Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

La Giunta di CCL-LM in Informatica è convocata per il giorno:
mercoledì 11 ottobre 2023 ore 14:00 in modalità online
(e aggiornamento della seduta al giorno 18 ottobre 2023 ore 16.30)
Collegamento alla riunione:
Giunta di CCL-LM
https://unito.webex.com/unito/j.php?MTID=mdbe8967504a50e5db2a30ff6ca229da2
Numero riunione: 2786 786 9610 - Password: UPe5xByTr32

per trattare il seguente Ordine del Giorno:

1. Comunicazioni
2. Approvazione verbale seduta precedente
3. Provvedimenti per la didattica
   3.1 Prima analisi delle proposte di apertura di nuovi insegnamenti L31 e LM18
4. Varie ed eventuali

La Presidente della Giunta di CCL-LM
(prof.ssa Liliana Ardissono)

ELENCO DEI COMPONENTI della Giunta di CCL-LM in Informatica:
Ardissono Liliana, Cardone Felice, Esposito Roberto, Gaeta Rossano, Petrone Giovanna, Pozzato Gian Luca, Sapino Maria Luisa, Sirovich Roberta, Sproston Jeremy James

IN CONGEDO: Pensa Ruggero Gaetano (*che interviene alla riunione in qualità di ospite*)

SONO PRESENTI: Ardissono Liliana, Cardone Felice, Esposito Roberto, Gaeta Rossano, Petrone Giovanna, Sirovich Roberta

ASSENTI GIUSTIFICATI: Pozzato Gian Luca, Sapino Maria Luisa, Sproston Jeremy James

OSPITI:
Matteo Baldoni, Enrico Bini, Susanna Donatelli, Idilio Drago, Paola Gatti, Marco Grangetto, Maurizio Lucenteforte, Matteo Sereno

La seduta ha inizio alle ore 14:00.

1. Comunicazioni

Nessuna comunicazione.

2. Approvazione verbale seduta precedente

Non ci sono verbali da approvare.

3. Provvedimenti per la didattica

3.1 Prima analisi delle proposte di apertura di nuovi insegnamenti L31 e LM18

Durante il mese di settembre 2023, la Presidente dei CdS in Informatica ha raccolto un elenco di proposte di apertura di nuovi insegnamenti da considerare nell'ambito dell'aggiornamento dell'offerta formativa dei CdS. Nel seguito si riportano i dati principali delle proposte e l'ipotesi di loro collocazione nei corsi di laurea e laurea magistrale. Le bozze di syllabus degli insegnamenti proposti si trovano nell'**Allegato del Verbale della Giunta di CCL-LM dell'11 ottobre 2023.**

Proposte per la LM18

- **Advanced cryptography**
  - LM18
  - 6 CFU
  - INF/01 - INFORMATICA
  - TAF - Type of Activity: B - caratterizzante
- **Blockchain, sistemi distribuiti e decentralizzati**
  - LM18
  - 9 CFU (6 di teoria + 3 laboratorio) - (i docenti propongono anche di rendere il corso fruibile alla triennale, offrendo per la L31 una versione ridotta dell'insegnamento, da 6 CFU (4 di teoria + 2 laboratorio)).
  - INF/01 - INFORMATICA
  - Type of Activity: a scelta
- **Computazione Quantistica**
  - LM18
  - 6 CFU
- ~~**Etica e società**~~ **Etica e l'impatto sociale dell'IA / Ethics and the social impact of AI**
  - LM18
  - 6 CFU - Numero di ore - Number of hours: 48 (in aula)
  - INF/01 - informatica
  - TAF - Type of Activity: B - caratterizzante
- ~~**Programmazione non lineare**~~ **Algoritmi per ottimizzazione non lineare**
  - LM18
  - 6 CFU (48 ore)
  - MAT/09
  - TAF - Type of Activity: C (affini e interdisciplinari)
- **Apprendimento Automatico Responsabile e Affidabile / Responsible & Trustworthy Machine Learning**
  - LM18
  - 6 CFU
  - INF/01 - INFORMATICA

    - TAF - Type of Activity: B - caratterizzante
- **Security analytics**
    - LM18
    - 6 CFU
    - INF/01 - INFORMATICA
    - TAF - Type of Activity: B - caratterizzante
- Sdoppiare l'insegnamento Sicurezza delle reti e dei sistemi (6 CFU) in:
    - **Sicurezza delle reti / Network security**
        - LM18
        - 6 CFU
        - INF/01 - INFORMATICA
        - TAF - Type of Activity: B - caratterizzante
    - **Sicurezza dei sistemi / System security**
        - LM18
        - 6 CFU
        - INF/01 - INFORMATICA
        - TAF - Type of Activity: B - caratterizzante
- Sdoppiare l'insegnamento Tecnologie del linguaggio naturale (9 CFU) in:
    - **Trattamento del linguaggio naturale (1)**
        - LM18
        - 6 CFU (48 ore)
        - mantiene molto dell'attuale TLN
    - **Trattamento Avanzato del Linguaggio Naturale con Deep Learning e Large Language Models**
        - LM18
        - 6 CFU (48 ore)
        - focalizzato su tecniche innovative

Discussione

Donatelli espone il contenuto della mail mandata alla presidente della giunta di CCL-LM e che si riporta di seguito nelle sue parti essenziali:

Nella discussione corrente su sistemi, per la magistrale, a parte piccoli aggiustamenti, si è rilevata l'assenza di un corso di cloud che possa fare da perfetto complemento "architetturale/sistemistico" al corso di TASS, quindi non solo studenti capaci di programmare per il cloud, ma capaci di gestire sistemi di high performance computer, con argomenti quali virtualizzazione, orchestrazione, file system distribuiti, etc. Sulla necessità di queste competenze nei nostri studenti Aldinucci, che ha molti contatti con industrie per via del Centro Nazionale, mi ha detto che sono competenze sempre più richieste e comunque (mio parere) per un informatico conoscere a fondo gli strumenti che usa è un punto distintivo. Le competenze per coprire questo corso in dipartimento ci sono (direi almeno Aldinucci, Birke, Colonnelli, ma credo anche diversi altri).

