures
- Attend exercises sessions and do the homework

Assessment methods

Final exam and graded homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Course website

EPFL

CS-433

Machine learning

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

Machine learning methods are becoming increasingly central in many sciences and applications. In this course, fundamental principles and methods of machine learning will be introduced, analyzed and practically implemented.

Content

1. *Basic regression and classification concepts and methods: Linear models, overfitting, linear regression, Ridge regression, logistic regression, k-NN, SVMs and kernel methods*
2. *Fundamental concepts: cost-functions and optimization, cross-validation and bias-variance trade-off, curse of dimensionality.*
3. *Neural Networks: Representation power, backpropagation, activation functions, CNN, regularization, data augmentation, dropout*
4. *Unsupervised learning: k-means clustering, gaussian mixture models and the EM algorithm. Basics of self-supervised learning*
5. *Dimensionality reduction: PCA and matrix factorization, word embeddings*
6. *Advanced methods: Adversarial learning, Generative adversarial networks*

Keywords

- *Machine learning, pattern recognition, deep learning, neural networks, data mining, knowledge discovery, algorithms*

Learning Prerequisites

Required courses

• Analysis I, II, III
• Linear Algebra
• Probability and Statistics (MATH-232)
• Algorithms I (CS-250)

Recommended courses

• *Introduction to machine learning (CS-233)*
• *...or similar bachelor lecture from other sections*

Important concepts to start the course

• *Basic probability and statistics (conditional and joint distribution, independence, Bayes rule, random variables, expectation, mean, median, mode, central limit theorem)*
• *Basic linear algebra (matrix/vector multiplications, systems of linear equations, SVD)*
• *Multivariate calculus (derivative w.r.t. vector and matrix variables)*
• *Basic Programming Skills (labs will use Python)*

Learning Outcomes

By the end of the course, the student must be able to:

• Define the following basic machine learning problems: Regression, classification, clustering, dimensionality reduction, time-series
• Explain the main differences between them
• Implement algorithms for these machine learning models
• Optimize the main trade-offs such as overfitting, and computational cost vs accuracy
• Implement machine learning methods to real-world problems, and rigorously evaluate their performance using cross-validation. Experience common pitfalls and how to overcome them
• Explain and understand the fundamental theory presented for ML methods
• Conduct a real-world interdisciplinary machine learning project, in collaboration with application domain experts
• Define the following basic machine learning models: Regression, classification, clustering, dimensionality reduction, neural networks, time-series analysis

Teaching methods

• Lectures
• Lab sessions
• Course Projects

Expected student activities

Students are expected to:

- attend lectures
- attend lab sessions and work on the weekly theory and coding exercises
- work on projects using the code developed during labs, in small groups

Assessment methods

- Written final exam
- Continuous control (Course projects)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Christopher Bishop, Pattern Recognition and Machine Learning
- Kevin Murphy, Machine Learning: A Probabilistic Perspective
- Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning
- Michael Nielsen, Neural Networks and Deep Learning
- (Jerome Friedman, Robert Tibshirani, Trevor Hastie, The elements of statistical learning : data mining, inference, and prediction)

Ressources en bibliothèque

- The elements of statistical learning : data mining, inference, and prediction / Friedman
- Pattern Recognition and Machine Learning / Bishop
- Understanding Machine Learning / Shalev-Shwartz
- Machine Learning / Murphy

Références suggérées par la bibliothèque

- Neural Networks and Deep Learning / Nielsen

Notes/Handbook

https://github.com/epfml/ML_course

Websites

- https://www.epfl.ch/labs/mlo/machine-learning-cs-433/

Moodle Link

- https://go.epfl.ch/CS-433

CS-421

Machine learning for behavioral data

Käser Tanja

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Remark

Cours biennal

Summary

Computer environments such as educational games, interactive simulations, and web services provide large amounts of data, which can be analyzed and serve as a basis for adaptation. This course will cover the core methods of user modeling and personalization, with a focus on educational data.

Content

The users of computer environments such as intelligent tutoring systems, interactive games, and web services are often very heterogeneous and therefore it is important to adapt to their specific needs and preferences.
This course will cover the core methods of adaptation and personalization, with a focus on educational data. Specifically we will discuss approaches to the task of accurately modeling and predicting human behavior within a computer environment. Furthermore, we we will also discuss data mining techniques with the goal to gain insights into human behavior. We will cover the theories and methodologies underlying the current approaches and then also look into the most recent developments in the field.

1. Cycle of adaptation : representation, prediction, intervention (e.g. recommendation)
2. Data Processing and Interpretation (missing data, feature transformations, distribution fitting)
3. Performance evaluation (cross-validation, error measures, statistical significance, overfitting)
4. Representation & Prediction (probabilistic graphical models, recurrent neural networks, logistic models, clustering-classification approaches)
5. Recommendation (collaborative filtering, content-based recommendations, multi-armed bandits)
6. Stealth Assessment (seemless detection of user traits)
7. Multimodal analytics (represent & analyze data from non-traditional sources. i.e. sensors, classroom analytics, human-robot interaction)

Learning Prerequisites

Required courses

The student must have passed a course in probability and statistics and a course including a programming project

Recommended courses

- CS-433 Machine learning or
- CS-233 Introduction to machine learning

Important concepts to start the course

Probability and statistics, basic machine learning knowledge, algorithms and programming, Python

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the main machine learning approaches to personalization, describe their advantages and disavantages and explain thedifferences between them.
- Implement algorithms for these machine learning models
- Apply them to real-world data
- Assess / Evaluate their performance
- Explain and understand the fundamental theory underlying the presented machine learning models

Teaching methods

- Lectures
- Weekly lab sessions
- Course project

Expected student activities

- Attend the lectures
- Attend the lab sessions and work on the homework assignments
- Project work

Assessment methods

- Project work (50%)
- Final exam (50%)

Resources

Moodle Link

- https://go.epfl.ch/CS-421

EPFL

MGT-427 Management de projet et analyse du risque

Wieser Philippe

Cursus	Sem.	Type
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Génie civil	MA1, MA3	Obl.
Informatique	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Microtechnique	MA1, MA3	Opt.
Mineur en Management, technologie et entrepreneuriat	H	Opt.
Robotique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Langue d'enseignement	français
Crédits	4
Retrait	Non autorisé
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Projet	1 hebdo
Nombre de places	

Il n'est pas autorisé de se retirer de cette matière après le délai d'inscription.

Résumé

Ce cours a pour objectif de présenter l'approche générale du management de projet en intégrant la gestion du risque dans toutes les étapes du projet.

Contenu

Le projet est un processus consistant à réaliser un système dans le but de satisfaire le besoin de futurs utilisateurs. Il sera ainsi aborder dans ce cours la présentation des différentes phases du projet, les organisations de projet, les moyens et méthodes de développement des variantes de projets, le choix multicritère de la variante à réaliser, la planification et le suivi dans la phase de réalisation. Une approche « risque » sera abordée à chaque étape du projet par une méthodologie probabiliste.