Ardissono fa notare che nel mese di novembre bisognerà procedere alla definizione degli obiettivi formativi dei nuovi corsi proposti.

Drago illustra le proposte avanzate al fine di dotare la laurea magistrale di un programma più moderno circa argomenti di sicurezza.

Ardissono, alla luce dell'ampio numero di proposte di nuove insegnamenti, chiede quali siano quelle a cui dare precedenza.

Gli insegnamenti che secondo Sereno dovrebbero avere una maggiore priorità sono:

- lo sdoppiamento di Sicurezza delle reti e dei sistemi;
- insegnamento di crittografia avanzata.

Baldoni fa notare che bisogna inquadrare queste proposte nel contesto della struttura globale dei CdS. Una possibilità potrebbe essere di inserire gli insegnamenti che sdoppiano Sicurezza delle reti e dei sistemi nel blocco 5 (dove è ora collocato l'insegnamento). Il corso di crittografia potrebbe invece essere inserito nel blocco 4. Simile discorso si può fare per l'insegnamento di blockchain, qui le alternative sono: inserirlo nel blocco 3, oppure rimandare una decisione al prossimo anno accademico, quando si potrà parlare di una ristrutturazione un po' più completa.

Baldoni suggerisce di aggiungere l'insegnamento di blockchain attualmente denotato come insegnamento da 6 CFU tra i crediti liberi della L31. Sereno fa notare che probabilmente è meglio ripensare i contenuti in modo da specializzarli per i rispettivi CdL (quindi un insegnamento per la triennale un po' meno approfondito, uno per la magistrale con i contenuti approfonditi).

> Il docente che propone gli insegnamenti (Claudio Schifanella) evidenzia l'importanza strategica del trattare questo argomento nei CdS in informatica e la modularità della proposta, in cui si separano i contenuti più avanzati da quelli più di base.
>
> La Giunta di CCL-LM conferma che si potrebbe istituire un insegnamento da 6 CFU con i contenuti di base per la L31 (TAF a scelta) e un insegnamento da 9 CFU per la LM18, mutuando i 6 CFU della triennale come parte dei 9 CFU della magistrale. Fa però notare che gli studenti che hanno superato l'esame da 6 CFU della triennale non potranno inserire l'insegnamento della magistrale nel loro piano di studio. Pertanto, suggerisce di ipotizzare l'istituzione dell'insegnamento per la magistrale da 12 CFU (6 che coincidono con l'insegnamento della triennale e 6 avanzati).

Esposito si domanda se l'insegnamento di Quantum Computing sia correttamente dimensionato. L'argomento è complesso e lontano dai paradigmi a cui le studentesse e gli studenti sono abituate/i. Solo 6 crediti sembrano pochi. Ardissono si domanda inoltre come questo insegnamento si contestualizzi in un quadro in cui Fisica e il Politecnico di Torino aprono percorsi dedicati a questi argomenti. Cardone fa notare che ci sono testi consolidati su questi argomenti e che gli sembra quindi che il pacchetto sia ben confezionato. Viene proposto di chiedere a Paolini e Roversi dove collocherebbero l'insegnamento e quali altri insegnamenti del proprio carico didattico lascerebbero in caso dovessero tenere il nuovo insegnamento.

Paolini risponde alle osservazioni della Giunta del CCS come segue.
In Unito l'unico percorso orientato al quantum computing è il "Corso di laurea magistrale Interateneo in Fisica dei sistemi complessi" che fornisce una preparazione specialistica rivolta allo studio e alla modellazione di sistemi e fenomeni complessi naturali e antropici, in particolare nell'ambito della fisica dei fluidi (turbolenza e dispersione), dell'econofisica, della bioinformatica e neurofisica, della computazione e informazione quantistica. Questa è una laurea interateneo e offre solo due insegnamenti propriamente quantistici: Introduzione alla computazione quantistica (6 CFU), Introduzione all'informazione quantistica (6 CFU).
L'insegnamento si distingue dall'insegnamento "Introduzione alla computazione quantistica" proposto nella Magistrale di "Fisica dei sistemi complessi" in almeno 3 punti: (i) l'attenzione posta verso la matematica necessaria alla comprensione di questa forma di computazione (che per studenti di sistemi complessi è ampiamente routine mentre nell'insegnamento per informatica sarebbe sviluppato ex-novo); (ii) l'attenzione rivolta ai sistemi di sviluppo software (assente nel corso a Fisica); (iii) l'apertura verso tutte le tematiche informatiche (algoritmi, sicurezza, comunicazioni, architetture, IA, ...). Il corso offerto darà per scontate solo le minime nozioni di calcolabilita/complessita (contrariamente a quello di Fisica) ragionevolmente già acquisite dai nostri studenti. L'impatto del corso sulla catena formativa mi sembra ampio: (i) potenziale per attirare nuovi studenti, (ii) connessione del tema con molte tematiche negli indirizzi della magistrale; (iii) possibilità di affrontare tesi su argomenti inerenti il proprio indirizzo di laurea nelle prospettive aperte dalla computazione quantistica (con i docenti competenti sul tema).

Al Politecnico viene offerta la laurea in QUANTUM ENGINEERING che forma laureate e laureati con una preparazione multidisciplinare che comprenda le competenze matematiche, fisiche, elettroniche e informatiche necessarie per una efficace applicazione delle tecnologie quantistiche in tre principali ambiti applicativi: comunicazioni, informatica e sensoristica.
Per quanto riguarda il carico didattico associato al nuovo insegnamento, Paolini si propone per la sua copertura ed è disponibile a tenerlo extra-carico almeno per un anno.

Per l'insegnamento di Etica e Società, si nota che i contenuti ora proposti non ne chiariscono bene il contenuto. Anche il titolo dovrebbe essere cambiato per renderlo più specifico. La collocazione come insegnamento caratterizzante non sembra essere giustificata alla luce dei contenuti descritti nella bozza di syllabus, probabilmente sarebbe meglio fosse collocato tra i corsi a scelta. Infine il settore scientifico disciplinare INF/01 non sembra essere completamente giustificato dai contenuti elencati, anche se il punto di vista filosofico sembra essere particolarmente importante e da preservare nel programma dell'insegnamento. Baldoni fa notare che anche queste scelte hanno impatto sulla collocazione dell'insegnamento. Sarebbe inoltre necessario chiarire a che livello di dettaglio vengono affrontati i contenuti del corso. Al momento sembra che gli argomenti siano trattati a livello molto generale, al fine di evitare un ri-etichettamento come non INF/01 sarebbe utile specializzare questi argomenti affrontando anche questioni tecniche specifiche, ritagliando maggiormente l'insegnamento sulla figura dell'informatico.

Il docente che propone l'insegnamento ha fornito una nuova bozza di syllabus che chiarisce i punti discussi in sede di riunione della Giunta. In particolare, il syllabus è maggiormente focalizzato sulle problematiche di carattere informatico, e sull'Intelligenza Artificiale. Il docente sviluppa un confronto con insegnamenti simili erogati in altre università, che può essere eliminato in una successiva versione del syllabus.

Per "Programmazione non lineare" (proposto da Grosso) si suggerisce di cambiare il titolo: l'insegnamento potrebbe essere importante per fornire le basi per altri insegnamenti (ad es. per apprendimento automatico e deep learning); l'attuale titolo probabilmente non aiuterebbe le studentesse e gli studenti a cogliere questo aspetto.

Il docente proponente (Grosso) ha aggiornato il titolo dell'insegnamento: Algoritmi per ottimizzazione non lineare.

Lo sdoppiamento dell'insegnamento di TLN sembra porre problematiche. Dal punto di vista del contenuto, l'insegnamento avanzato di TLN (TLN-2) ha contenuti che sono già parte di altri insegnamenti (ad es. i transformers sono insegnati in Reti Neurali). Quindi, va valutato come mettere a fattor comune gli argomenti. Si nota anche che il passaggio da un insegnamento di 9 CFU a due insegnamenti da 6 CFU implicherebbe un passaggio in un altro blocco di corsi.

Dopo aver interagito con gli attuali docenti dell'Insegnamento TLN, la Giunta di CCL-LM ritiene che la proposta di sdoppiamento dell'insegnamento richieda un approfondimento ulteriore.

Proposte per la L31

- ~~**Problem solving e programmazione competitiva**~~ **Problem solving avanzato**
  - L31
  - 3 CFU (ore aula: 24) [eventualmente aumentabili a 6 CFU (48 ore) con opportuna estensione della proposta]
  - INF/01 - informatica
  - crediti liberi
- Si veda **Blockchain, sistemi distribuiti e decentralizzati**

Per quanto riguarda il corso di "Problem solving e programmazione competitiva", si nota che il titolo potrebbe essere frainteso, si suggerisce di trovare una formulazione che escluda l'idea di competizione. Un possibile suggerimento è "Advanced problem solving", peraltro usato in altri CdS.

I docenti proponenti (Audrito e Amparore) hanno recepito il suggerimento e rinominato l'insegnamento e ridotto l'accento sulla competizione nella descrizione dell'insegnamento.

Per i commenti su Blockchain, sistemi distribuiti e decentralizzati vedere sopra.

4. Varie ed eventuali

4.1 Contributo aggiuntivo Mobilità Erasmus

Interviene Bini, ospite della Giunta di CCL-LM, in qualità di nuovo presidente della Commissione Erasmus e Internazionalizzazione, subentrato a Pensa attualmente in congedo.

Bini ricorda che la Commissione ogni anno chiede al Dipartimento di stanziare 3000 Euro come contributo integrativo erogato a studentesse e studenti rientrati dal periodo di mobilità Erasmus. Tale contributo viene diviso in 30 mensilità aggiuntive da 100 Euro che vengono poi distribuite fra le richiedenti e i richiedenti rientrati dal periodo della mobilità. Il contributo aggiuntivo integra la borsa Erasmus standard di circa 300 Euro mensili.