- Les principes généraux du management de projet
- Les phases d'un projet
- L'approche stratégique et opérationnel d'un projet
- Le management des risques
- Les formes organisationnelles de projets
- Evaluation physique de variantes de projets
- Evaluation comportementale de variantes de projets par simulation numérique
- Evaluation économique de projet
- choix de variante(s), analyse multicritère
- Planification et ordonnancement de projet
- Exemples et études de cas
- Applications informatiques

Mots-clés

Management de projet, risques, évaluation économique, planification, analyse multicritère

Compétences requises

Cours prérequis obligatoires

Néant

Cours prérequis indicatifs

Notions de base de statisques

Concepts importants à maîtriser

Ouvert, curieux et capable d'aborder un domaine complexe multidisciplinaire et multiculturel.
L'étudiant devra avoir une vision transversale des processus et être capable de raisonner de manière systémique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Diriger une équipe de projet
- Elaborer des variantes de projet
- Planifier les variantes de projet
- Sélectionner ou choisir la variante retenue
- Anticiper les risques
- Organiser
- Restituer et communiquer
- Implémenter

Compétences transversales

- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Etre responsable des impacts environnementaux de ses actions et décisions.
- Dialoguer avec des professionnels d'autres disciplines.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Recueillir des données.
- Faire une présentation orale.

Méthode d'enseignement

Ex cathedra, projet avec présentation orale

Travail attendu

Suivi des cours et étude des documents de cours distribués
Réalisation d'un projet durant le semestre. Ce projet est réalisé en groupe et présenté dans les dernières séances du cours

Méthode d'évaluation

- 50% projet durant le semestre
- 50% examen final

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Autres	Disponibilité de l'enseignant par email, téléphone ou visite à son bureau

Ressources

Bibliographie

Indiquée en rapport avec chaque chapitre du cours

Polycopiés

Cours et copies des slides de présentation envoyés à chaque étudiant sous format électronique

Liens Moodle

- https://go.epfl.ch/MGT-427

Préparation pour

Travaux de semestre
Projet de Master

COM-516 Markov chains and algorithmic applications

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

The study of random walks finds many applications in computer science and communications. The goal of the course is to get familiar with the theory of random walks, and to get an overview of some applications of this theory to problems of interest in communications, computer and network science.

Content

Part 1: Markov chains (~6 weeks):
- basic properties: irreducibility, periodicity, recurrence/transience, stationary and limiting distributions,
- ergodic theorem: coupling method
- detailed balance
- convergence rate to the equilibrium, spectral gap, mixing times
- cutoff phenomenon

Part 2: Sampling (~6 weeks)
- classical methods, importance and rejection sampling
- Markov Chain Monte Carlo methods, Metropolis-Hastings algorithm, Glauber dynamics, Gibbs sampling
- applications: function minimization, coloring problem, satisfiability problems, Ising models
- coupling from the past and exact simulation

Keywords

random walks, stationarity, ergodic, convergence, spectral gap, mixing time, sampling, Markov chain Monte Carlo, coupling from the past

Learning Prerequisites

Required courses

Basic probability course
Basic linear algebra and calculus courses

Recommended courses

Stochastic Models for Communications (COM-300)

Important concepts to start the course

Good knowledge of probability and analysis.
Having been exposed to the theory of Markov chains.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze the behaviour of a random walk
- Assess / Evaluate the performance of an algorithm on a graph
- Implement efficiently various sampling methods

Teaching methods

ex-cathedra course

Expected student activities

active participation to exercise sessions and implementation of a sampling algorithm

Assessment methods

midterm (20%), mini-project (20%), final exam (60%)

Resources

Bibliography

Various references will be given to the students during the course, according to the topics discussed in class.

Ressources en bibliothèque

- Probability and random processes / Grimmett

Notes/Handbook

Lecture notes will be provided

Websites

- https://moodle.epfl.ch/course/view.php?id=15016

Moodle Link

- https://go.epfl.ch/COM-516

Prerequisite for

This course is not so to speak a prerequisite for other courses, but could complement well the course COM-512 on Networks out of control, as well as other courses in statistics.

COM-405

Mobile networks

Al Hassanieh Haitham

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course provides a detailed description of the organization and operating principles of mobile and wireless communication networks, as well as the use of wireless signals for sensing and imaging.

Content

- Introduction to wireless networks
- Wireless PHY Layer Techniques
- MAC (Medium Access Control) Layer Protocols
- Wi-Fi & Bluetooth
- Cellular networks (3G, 4G, 5G).
- Internet of Things (IoT) Networks and Technologies.
- Multi-Hop Networks, Mesh Networks, and Sensor Networks
- Routing in Wireless Networks
- Network Coding
- Cross Layer Networking
- Wireless Localization
- Wireless Sensing
- Wireless Imaging

Keywords

Communication networks, protocols, wireless, IoT

Learning Prerequisites

Required courses

CS-202 Computer systems or COM-208 Computer Networks
COM-302 Principles of Digital Communications

Recommended courses

COM-430 Modern Digital Communications: A Hands-on Approach

Important concepts to start the course

Operating principles of communication protocols and layer organization.

Learning Outcomes

By the end of the course, the student must be able to:

- Synthesize the way a mobile network operates
- Interpret the behavior of such networks
- Propose evolutions to existing protocols
- Identify weaknesses and bottlenecks

Teaching methods

Lectures
Weekly Readings
Exercise sessions
Homework Problems
Labs

Expected student activities

Class Participation, Quizzes, Homework, Labs, Exercise Sessions

Assessment methods

Homeworks
Quizzez
Labs
Final exam

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Handouts, recommended books (see course URL)

Ressources en bibliothèque

-

Moodle Link

- https://go.epfl.ch/COM-405

EPFL

COM-430 **Modern digital communications: a hands-on approach**

Chiurtu Nicolae

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course complements the theoretical knowledge learned in PDC with more advanced topics such as OFDM, MIMO, fading channels, and GPS positioning. This knowledge is put into practice with hands-on exercises based on Matlab or Python (at choice) and on a software-defined radio platform.

Content

1. Software radio : key concepts.
2. Matlab/Python implementation of the signal processing chain to the level of detail taught in Principles of Digital Communications (PDC: COM-302).
3. Channel modeling, estimation, equalization.
4. Implementation of a basic wireless communication system using a software-defined radio testbed.
5. Fading and diversity.
6. OFDM and MIMO : theory and implementation.
7. CDMA in the context of a GPS system.
8. Decoding of a GPS signal and positioning.

Keywords

Wireless, OFDM, Diversity, Coding, GPS, CDMA, MMSE, Rayleigh fading, software-defined radio, channel estimation.