Come evidenziato anche da Pensa nel suo messaggio di passaggio di consegne, il numero di partenze effettive è significativamente inferiore al numero di vincitrici e vincitori di borsa. E' del parere che una delle possibili cause di questo fenomeno potrebbe essere anche il fatto che la borsa Erasmus (anche se integrata) sia poco cospicua. Inoltre, stante il budget, non si riesce ad assegnare il contributo aggiuntivo a tutti gli aventi diritto, ma è necessaria una selezione (basata su una combinazione di CFU sostenuti all'estero, votazione, ISEE).

In sintesi, la Commissione vorrebbe chiedere un maggiore stanziamento per questo scopo, che vada sia ad allargare la platea dei beneficiari del contributo aggiuntivo, che l'entità del contributo. Investire sulle nostre studentesse e sui nostri studenti, sulla loro possibilità` di sviluppo culturale e di formazione, sulla loro apertura verso il mondo, pare davvero uno dei modi migliori di investire le nostre risorse.

Consultata la Direzione del Dipartimento, che si è detta d'accordo, la stessa ha coinvolto la Giunta di CCL-LM per un parere.

Di seguito gli appunti che Bini ha condiviso con Direzione e Giunta di CDD e di CCL-LM:

Si premette che:

- La borsa Erasmus+ erogata dall'Ateneo ammonta alla somma delle due voci (usando l'anno 2023/24 come riferimento):
  - voce A: 250/300/350 Euro/mese
  - voce B: 0--450 Euro/mese (0 se ISEE>50K, 450 se ISEE <13K, interpolato altrimenti. Si veda Tabella 4 del bando) erogato solo se riconosciuti in carriera almeno 2 CFU/mese

mobilità. Estratto dal bando: "Per l'erogazione del contributo della (VOCE B), si darà precedenza alle studentesse e agli studenti con ISEE inferiore, fino ad esaurimento del budget disponibile".
    - (source: https://www.unito.it/internazionalita/studiare-e-lavorare-allestero/erasmus/erasmus-studio/bando-erasmus-studio)
- Il numero di partenze effettive e` significativamente inferiore al numero degli assegnatari della borsa.
- Al fine di premiare i meritevoli e ritenendo che l'esiguità della borsa possa essere uno dei motivi che concorra alle mancate partenze, il dipartimento stanzia annualmente 3000 Euro da distribuire come contributo integrativo erogato agli studenti rientrati dal periodo di studi Erasmus. Tale contributo viene diviso in 30 mensilità aggiuntive da 100 Euro che vengono poi distribuite fra gli studenti rientrati dal periodo della borsa.
- Il contributo aggiuntivo erogato dal nostro dipartimento pare notevolmente sotto-dimensionato rispetto agli altri dipartimenti. Prendendo gli esiti del bando 2023/24 come riferimento
    - Dip. Fisica
        - contributo aggiuntivo in bando 2023/24: 7000 Euro
        - numero assegnatari 2023/24: 15
        - contributo/assegnatari: 467
    - Dip. Scienze Terra
        - contributo aggiuntivo in bando 2023/24: 3500 Euro
        - numero assegnatari 2023/24: 11
        - contributo/assegnatari: 318
    - Dip. Informatica
        - contributo aggiuntivo in bando 2023/24: 3000 Euro
        - numero assegnatari 2023/24: 25
        - contributo/assegnatari: 120 (circa **3-4 volte inferiore agli altri dipartimenti**)
- Secondo le regole dell'ultimo bando il contributo dipartimentale viene ripartito secondo i seguenti criteri e modalità:
    1. il contributo è destinato agli studenti che, **al rientro**, hanno riconosciuti in carriera almeno 3 CFU/mese
    2. il contributo viene calcolato sulla base del numero di giorni GG di mobilità effettivamente svolti sulla base della formula: GG/30*100 Euro
    3. il contributo viene erogato entro la fine dell'anno solare di fine mobilità
    4. qualora il budget messo a disposizione dal Dipartimento non sia sufficiente a coprire tutte le necessità, si procederà a stilare una graduatoria basata su (nell'ordine):
        a. media pesata degli esami (alla convalida)
        b. crediti superati
        c. ISEE
        d. età anagrafica, con privilegio dei candidati più giovani
- Ritenendo che l'investimento sui nostri studenti, sulla loro possibilità di sviluppo culturale e di formazione, sulla loro apertura verso il mondo, sia uno dei modi migliori di investire le nostre risorse.

La Commissione Erasmus chiede che
- venga stanziato un contributo dipartimentale maggiore pari a 10K Euro, in modo da allinearlo agli altri dipartimenti e tale da permettere di erogare un contributo maggiore e ad una platea più ampia di beneficiari.

Infine si ricorda che è necessario prendere una decisione entro il CdD del 18 Ottobre per poter scrivere nel bando Erasmus+ 2024/25 queste informazioni.

Ardissono sostiene la proposta. L'Erasmus è un investimento che ha impatto sia sulla formazione delle studentesse e degli studenti, sia sulla valutazione del Dipartimento. Ardissono chiede un parere agli altri membri della Giunta.

Donatelli propone e Bini accoglie la proposta che la commissione Erasmus valuti una proposta dettagliata in cui si cerchi di premiare gli studenti più meritevoli.

Pensa si dice contrario alla proposta di dare una quota maggiore agli studenti più in alto in graduatoria, propone invece di destinare le risorse aggiuntive al fine di aumentare il numero di borse. Ardissono concorda con Pensa, è meglio dare un po' di più a tutti e aumentare il numero di borse.

Dopo avere ascoltato le opinioni di Pensa e Ardissono, Bini si dice maggiormente convinto dall'ipotesi di destinare eventuali risorse aggiuntive come da loro suggerito.

La Giunta unanime appoggia la proposta che verrà presentata al Consiglio di Dipartimento del 18 ottobre p.v..

Esauriti gli argomenti del giorno, la Giunta conclude i propri lavori alle ore 18:00, approvando di riunirsi il giorno 18 ottobre 2023 ore 16.30 per aggiornamenti.

Il punto 4.1 del presente verbale viene **approvato seduta stante** dalla Giunta del CCL-LM.

La Presidente
Prof.ssa Liliana Ardissono

Il Segretario verbalizzante
Prof. Roberto Esposito

Allegato n. 1 del Verbale della Giunta di CCL-LM dell'11 ottobre 2023

Descrizioni complete delle bozze di proposta di nuovi insegnamenti.

Le descrizioni sono bozze di syllabus volte a inquadrare l'argomento e la collocazione degli insegnamenti nei piani di studio.

********** LM18 **********

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Elementi avanzati di Crittografia

Numero di CFU - 6
SSD attività didattica - INFORMATICA
TAF - Type of Activity: B - caratterizzante

Informazioni generali
Docenti – da definire (disponibilità di L Terracini e M. Sereno)
Erogazione: Tradizionale
Numero di ore - 48 (attività laboratoriale eventuale)

Prerequisiti

- Competenze attese in ingresso (richieste all'inizio del corso)
  - o I contenuti di base di algebra e matematica discreta dei corsi di matematica della Laurea Triennale
  - o Conoscenza di base di reti, sicurezza e sistemi operativi.
- Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso") o Reti di Elaboratori o Reti I (triennale)
  - o Sistemi Operativi (triennale)
  - o Sicurezza (triennale)

Obiettivi formativi

Il corso mira a rivisitare ed approfondire argomenti fondamentali della crittografia in un quadro matematico più preciso fornendo gli strumenti matematici e algoritmici per poter definire in termini matematico/algoritmici cosa si intende per sistemi crittografici sicuri e quali sono le relazioni/implicazioni ed impatto che questi sistemi crittografici sicuri hanno.

Inoltre, il corso tratterà anche argomenti di crittografia avanzata (ad esempio zero knowledge proofs, elliptic curves cryptography, crittografia omomorfa, crittografia post-quantum).

Risultati dell'apprendimento attesi

Le principali conoscenze acquisite saranno:

- Familiarità con l'aritmetica dei campi finiti
- Familiarità con le basi di teoria algoritmica dei numeri e delle curve ellittiche.
- Dimestichezza con i concetti di crittosistema, crittografia a chiave pubblica, firma digitale, autenticazione, crittografia simmetrica.
- Gli studenti acquisiranno le conoscenze teoriche e pratiche per collaborare in attività di ricerca, progettazione e implementazione in campo crittografico. In particolare sapranno
  - o valutare la sicurezza di un crittosistema simmetrico e asimmetrici,
  - o valutare la difficoltà computazionale di problemi di teoria dei numeri e la

loro applicabilità a tecniche crittografiche,
o definire i parametri di un'infrastruttura di crittografia a chiave pubblica sicura

Programma preliminare

1. Preliminari di matematica: algebra dei polinomi, campi finiti, complementi di aritmetica modulare
2. Crittografia simmetrica: block ciphers (AES)
3. Qualche tecnica di crittanalisi simmetrica (crittoanalisi lineare e differenziale, side channel attack)
4. Curve ellittiche e crittosistemi EC based
5. Firme digitali (DSA e ECDSA)
6. Zero knowledge proof
7. Generatori pseudo casuali (Linear feedback registers, Blum Blum Shub) 8. Sviluppi recenti in crittografia asimmetrica: crittografia omomorfa
9. Cenni su crittografia post quantum (In particolare lattices e NTRU)

Modalità di verifica dell'apprendimento

La valutazione dell'esame si comporrà di (i) la consegna delle soluzioni degli esercizi proposti durante il corso; (ii) (probabilmente ma questo è un aspetto da definire, una prova pratica di crittanalisi o una relazione tecnica su un argomento a scelta) (iii) una prova orale di teoria. Tutte le parti dell'esame devono essere superate e contribuiscono a determinare il voto finale secondo una proporzione pre-determinata. La consegna degli esercizi e il superamento della prova pratica sono condizioni necessarie per accedere all'esame orale.