Learning Prerequisites

Required courses

COM-302 Principles of Digital Communications (PDC) or equivalent.

Important concepts to start the course

Solid understanding of linear algebra and probability as well as real and complex analysis.

Learning Outcomes

By the end of the course, the student must be able to:

- Design and implement an advanced digital communication system (data rate, spectral bandwidth, energy requirements, error probability, implementation complexity).
- Model the physical properties of wired and wireless communication channels.
- Implement various parts of a "physical-layer" digital communication system.
- Understand what software-defined radio is all about.

Teaching methods

Ex cathedra lectures and small projects.

Expected student activities

Follow lectures; guided as well as independent work on projects.

Assessment methods

Written and practical midterm and final exam during the semester.
40% midterm exam, 60% final exam.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Notes/Handbook

Lecture notes

Moodle Link

- https://go.epfl.ch/COM-430

EE-452

Network machine learning

Frossard Pascal, Thanou Dorina

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Fundamentals, methods, algorithms and applications of network machine learning and graph neural networks

Content

Context

In the last decade, our information society has mutated into a data society, where the volume of worldwide data grows increasingly fast. An increasing amount of this data is structured on networks of different forms. How to make sense of such tremendous volume of data? Developing effective techniques to extract meaningful information from large-scale and high-dimensional network datasets has become essential for the success of business, government and science.

Objective

The goal of this course is to provide a broad introduction to effective methods algorithms in data science, network analysis and network machine learning. A major effort will be given to show that existing data analysis techniques can be defined and enhanced on graphs. Graphs can encode complex structures like cerebral connection, stock exchange, and social network. Strong mathematical tools have been developed based on statistics, or linear and non-linear graph spectral harmonic analysis to advance the standard data analysis algorithms. At the same time, modern machine learning tools such as neural networks have been adapted to process data defined on network structures. The objective of the class is to develop fundamentals and review algorithms that permit to develop modern network data analysis methods. The main topics of the course are networks, network data analysis, unsupervised and supervised learning on graphs and networks, graph generative models, sparse representation, multi-resolution analysis, graph neural networks.

Structure

The course is organized into two parts: lectures (2 hours) and lab assignements and projects (1 hour). The essential objective of the exercises and lab assignements is to apply the theory on real-world cases. The objective of the projects is to study practical network machine learning cases, and develop effective solutions based on tools studied in the class.

Evaluation

Evaluation will be conducted on a continuous basis: homeworks and coding assignments.

Keywords

graph representation learning, machine learning, network science

Learning Prerequisites

Required courses

Fundamentals of Machine Learning, or equivalent
Signal Processing, or equivalent
Introduction to Statistics, or equivalent
Python programming

Learning Outcomes

By the end of the course, the student must be able to:

- Apply modern machine learning techniques to network data
- Analyze network properties, network data distributions, and properties of the most common network machine learning algorithms
- Propose solutions for network data analysis problems

Transversal skills

- Use a work methodology appropriate to the task.
- Give feedback (critique) in an appropriate fashion.
- Communicate effectively, being understood, including across different languages and cultures.

Resources

Moodle Link

- https://go.epfl.ch/EE-452

EPFL

COM-512 **Networks out of control**

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - Pas donné en 2024-25

Summary

The goal of this class is to acquire mathematical tools and engineering insight about networks whose structure is random, as well as learning and control techniques applicable to such network data.

Content

- Random graph models: Erdös-Renyi, random regular, geometric, percolation, small worlds, stochastic block model
- Learning graphs from data: centrality metrics, embeddings, Hawkes processes, network alignment
- Control of processes on graphs: epidemics, navigation

Keywords

Random graphs, network data, machine learning, graph processes.

Learning Prerequisites

Required courses

Stochastic models in communication (COM-300), or equivalent.

Important concepts to start the course

Basic probability and stastistics; Markov chains; basic combinatorics.

Teaching methods

Ex cathedra lectures, exercises, mini-project

Expected student activities

Attending lectures, bi-weekly homeworks, mini-project incl. student presentation at the end of semester, final exam.

Assessment methods

1. Homeworks 10%
2. Mini-project 40%
3. Final exam 50%.

EPFL

Resources

Bibliography

- A. D. Barbour, L. Holst and S. Janson, Poisson Approximation, Oxford Science Publications, 1992.
- B. Bollobas, Random Graphs (2nd edition), Cambridge University Press, 2001.
- R. Durrett, Random Graph Dynamics, Cambridge University Press, 2006 (electronic version).
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010 (electronic version).
- G. Grimmett, Percolation (2nd edition), Springer, 1999.
- S. Janson, T. Luczak, A. Rucinski, Random Graphs, Wiley, 2000.
- R. Meester and R. Roy, Continuum Percolation, Cambridge University Press, 1996.

Ressources en bibliothèque

- Random Graphs / Bollobas
- Random Graphs / Janson
- Continuum Percolation / Meester
- Percolation / Grimmett
- Networks, Crowds and Markets / Easley
- Poisson Approximation / Barbour
- Random Graph Dynamics / Durrett

Notes/Handbook

Class notes will be available on the course website.

Moodle Link

- https://go.epfl.ch/COM-512

EPFL

MATH-489 **Number theory II.c - Cryptography**

Jetchev Dimitar

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The goal of the course is to introduce basic notions from public key cryptography (PKC) as well as basic number-theoretic methods and algorithms for cryptanalysis of protocols and schemes based on PKC.

Content

Basic notions and algorithms from public key cryptography such as RSA, ElGamal, key exchange protocols, zero knowledge proofs. Main topics may include, but are not limited to

- modular and finite field arithmetic
- primality testing
- polynomial and integer factorization algorithms
- index calculus and discrete logarithm-based schemes
- elliptic curve arithemtic and cryptography
- basic notions from lattice-based cryptography: lattice-basis reduction algorithms, learning-with-errors, applications to homomorphic encryption

Keywords

public key cryptography, key exchange, digital signatures, zero knowledge proofs, RSA, ElGamal, integer factorization, index calculus, elliptic curve cryptography, lattice-based cryptography

Learning Prerequisites

Recommended courses

Some knowledge of abstract algebra (groups, rings and fields) is desirable, but not mandatory. Knowledge of basic algorithms is a plus.

Assessment methods

Homework assignments: Weekly problem sets focusing on number-theoretic and complexity-theoretic aspects. These will be complemented by programming exercises in SAGE which is a Python-based computer algebra system. No prior experience with SAGE or Python is required. A subset of the homework will be handed in and graded, counting for 40% of the final grade.
The written **final exam** counts for 60% of the final grade. There will be a mock midterm exam not to be part of the final grade. The final exam will test theoretical understanding as well as understanding of the algorithms and protocols. The exam will include no SAGE programming exercises. If needed, algorithms could be presented with pseudo-code. The exact final exam format will be adapted to the epidemiological situation and resulting guidelines.