Blockchain, sistemi distribuiti e decentralizzati

laurea magistrale

Numero di CFU - Credits: 9 (6 di teoria + 3 laboratorio)
(Intendiamo anche rendere il corso fruibile alla triennale, offrendo una versione ridotta del corso, in modo che sia tarato su 6CFU (4 di teeoria + 2 laboratorio)) SSD attività didattica - Scientific Sector of Activity: INF/01 - INFORMATICA Type of Activity: a scelta

Informazioni generali / General Information

Docenti proponenti: Prof. Andrea Bracciali e Prof. Claudio Schifanella

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Erogazione - Teaching Modality: Tradizionale
Lingua - Language: Italiano
Frequenza - Attendance: Facoltativa ma consigliata
Numero di ore - Number of hours: 78 (48 in aula / 30 in laboratorio)

1. Prerequisiti

- **Competenze attese in ingresso (richieste all'inizio del corso)**
  Il corso è per quanto possibile auto-contenuto e i concetti specifici necessari vengono reintrodotti ove necessario. Si assumono le seguenti conoscenze informatiche di base:
  - Concetti di base su algoritmi e programmazione
  - Conoscenza di base sul funzionamento delle reti di calcolatori
- **Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso")**
  - Algoritmi (triennale)
  - Reti I (triennale)

2. Obiettivi formativi

Il principale obiettivo del corso è sviluppare una solida comprensione delle tecnologie blockchain nel contesto dei sistemi distribuiti. Queste tecnologie, sviluppate recentemente, hanno introdotto il nuovo modello di calcolo decentralizzato che si basa su contributi e idee provenienti da differenti discipline, quali crittografia, economia, informatica, … La comprensione di questo modello richiede competenze avanzate che il corso si propone di fornire allo studente, con un approccio sia teorico che pratico, con una consistente parte di laboratorio e sperimentazione con tecnologie alcune delle quali attualmente in corso di

[1] Vorremmo offrire un corso da 9 CFU alla magistrale ed una sua versione da 6 CFU alla triennale

sviluppo, con l'obiettivo di formare esperti che possano sia partecipare allo sviluppo della tecnologia, per esempio con percorsi di ricerca, sia trovare spazio nel selettivo mercato del lavoro nel settore, che è attualmente in forte domanda.

Partendo dalla teoria dei sistemi distribuiti, si introducono le motivazioni per la computazione decentralizzata e i problemi che essa vuole risolvere, con particolare riferimento al noto problema del consenso distribuito e sue applicazioni. Il framework di Bitcoin, che ha permesso lo sviluppo della tecnologia, verrà analizzato in dettaglio come caso paradigmatico, fornendo allo studente gli strumenti per un'analisi critica di innovazione e limitazioni.

In seguito, viene studiata la programmazione decentralizzata per mezzo degli *smart contracts,* sia su Bitcoin che su Ethereum, una blockchain mainstream per questo tipo di applicazioni. Lo studente così acquisisce sia la capacità di sviluppare applicazioni decentralizzate, che di analizzare il loro funzionamento e progettare soluzioni corrette.

La parte finale del corso offre allo studente una visione sugli sviluppi più recenti della tecnologia in evoluzione. In prima istanza viene presentata una blockchain con un modello di consenso efficiente e un ambiente di programmazione per smart contracts avanzato. La possibilità di variare gli argomenti avanzati permette di offrire agli studenti competenze aggiornate con lo sviluppo della tecnologia.

Il corso integra teoria e pratica tramite un approccio *learning by doing,* supportato da un sostanziale numero di ore di laboratorio, che permetterà agli studenti di imparare a progettare e implementare applicazioni software decentralizzate. Verranno organizzati seminarti tenuti da rappresentanti dell'industria e del mondo della ricerca che favoriranno futuri rapporti con gli studenti e con il dipartimento.

3. Risultati dell'apprendimento attesi

Alla fine di questo insegnamento, le/gli studentesse/studenti saranno in grado di:

- Conoscere le caratteristiche e le proprietà delle architetture decentralizzate
- Comprendere il ruolo e le caratteristiche dei differenti algoritmi di consenso
- Saper valutare criticamente l'utilizzo di tecnologie decentralizzate e i principi generali della loro progettazione
- Comprendere il funzionamento della rete bitcoin e del suo linguaggio di scripting
- Conoscere le caratteristiche delle piattaforme blockchain basate su smart contract, con particolare riferimento alla piattaforma Ethereum
- Avere una visione dei problemi aperti e gli sviluppi più interessanti nel contesto dei sistemi decentralizzati e tecnologie blockchain – come per esempio Algorand
- Progettare e implementare applicazioni decentralizzate

A. Programma 78h (48+30) (per la LM18)

- Introduction to distributed systems **4T**
  - Distributed programming execution model
  - Distributed algorithms, examples
- Decentralised technologies **3T**
  - Defining decentralisation and its motivations
  - Use case: internet money
- Distributed consensus **2T**
  - Summary and classical impossibility results
- The Blockchain decentralised consensus: **14T**
  - Cryptography: a recap
    - Main primitives
    - Attacks and complexity notions
  - Bitcoin's Proof-of-Work
  - Incentives and tokenomics
  - Transactions and transaction types
  - the SCRIPT verification language
  - limitations

        - the blockchain trilemma
        - governance and decentralisation
    - Other Proof-of
        - Ethereum's Proof-of-Stake
- LAB: **6L** ○ Interaction with the Bitcoin testnet
    - Verifying transactions
- Smart contracts **10T**
    - Definitions and decentralised execution model
    - Introduction to Ethereum's EVM: accounts, smart contracts and gas
    - Programming smart contracts: the Solidity language
    - ERC standards and libraries, tokenization, non-fungible tokens
- Formal verification of smart contracts **4T**
- LAB: Ethereum smart contracts **16L**
        - Structure of a Decentralized Application (dAPP)
            - Programming a dApp: frontend and interaction with smart contracts
            - Decentralized storage: IPFS
            - UX/UI design: best practices
    - Development tools, and execution environments
    - Development of simple smart programs
        - Decentralised games, auctions, lotteries, …
        - More advanced applications and their evaluation.
    - Guided (group) project
- Advanced topics[2]
    - The Algorand blockchain **8T**
        - The Algorand consensus and the blockchain trilemma
        - Algorand's Smart Contracts, an introduction
    - LAB: Programming the Algorand blockchain **8L**