Resources

Moodle Link

- https://go.epfl.ch/MATH-489

EPFL

CS-439 **Optimization for machine learning**

Flammarion Nicolas

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	1 weekly
Number of positions	

Summary

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.

Content

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.
Fundamental Contents:

- Convexity, Gradient Methods, Proximal algorithms, Stochastic and Online Variants of mentioned methods, Coordinate Descent Methods, Subgradient Methods, Non-Convex Optimization, Frank-Wolfe, Accelerated Methods, Primal-Dual context and certificates, Lagrange and Fenchel Duality, Second-Order Methods, Quasi-Newton Methods, Gradient-Free and Zero-Order Optimization.

Advanced Contents:

- Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods, Alternating minimization for matrix and tensor factorizations
- Parallel and Distributed Optimization Algorithms, Synchronous and Asynchronous Communication
- Lower Bounds

On the practical side, a graded **group project** allows to explore and investigate the real-world performance aspects of the algorithms and variants discussed in the course.

Keywords

Optimization, Machine learning

Learning Prerequisites

Recommended courses

- CS-433 Machine Learning

Important concepts to start the course

- Previous coursework in calculus, linear algebra, and probability is required.
- Familiarity with optimization and/or machine learning is useful.

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the most important algorithms, function classes, and algorithm convergence guarantees
- Compose existing theoretical analysis with new aspects and algorithm variants.
- Formulate scalable and accurate implementations of the most important optimization algorithms for machine learning applications
- Characterize trade-offs between time, data and accuracy, for machine learning methods

Transversal skills

- Use both general and domain specific IT resources and tools
- Summarize an article or a technical report.

Teaching methods

- Lectures
- Exercises with Theory and Implementation Assignments

Expected student activities

Students are expected to:

- Attend the lectures and exercises
- Give a short scientific presentation about a research paper
- Read / watch the pertinent material
- Engage during the class, and discuss with other colleagues

Assessment methods

- Continuous control (course project)
- Final Exam

Resources

Websites

- https://github.com/epfml/OptML_course

Moodle Link

- https://go.epfl.ch/CS-439

Videos

- https://www.youtube.com/playlist?list=PL4O4bXkI-fAeYrsBqTUYn2xMjJAqlFQzX

EPFL

CS-596

Optional research project in computer science II

Profs divers *

Cursus	Sem.	Type
Computer science minor	E, H	Opt.
Computer science	MA1, MA2, MA3, MA4	Opt.
Cybersecurity	MA1, MA2, MA3, MA4	Opt.

Language of teaching	English
Credits	8
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**2 weekly**
Project	2 weekly
Number of positions	

Remark

for students doing a minor in Computer science:Registration upon approval of the section. Only for 2nd year Master students. Supervision by an IC authorized professor

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Supervisor and subject to be chosen among the themes proposed on the web site :
Projects by laboratory

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Teaching methods

Individual and independant work, under the guidance of a professor or an assistant.

Expected student activities

Written report due within the allotted time.
Information on the format and the content of the report is provied by the project supervisor.

Assessment methods

Automn : The written report must be returned to the laboratory no later than **the Friday of the second week** after the end of classes.
Spring : The written report must be returned to the laboratory no later than **the Friday of the first week** after the end of classes.
The oral presentation is organized by the laboratory.

Resources

Websites

- https://www.epfl.ch/schools/ic/education/master/semester-project-msc/

Moodle Link

- https://go.epfl.ch/CS-596

EPFL

CS-522 **Principles of computer systems**

Argyraki Katerina, Candea George

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	4 weekly
Number of positions	

Summary

This advanced graduate course teaches the key design principles underlying successful computer and communication systems, and shows how to solve real problems with ideas, techniques, and algorithms from operating systems, networks, databases, programming languages, and computer architecture.

Content

A modern computer system spans many layers: applications, libraries, operating systems, networks, and hardware devices. Building a good system entails making the right trade-offs (e.g., between performance, durability, and correctness) and understanding emergent behaviors. Great system designers make these trade-offs in a principled fashion, whereas average ones make them by trial-and-error. In this course we develop a principled framework for computer system design, covering the following topics:

- Modules and interfaces
- Names
- Layers
- Indirection and virtualization
- Redundancy and fault tolerance
- Client/server architectures
- Decentralized architectures
- Transactional building blocks

Learning Prerequisites

Required courses

The course is intellectually challenging and fast-paced, and it requires a solid background in operating systems, databases, networking, programming languages, and computer architecture. The basic courses on these topics teach how the elemental parts of modern systems work, and this course picks up where the basic courses leave off. To do well, a student must master the material taught in the following courses:

- CS-200 Computer architecture
- CS-214 Software construction
- CS-311 The Software enterprise - from ideas to product
- CS-300 Data-intensive systems

Recommended courses

The following EPFL courses cover material that significantly helps students taking this course, however they are not strictly required:
- CS-320: Computer language processing
- CS-470: Advanced computer architecture
- CS-422: Database systems
- COM-407: TCP/IP networking

Learning Outcomes

By the end of the course, the student must be able to:
- Design computer and communication systems that work well
- Make rational design trade-offs (e.g., performance vs. correctness, latency vs. availability)
- Anticipate emergent system behaviors (e.g., failure cascades, security vulnerabilities)
- Integrate multiple techniques, ideas, and algorithms from different fields of computing/communication into a working system

Teaching methods

- A combination of online and in-class lectures
- Interactive design sessions
- Reading assignments
- Homework assignments

Expected student activities

- Attend lectures and design sessions
- Complete the reading and writing assignments
- Participate actively in the course (physically and online)

Assessment methods

- 50% OPs
- 50% final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

See course website for the latest information and an up-to-date bibliography.

Ressources en bibliothèque

- Principles of computer system design : an introduction / Saltzer

Websites

- https://pocs.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-522

EPFL

CS-496

Research project in Cyber security

Profs divers *

Cursus	Sem.	Type
Cybersecurity	MA1, MA2, MA3, MA4	Obl.

Language of teaching	English
Credits	12
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	360h
Weeks	14
Hours	**12 weekly**
Project	12 weekly
Number of positions	

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Supervisor and subject to be chosen among the themes proposed on the web site :
Projects by laboratory

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Teaching methods

Individual and independant work, under the guidance of a professor or an assistant.

Expected student activities

Written report due within the allotted time.
Information on the format and the content of the report is provied by the project supervisor.

Assessment methods

Automn : The written report must be returned to the laboratory no later than **the Friday of the second week** after the end of classes.
Spring : The written report must be returned to the laboratory no later than **the Friday of the first week** after the end of classes.
The oral presentation is organized by the laboratory.

Resources

Websites

- https://www.epfl.ch/schools/ic/education/master/semester-project-msc/

Moodle Link

- https://go.epfl.ch/CS-596

EPFL

CS-412 **Software security**

Payer Mathias

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	1 weekly
Number of positions	

Summary

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with security pitfalls.

Content

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students will learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with common security pitfalls.
Software running on current systems is exploited by attackers despite many deployed defence mechanisms and best practices for developing new software. In this course students will learn about current security threats, attack vectors, and defence mechanisms on current systems. The students will work with real world problems and technical challenges of security mechanisms (both in the design and implementation of programming languages, compilers, and runtime systems).

- Secure software lifecycle: design, implementation, testing, and deployment
- Basic software security principles
- Reverse engineering : understanding code
- Security policies: Memory and Type safety
- Software bugs and undefined behavior
- Attack vectors: from flaw to compromise
- Runtime defense: mitigations
- Software testing: fuzzing and sanitization
- Focus topic: Web security
- Focus topic: Mobile security

Keywords

Software security, mitigation, software testing, sanitization, fuzzing

Learning Prerequisites

Required courses

- COM-402 Information security and privacy (or an equivalent security course)
- A systems programming course (with focus on C/C++)
- An operating systems course

Important concepts to start the course

Basic computer literacy like system administration, build systems, C/C++ programming skills, debugging, and development skills. Understanding of virtual machines and operating systems.

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the top 20 most common weaknesses in software security and understand how such problems can be avoided in software.
- Identify common security threats, risks, and attack vectors for software systems.
- Assess / Evaluate current security best practices and defense mechanisms for current software systems. Become aware of limitations of existing defense mechanisms and how to avoid them.
- Identify security problems in source code and binaries, assess the associated risks, and reason about their severity and exploitability.
- Assess / Evaluate the security of given source code or applications.

Transversal skills

- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Keep appropriate documentation for group meetings.
- Summarize an article or a technical report.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Make an oral presentation.

Teaching methods

The lectures are denser early in the semester, then tapering off before the end. They are backed up by PDF files of all the lecture material, as well as a few textbook recommendations.
The exercises sessions start slowly early in the semester but pick up and occupy all time towards the end. Homework exercises consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.
The labs focus on practical software security aspects and during the course the students will be assessed through their

completion of several challenging "hands on" labs.

Expected student activities

Students are encouraged to attend lectures and exercise sessions. In addition to normal studying of the lecture and practice of the exercises, the reading assignment consists of analyzing a few suggested scientific papers on a large selection of topics; the presentation assignment consists of holding a 15-minute presentation on the selected topic; and the writing assignment of documenting what was learned in a term paper due at the end of the semester.

Assessment methods

The grade will continuously be evaluated through a combination of pracitcal assignments in the form of several labs and theoretical quizzes throughout the semester. The labs will account for 50%, the quizzes and tests to 50%.
The exact dates of the labs/quizzes will be communicated at the beginning of the class.

Resources

Notes/Handbook

Software Security: Principles, Policies, and Protection (SS3P, by Mathias Payer)
https://nebelwelt.net/SS3P/

Moodle Link

- https://go.epfl.ch/CS-412

EPFL

PHYS-512 **Statistical physics of computation**

Erba Vittorio

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Ing.-phys	MA1, MA3	Opt.
Physicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The students understand tools from the statistical physics of disordered systems, and apply them to study computational and statistical problems in graph theory, discrete optimisation, inference and machine learning.

Content

Interest in the methods and concepts of statistical physics is rapidly growing in fields as diverse as theoretical computer science, probability theory, machine learning, discrete mathematics, optimization, signal processing and others. Large part of the related work has relied on the use of message-passing algorithms and their connection to the statistical physics of glasses and spin glasses.

This course covers this active interdisciplinary research landscape. Specifically, we will review the statistical physics approach to problems ranging from graph theory (e.g. community detection) to discrete optimization and constraint satisfaction (e.g. satisfiability or coloring) and to inference and machine learning problems (learning in neural networks, clustering of data and of networks, compressed sensing or sparse linear regression, low-rank matrix factorization).

We will expose theoretical methods of analysis (replica, cavity, ...) algorithms (message passing, spectral methods, etc), discuss concrete applications, highlight rigorous justifications as well as present the connection to the physics of glassy and disordered systems.

This is an advanced theoretical course that is designed for students with background in mathematics, electrical engineering, computer science or physics. This course exposes advanced theoretical concepts and methods, with exercises in the analytical methods and usage of the related algorithms.

Learning Prerequisites

Important concepts to start the course

For physics students Statistical physics I and II (or equivalent) is required.
This lecture is accessible to students in mathematics, electrical engineering, computer science without any previous training in statistical physics. Those students are expected to have strong interest in theory, probabilistic approaches to analysis of algoriths, high-dimensional statistics or probabilistic signal processing.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze theoretically a range of problems in computer science and learning.
- Derive algorithms for a range of computational problems using technics stemming from statistical physics.

Teaching methods

2h of lecture + 2h of excercise

Assessment methods

Final written exam counting for 50% and graded homework during the semester counting for the other 50%.

Resources

Bibliography

Information, Physics and Computation (Oxford Graduate Texts), 2009, M. Mézard, A. Montanari
Statistical Physics of inference: Thresholds and algorithms, Advances in Physics 65, 5 2016, L. Zdeborova & F. Krzakala, available at https://arxiv.org/abs/1511.02476

Ressources en bibliothèque

- Information, Physics and Computation / Mézard

Références suggérées par la bibliothèque

- Statistical Physics of inference: Thresholds and algorithms / Zdeborova & Krzakala

Notes/Handbook

Policopié "Statistical Physics methods in Optimization & Machine Learning" by L. Zdeborova & F. Krzakala, available athttps://sphinxteam.github.io/EPFLDoctoralLecture2021/Notes.pdf

Moodle Link

- https://go.epfl.ch/PHYS-512

EPFL

COM-500 Statistical signal and data processing through applications

Ridolfi Andrea

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Obl.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly
Number of positions	

Summary

Building up on the basic concepts of sampling, filtering and Fourier transforms, we address stochastic modeling, spectral analysis, estimation and prediction, classification, and adaptive filtering, with an application oriented approach and hands-on numerical exercises.

Content

1. Fundamentals of Statistical Signal and Data Processing: Signals and systems from the deterministic and the stochastic point of view; Processing and analysing signals and systems with a mathematical computing language.
2. Models, Methods, and Algorithms: Parametric and non-parametric signal models (wide sense stationary, Gaussian, Markovian, auto-regressive and white noise signals); Linear prediction and estimation (orthogonality principle and Wiener filter); Maximum likehood estimation and Bayesian a priori; Maximum a posteriori estimation.
3. Statistical Signal and Data Processing Tools for Spread Spectrum Wireless Transmission: Coding and decoding of information using position of pulses (annihilating filter approach); Spectrum estimation (periodogram, line spectrum methods, smooth spectrum methods, harmonic signals).
4. Statistical Signal and Data Processing Tools for the Analysis of Neurobiological Recordings: Poisson process for neurobiological spikes; Characterization of multi-state neurons (Markovian models and maximum likelihood estimation); Classifying firing rates of neuron (Mixture models and the EM algorithm); Hidden Markov models; Spike sorting and Principal Component Analysis.
5. Statistical Signal and Data Processing Tools for Echo Cancellation: Adaptive filtering (least mean squares and recursive least squares); Adaptive echo cancellation and denoising.