[2] Questa parte del modulo si concentra su argomenti avanzati che possono cambiare nel corso degli anni, lasciando una certa flessibilità nell'abbracciare gli ultimi sviluppi dell'area in rapida evoluzione dell'informatica decentralizzata. Può includere nuovi modelli, ad esempio Algorand per iniziare, o applicazioni, ad esempio finanza decentralizzata, NFT, giochi, applicazioni sociali, ... Gli argomenti avanzati offrono anche l'opportunità di stabilire collegamenti industriali.

5. Modalità di verifica dell'apprendimento

La valutazione dello studente si basa su (i) un progetto pratico guidato da concordare con il docente, (ii) una presentazione orale del progetto svolto con discussione, e (ii) una prova orale di teoria sul contenuto del corso. Tutte le componenti della valutazione devono essere superate e contribuiscono a determinare il voto finale secondo una proporzione

predeterminata. La consegna del progetto è condizione necessaria per accedere agli esami orali.

6. Modalità di insegnamento

L'insegnamento è diviso in una parte di teoria e una di laboratorio. La parte di teoria prevede 48 ore di lezioni frontali. Per la parte di laboratorio sono previste 30 ore di attività in laboratorio in cui si fa pratica con implementazioni di blockchain e ambienti di sviluppo e si svolgono esercizi di programmazione di applicazioni decentralizzate basate su *smart contract*. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non-frequentanti. La partecipazione ad almeno una sessione di presentazione dei progetti è fortemente consigliata.

7. Attività di supporto

Il materiale didattico di supporto (e.g., lucidi, link, codice degli esercizi) è disponibile presso il supporto on-line ai corsi I-learn. I temi dei progetti e degli esercizi saranno resi disponibili online durante il corso sullo stesso sito. https://informatica.i-learn.unito.it/. Ove possibile, verranno organizzati seminari industriali e su temi di ricerca e problemi aperti.

8. Testi consigliati e bibliografia

La maggior parte del materiale didattico sarà distribuito tramite slides del corso, con riferimenti a pubblicazioni scientifiche ove opportuno – da usare come letture di approfondimento. Il testo

> Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller,
> Steven Goldfeder
> Bitcoin and Cryptocurrency Technologies - A Comprehensive
> Introduction Princeton University Press

offre una comprensiva introduzione alla tecnologia blockchain per la maggior parte del contenuto teorico del corso, con utili esercizi suggeriti.
Riferimenti a manuali per l'uso di programmi, linguaggi e ambienti di sviluppo saranno comunicati agli studenti.

A. Programma dell'insegnamento triennale da 6 CFU (per la L31)

52h (32+20)

L'argomento e la struttura del corso si prestano a mutuare a costo zero un corso facoltativo per la laurea triennale di sei crediti, ottenuto riducendo alcuni degli argomenti più tecnici della teoria e la parte di argomenti avanzati, con riduzione proporzionale di lezioni frontali e laboratori. Le parti ridotte sono in grigio nella seguente lista di argomenti.
Nel seguito, le parti in grigio sono escluse dal programma dell'insegnamento.

- Introduction to distributed systems **4T**

- Distributed programming execution model
- Distributed algorithms, examples

- Decentralised technologies **3T**
  - Defining decentralisation and its motivations
  - Use case: internet money
- Distributed consensus **2T**
  - Summary and classical impossibility results
- The Blockchain decentralised consensus: **14T** ○ Cryptography: a recap
    - Main primitives
    - Attacks and complexity notions
  - Bitcoin's Proof-of-Work
  - Incentives and tokenomics
  - Transactions and transaction types
  - the SCRIPT verification language
  - limitations
    - the blockchain trilemma
    - governance and decentralisation
  - Other Proof-of
    - Ethereum's Proof-of-Stake
- LAB: **6L**
  - Interaction with the Bitcoin testnet
  - Verifying transactions
- Smart contracts **10T**
  - Definitions and decentralised execution model
  - Introduction to Ethereum's EVM: accounts, smart contracts and gas
  - Programming smart contracts: the Solidity language
  - ERC standards and libraries, tokenization, non-fungible tokens
- Formal verification of smart contracts **4T**
- LAB: Ethereum smart contracts **14L**
  - Structure of a Decentralized Application (dAPP)
    - Programming a dApp: frontend and interaction with smart contracts
    - Decentralized storage: IPFS
    - UX/UI design: best practices
  - Development tools, and execution environments
  - Development of simple smart programs
    - Decentralised games, auctions, lotteries, …
  - Group project
- Advanced topics

  This part of the module focusses on advanced topics that may change in the years, leaving some flexibility in embracing latest development of the fast-pace evolving area of decentralised computing. May include new models, e.g. Algorand

to start with, or applications, e.g. Decentralised Finance, NFTs, Gaming, Social applications, … Advanced topics also offer the opportunity of establishing industrial links.