Keywords

Statistical tools, spectral analysis, prediction, estimation, annihilating filter, mixture models, principal component analysis, stochastic processes, hidden Markov models, adaptive filtering, mathematical computing language (Matlab, Python, or similar).

Learning Prerequisites

Required courses

Stochastic Models in Communications (COM-300), Signal Processing for Communications (COM-303) / Signal Processing (COM-202).

Important concepts to start the course

Calculus, Algebra, Fourier Transform, Z Transform, Probability, Linear Systems, Filters.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose appropriate statistical tools to solve signal processing problems;
- Analyze real data using a mathematical computing language;
- Interpret spectral content of signals;
- Develop appropriate models for observed signals;
- Assess / Evaluate advantages and limitations of different statistical tools for a given signal processing problem;
- Implement numerical methods for processing signals.

Transversal skills

- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking
- Access and evaluate appropriate sources of information.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Ex cathedra with exercises and numerical examples.

Expected student activities

Attendance at lectures, completing exercises, testing presented methods with a mathematical computing language (Matlab, Python, or similar).

Assessment methods

- 20% midterm
- 20% mini project
- 60% Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Background texts

- P. Prandoni, *Signal Processing for Communications*, EPFL Press;
- P. Bremaud, *An Introduction to Probabilistic Modeling*, Springer-Verlag, 1988;
- A.V. Oppenheim, R.W. Schafer, *Discrete Time Signal Processing*, Prentice Hall, 1989;
- B. Porat, *A Course in Digital Signal Processing*, John Wiley & Sons,1997;
- C.T. Chen, *Digital Signal Processing*, Oxford University Press;
- D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability,* Athena Scientific, 2002 (excellent book on probability).

More advanced texts

- L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*, Springer-Verlag, 1988;

• A.N. Shiryaev, *Probability*, Springer-Verlag, New York, 2nd edition, 1996;
• S.M. Ross, *Introduction to Probability Models*, Third edition, 1985;
• P. Bremaud, Markov Chains, Springer-Verlag, 1999;
• P. Bremaud, Mathematical Principles of Signal Processing, Springer-Verlag, 2002;
• S.M. Ross, *Stochastic Processes*, John Wiley, 1983;
• B. Porat, *Digital Processing of Random Signals*, Prentice Hall,1994;
• P.M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, 1993;
• P. Stoïca and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, 1997.

Ressources en bibliothèque

• Probability / Shiryaev
• Stochastics Processes / Ross
• Discrete Time Signal Processing / Oppenheim
• Introduction to Spectral Analysis / Stoïca
• Digital Processing of Random Signals / Porat
• Introduction to Probability / Bertsekas
• Introduction to Hilbert Spaces with Applications / Debnath
• Signal Processins for Communications / Prandoni
• An Introduction to Probabilistic Modeling / Bremaud
• A Course in Digital Signal Processing / Porat
• Optimal and Adaptive Signal Processing / Clarkson
• Digital Signal Processing / Chen
• Introduction to Probability Models / Ross

Notes/Handbook

• Slides handouts;
• Collection of exercises.

Moodle Link

• https://go.epfl.ch/COM-500

EPFL

COM-506 **Student seminar: security protocols and applications**

Vaudenay Serge

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Session	Summer
Semester	Spring
Exam	During the semester
Workload	90h
Weeks	14
Hours	**2 weekly**
Lecture	2 weekly
Number of positions	

Summary

This seminar introduces the participants to the current trends, problems, and methods in the area of communication security.

Content

We will look at today's most popular security protocols and new kinds of protocols, techniques, and problems that will play an emerging role in the future. Also, the seminar will cover methods to model and analyze such security protocols. This course will be held as a seminar, in which the students actively participate. The talks will be assigned in the first meeting to teams of students, and each team will have to give a 45 minutes talk, react to other students' questions, and write a 3-4 pages summary of their talk.

Keywords

network security, security protocols, cryptography

Learning Prerequisites

Required courses

- Computer security (COM-301)
- Cryptography and security (COM-401)

Learning Outcomes

By the end of the course, the student must be able to:

- Synthesize some existing work on a security protocol
- Analyze a security protocol
- Present a lecture

Transversal skills

- Make an oral presentation.
- Summarize an article or a technical report.

Expected student activities

- prepare a lecture (presentation and a 4-page report)
- present the lecture
- attend to others' lectures and grade them

Assessment methods

- lecture and attendance to others' lectures

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-506

Videos

- https://mediaspace.epfl.ch/channel/COM-506+Student+Seminar+on+Security+Protocols+and+Applications

EPFL

CS-448 **Sublinear algorithms for big data analysis**

Kapralov Michael

Cursus	Sem.	Type
Computer and Communication Sciences		Obl.
Computer science minor	E	Obl.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal, donné les années impaires

Summary

In this course we will define rigorous mathematical models for computing on large datasets, cover main algorithmic techniques that have been developed for sublinear (e.g. faster than linear time) data processing. We will also discuss limitations inherent to computing with constrained resources.

Content

The tentative list of topics is:

Streaming: given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems such as approximating shortest path distances, maximum matchings etc.;
Sketching: what can we learn about the input from a few carefully designed measurements (i.e. a `sketch') of the input, or just a few samples of the input? We will cover several results in sparse recovery and property testing that answer this question for a range of fundamental problems;
Sublinear runtime: which problems admit solutions that run faster than it takes to read the entire input? We will cover sublinear time algorithms for graph processing problems, nearest neighbor search and sparse recovery (including Sparse FFT);
Communication: how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs, as well as other techniques.

Keywords

streaming, sketching, sparse recovery, sublinear algorithms

Learning Prerequisites

Required courses

Bachelor courses on algorithms, complexity theory, and discrete mathematics

Important concepts to start the course

Discrete probability; mathematical maturity

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class
- Analyze space/time/communication complexity of randomized algorithms
- Prove space/time/communication lower bounds for variations of problems discussed in class
- Choose an appropriate algorithmic tool for big data problem at hand

Teaching methods

Ex cathedra, homeworks, final

Assessment methods

Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-448

CS-473

System programming for Systems-on-chip

Kluter Ties Jan Henderikus

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

To efficiently program embedded systems an understanding of their architectures is required. After following this course students will be able to take an existing SoC, understand its architecture, and efficiently program it.

Content

Hardware elements found in embedded systems:

- Flash, tightly couples memories, SDRAM, DDR.
- IO-interfaces and protocols (RS232, I2C, I2S, SPI).
- Bus architectures.

Architecture imposed restrictions:

- Memory map and memory holes.
- Cached and non-cached regions.
- Interrupt latencies, bus latencies, task-switch latencies.

Software techniques:

- BIOS/firmware
- DMA and computation/data-transfer overlaps
- Hot-spot detection and hardware/software profiling

Learning Prerequisites

Recommended courses

CS-200 Computer architecture or CS-208 Computer architecture I; CS-209 Computer architecture II

Important concepts to start the course

- C/C++ programming skills
- Basic Verilog knowledge

Learning Outcomes

- Analyze and understand the architecture of embedded systems (SoC's)
- Write a firmware that initializes an embedded system and efficiently implement the required functionality
- Explain the different latencies present in an embedded system and how these latencies influence the execution time on the firmware
- Profile an embedded system and pin-point the hardware related and software induced hot-spots
- understand the different types of memories present in an embedded systems and how to use them
- Program the different types of I/O devices present in an embedded system and know how their protocol works

Teaching methods

Ex cathedra with practical exercises (in groups of 2 students)

Expected student activities

- Reports of practical exercises
- Written exam

Assessment methods

- Lab reports : 50%
- Final written exam : 50%

Supervision

Office hours	No
Assistants	Yes
Others	Electronic forum and Moodle

Resources

Moodle Link

- https://go.epfl.ch/CS-473

EPFL

CS-460 **Systems for data management and data science**

Ailamaki Anastasia, Kermarrec Anne-Marie

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course is intended for students who want to understand modern large-scale data analysis systems and database systems. The course covers fundamental principles for understanding and building systems for managing and analyzing large amounts of data. It covers a wide range of topics and technologi

Content

Topics include large-scale data systems design and implementation, and specifically :

- Distributed data management systems
- Data management : locality, accesses, partitioning, replication
- Modern storage hierarchies
- Query optimization, database tuning
- Transaction management
- Data structures : File systems, Key-value stores, DBMS
- Consistency models
- Large-scale data analytics infrastructures
- Parallel Processing
- Data stream and graph processing

Learning Prerequisites

Recommended courses

- CS-107 Introduction to programming
- CS-214 Software construction
- CS-300 Data-Intensive Systems
- CS-202 Computer systems
- CS-452 Foundations of software

Important concepts to start the course

- Algorithms and data structures.
- Scala and/or Java programming languages will be used throughout the course. Programming experience in one of these languages is strongly recommanded.
- Basic knowledge or computer networking and distributed systems.

Learning Outcomes

By the end of the course, the student must be able to:

- Understand how to design big data analytics systems using state-of-the-art infrastructures for horizontal scaling, e.g., Spark
- Implement algorithms and data structures for streaming data analytics
- Decide between different storage models based on the offered optimizations enabled by each modl and the expected query workload
- Compare concurrency control algorithms, and algorithms for distributed data management
- Configure systems parameters, data layouts, and application designs for database systems
- Develop data-parallel analytics programs that make us of modern clusters and cloud offerings to scale up to very large workloads
- Analyze the trade-offs between various approaches to large-scale data management and analytics, depending on efficiency, scalability, and latency needs

Teaching methods

Lectures, project, homework, exercises and practical work

Expected student activities

- Attend lectures and participate in class
- Complete a project as per the guidelines posted by the teaching team

Assessment methods

- Project
- Midterm (as needed)
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

J. Hellerstein & M. Stonebraker, Readings in Database Systems, 4th Edition, 2005
R. Ramakrishnan & J. Gehrke: "Database Management Systems", McGraw-Hill, 3rd Edition, 2002.
A. Rajaraman & J. Ullman: "Mining of Massive Datasets", Cambridge Univ. Press, 2011.

Ressources en bibliothèque

- Mining of Massive Datasets / Rajaraman
- Database Management Systems / Ramakrishnan
- Readings in Database Systems / Hellerstein

Moodle Link

- https://go.epfl.ch/CS-460

EPFL

COM-407 **TCP/IP networking**

Nikolopoulos Pavlos

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

In the lectures you will learn and understand the main ideas that underlie and the way communication networks are built and run. In the labs you will exercise practical configurations.

Content

-- The internet architecture.
-- Layer 2 networking; switching/bridging.
-- The Internet protocol versions 4 and 6.
-- The transport layer, TCP, UDP, sockets, QUIC.
-- Routing algorithms: Link state routing, OSPF, Distance Vector routing. Interdomain routing, BGP.
-- Congestion control principles. The fairness of TCP. Application to the Internet (TCP Reno, Cubic, DCTCP, BBR).
-- Tunnels and hybrid architectures.
-- A few things about internet security.
-- Application layer protocols.

Keywords

TCP/IP
Computer Networks

Learning Prerequisites

Required courses

A first programming course (Python)

Recommended courses

An undergraduate course on Computer Networks

Learning Outcomes

By the end of the course, the student must be able to:

- Run and configure networks
- Understand the main ideas that underlie the Internet
- Write simple communicating programs
- Use communication primitives for internet and industrial applications.

Transversal skills

- Access and evaluate appropriate sources of information.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Lectures.
Online quizzes.
Labs on student's computer.

Expected student activities

Participate in lectures
Participate in online quizzes
Make lab assignments (in the rule, every other week)

Assessment methods

Theory grade = final exam
Practice grade = average of labs
Final grade = mean of theory grade (50%) and practice grade (50%).
The research exercice may add a bonus of at most 0.5 points in 1-6 scale to the practice grade.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

"Computer Networking: A Top-Down Approach (6th or later Edition)". James F. Kurose and Keith W. Ross. 2012. Pearson.

"Computer Networking : Principles, Protocols and Practice". O. Bonaventure, open source textbook, http://inl.info.ucl.ac.be/CNP3

Ressources en bibliothèque

-
- Computer Networking / Kurose

Notes/Handbook

Slides are on moodle

Websites

- http://moodle.epfl.ch/course/view.php?id=523

Moodle Link

- https://go.epfl.ch/COM-407

EPFL

CS-510

Topics in software security

Payer Mathias

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	3
Session	Winter
Semester	Fall
Exam	During the semester
Workload	90h
Weeks	14
Hours	**2 weekly**
Lecture	1 weekly
Exercises	1 weekly
Number of positions	

Summary

Memory corruption and type safety flaws dominate the threat landscape. We will approach current research from three dimensions: sanitization (finding flaws through runtime monitors); fuzzing (testing software automatically); and mitigation (protecting software at runtime).

Content

Unsafe languages like C/C++ are widely used for their great promise of performance. Unfortunately, these languages are prone to a large set of different types of memory and type errors that allow the exploitation of several attack vectors such as code reuse, privilege escalation, or information leaks.
On a high level memory and type safety (and type safety) would solve all these problems. Safe languages can (somewhat) cheaply enforce these properties.
Unfortunately, these guarantees come at a high cost if retrofitted onto existing languages.
When working with unsafe languages, three fundamental approaches exist to protect against software flaws: formal verification (proving the absence of bugs), software testing (finding bugs), and mitigation (protecting against the exploitation of bugs). In this seminar, we will primarily focus on the latter two approaches. Formal verification, while giving strong guarantees, struggles to scale to large software.
This seminar explores three areas: the understanding of attack vectors,approaches to software testing, and mitigation strategies. First you need to understand what kind of software flaws exist in low level software and how those flaws can be exploited.

Learning Prerequisites

Required courses

A security course like COM-301
An operating/systems course like CS-323

Recommended courses

COM-402 Information security and privacy
CS-412 Software security

Learning Outcomes

By the end of the course, the student must be able to:

- Investigate select advanced concepts in software security
- Promote their programming and systems skills in core security topics
- Assess / Evaluate the contributions of a software security research paper

- Investigate software security research papers
- Present a research paper and lead the resulting discussion

Teaching methods

In this seminar course, students will read, prepare, and present recent research papers in the filed of software security. The papers will be discussed in class. The presenter will organize the discussion among their peers and prepare a set of discussion points.

Expected student activities

The students are expected to

- Prepare and hold the presentation of their assigned research paper
- Summarize the paper along with the class discussion after their presentation
- Participate in the presentations and discussions of the other students

Assessment methods

- Presentation : 40%
- Summary/review : 50%
- Class participation : 10%

Resources

Websites

- https://go.epfl.ch/cs510

Moodle Link

- https://go.epfl.ch/CS-510

EPFL

CS-455 **Topics in theoretical computer science**

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - pas donné en 2024-25

Summary

The students gain an in-depth knowledge of several current and emerging areas of theoretical computer science. The course familiarizes them with advanced techniques, and develops an understanding of fundamental questions that underlie some of the key problems of modern computer science.

Content

Examples of topics that will be covered include:

• Laplacians, random walks, graph sparsification: It is possible to compress graphs while approximately preserving their spectral properties (in particular, properties of random walks)? We will cover the main results from the recent influential line of work on spectral sparsification that provides such compression schemes.

• Laplacian system solvers: given a linear system Ax=b, how quickly can we find x? We will cover nearly linear time algorithms for solving Ax=b when A is a symmetric diagonally dominant matrix (a common scenario in practice) that crucially rely on spectral graph sparsification.

• Spectral clustering: given a graph, can we find a partition of the graph into k vertex disjoint parts such that few edges cross from one part to another? This is the fundamental graph clustering problem that arises in many applications. We will cover several results on spectral graph partitioning, where one first embeds vertices of the graph into Euclidean space using the bottom few eigenvectors of the graph Laplacian, and then employs Euclidean clustering primitives to find the partition.

• Local clustering with random walks: Given a very large graph and a seed node in it, can we find a small cut that separates the seed node from the rest of the graph, without reading the entire graph? We will cover local clustering algorithms, which identify such cuts in time roughly proportional to the number of vertices on the small side of the cut, by carefully analyzing distributions of random walks in the graph.

Keywords

spectral graph theory, sparsification, clustering, random walks

Learning Prerequisites

Required courses

Bachelor courses on algorithms and discrete mathematics, mathematical maturity.

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class;
- Analyze approximation quality of spectral graph algorithms;

Teaching methods

Ex cathedra, homeworks, reading

Expected student activities

Attendance at lectures, completing exercises, reading written material

Assessment methods

- Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Others	Electronique forum : Yes

Resources

Bibliography

There is no textbook for the course. Notes will be posted on the course website.

Ressources en bibliothèque

- Randomized Algorithms / Motwani

Moodle Link

- https://go.epfl.ch/CS-455

CS-444 **Virtual reality**

Boulic Ronan

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly
Number of positions	

Summary

The goal of VR is to embed the users in a potentially complex virtual environment while ensuring that they are able to react as if this environment were real. The course provides a human perception-action background and describes the key programming techniques for achieving efficient VR applications

Content

The first lectures focus more on the technical means (hw & sw) for acheiving the hands-on coding sessions:

- Visual display
- Interaction devices and sensors
- Software environment (UNITY3D, programming in C#)

The proportion of more theoretical VR and Neuroscience background increases over the semester:

- Key Human perception abilities, cybersickness, immersion, presence and flow
- Basic 3D interaction techniques: Magic vs Naturalism
- The perception of action
- Haptic interaction
- What makes a virtual human looking alive ?
- VR, cognitive science and true experimental design

The group project focuses on CODING a 3D VR game. So we recommend to rather attend DH-414 "Game design & prototying" in case of lack of coding background as this latter does not require coding skills.

Spring 2024-25 will be the last instance of CS-444 in the study plan.

Keywords

3D interaction, display, sensors, immersion, presence, embodiment

Learning Prerequisites

Required courses

Mastering an Object-Oriented programming language

Recommended courses

(CS-341) Computer graphics

Important concepts to start the course

1) Object Oriented programming lies at the core of the project development in C# with Unity3D. Some programming experience with this approach is compulsory as all students will be assessed on the individual coding of some features of the project.
2) from Computer Graphics:
- perspective transformations
- representation and manipulation of 3D orientation
- 3D modelling hierarchy
- matrix algebra: translation, orientation, composition

Learning Outcomes

By the end of the course, the student must be able to:

- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersions, presence and flow
- Give an example of applications of VR in different industrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions
- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersion, presence and flow
- Give an example of applications of VR in different inustrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions
- Give an example of applications of VR in different industrial sectors

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedra + Hands-on sessions on VR devices in the first half of the semester,
A mini-project in groups of 2-3 persons will have to integrate various components of 3D real-time interaction (in C# within Unity3D). The group will submit their project proposal to the course responsible TAs who will assess whether it meets the key specifications and is original enough. The proposal will include the use of some VR devices that the IIG research group will lend during the mini-project period. The project development will have to be conducted with git.

Expected student activities

exploit citation analysis tools to evaluate a scientific paper
combine 3D interaction components to produce an original 3D experience
experiment the hands-on practical work in the lab
synthesize the knowledge acquired in course and hands-on in the theoretical oral and the project oral

Assessment methods

Scientific paper study : summary of contributions and citation analysis (around week6 of the semester)
Theoretical oral exam (last week of the semester)
Project assessment through code repository, report and oral exam around the end of the semester

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Course notes will be updated and made available after each course, with links to key sites and on-line documents
- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2017. 3D User Interfaces: Theory and Practice. Second edition, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O'Reilly 2015

Ressources en bibliothèque

- 3D User Interfaces / Bowman
- Learning Virtual Reality / Parisi
- The VR book / Jerald

Notes/Handbook

pdf of slides are made visible after the ex-cathedra courses

Websites

- http://www.thevrbook.net/
- http://gitlab.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-444