- The Algorand blockchain
  - The Algorand consensus and the blockchain trilemma **8T**
  - Algorand's Smart Contracts, an introduction
- LAB: Programming the algorand blockchain **8L**

Computazione quantistica

6 CFU

Magistrale

Prerequisiti

Conoscenza della matematica di base: numeri complessi e trigonometria, algebra di base. Le nozioni matematiche avanzate e necessarie (spazi vettoriali, prodotto tensore etc.) verranno introdotte durante il corso. Conoscenze elementari sui circuiti combinatori e sulla teoria della calcolabilità.

Obiettivi formativi

L'insegnamento si propone di fornire agli studenti una generale comprensione di come la meccanica quantistica possa essere applicata a problemi computazionali. Partendo da concetti di logica classica, si introducono le principali porte logiche a singolo qubit e a due qubit per giungere ad analizzare i principali algoritmi quantistici. Scopo specifico dell'insegnamento è anche quello di fornire agli studenti gli strumenti matematici e fisici adeguati ad affrontare le problematiche discusse e di sviluppare conoscenze relative ai sistemi fisici utilizzati per la realizzazione pratica di un computer quantistico evidenziando i problemi sperimentali ad essi associati.

Risultati dell'apprendimento attesi

Il primo obiettivo formativo del corso e' di apprendere i concetti e i fenomeni principali alla base dei dell'informatica e dei computer quantistici - quali il principio di sovrapposizione degli stati, il qubit, l'entanglement e le porte quantistiche - e capire il funzionamento di alcuni algoritmi quantistici. Parte integrante del corso sono le lezioni pratiche in cui si applicano i concetti imparati scrivendo dei codici per i computer quantistici (IBM-Qiskit, Microsoft-QDK) per i calcolatori quantistici disponibili in rete.

Programma

Panoramica su computazioni non-convenzionali: analogiche, non-deterministiche, probabilistiche, reversibili, quantistiche. Universalità e ripasso discussioni sulla loro complessità computazionale. Universalità e ripasso discussioni sulla loro complessità computazionale.
Introduzione al mondo Quantum: panoramica sulle tecnologie, interference, decoherence, noise and Di Vincenzo criteria.
Qubits e stati puri, sfera di Bloch, superposizione ed entanglment, operatori ed evoluzione, osservabili e misure.
Simulatori e ambienti di sviluppo: da Python a QiSkit, QDK e Q#.
Modelli di computazione a circuiti uniformi. Modelli classici e reversibili e la loro rappresentazione algebrica.
Spazi vettoriali complessi, spazi di Hilbert e loro operatori, stati puri, sfera di Bloch, superposizione, entranglement, osservabili e misurazioni.
Rappresentazione dei dati: codici superdensi e teletrasporto quantistico.
Algoritmi: Deutsch-Josza, Simon, Amplitute Amplification, Grover search, Shor Factoring.
Crittografia quantistica. Protocolli crittografici quantistici Bennett-Brassard (BB84) e Ekert91. Cenni ai protocolli BBM92 e B92.
Cenni agli stati misti ed quantum information theory.
Advanced hardware hints: Nisq and quantum annealing. Riduzioni di problemi intrattabili classici ed applicativi a Modelli QUBO da risolvere tramite Quantum Annealing.

Modalità di insegnamento

Le lezioni sono erogate in maniera tradizionale, ossia frontalmente. Verranno svolti esercizi negli ambienti di sviluppo presentati nel corso (Qiskit e QDK).

Modalità di verifica dell'apprendimento

Esame scritto: esercizi e domande.

Testi consigliati e bibliografia

Introduction to Quantum Computing with Q# and QDK - Filip Wojcieszyn
https://link.springer.com/book/10.1007/978-3-030-99379-5

con integrazioni da:

- Quantum Computing for Computer Scientists - N. Yanofsky, M. Mannucci
  https://www.cambridge.org/core/books/quantum-computing-for-computer-scientists/8AEA723BEE5CC9F5C03FDD4BA850C711
- Quantum Computation and Quantum Information - I. Chuang, M. Nielsen
  https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE#overview
- D-Wave Quantum Leap Software and Hardware Technology
  https://docs.dwavesys.com/docs/latest/handbook_intro.html

Etica e Società (syllabus sostituito con nuova proposta inserita nell'Allegato al verbale della riunione del 18 ottobre 2023)

Ethics and Society

Corso di studio
[008515] Laurea magistrale in informatica
Anno
1° anno, 2° anno
Tipologia
Caratterizzante
Crediti/Valenza
6 CFU - Numero di ore - Number of hours: 48 (in aula)
Crediti percorso 24 CFU
2
SSD attività didattica
INF/01 - informatica
Erogazione
Tradizionale
Lingua
Italiano
Frequenza
Facoltativa
Tipologia esame
Orale
Tipologia unità didattica
corso
Prerequisiti
Nessuna

Insegnamenti propedeutici (forniscono le competenze attese in ingresso):

Nessuno

Obiettivi formativi

L'insegnamento concorre al raggiungimento degli obiettivi formativi specifici del Corso di Laurea Magistrale in Informatica (LM18) fornendo un'introduzione i problemi relativi alla società e all'etica sollevati dai sistemi informativi nella società contemporanea in particolare alla luce dei recenti sviluppi dell'AI generativa. Si propone perciò di fornire le competenze interdisciplinari necessarie a valutare impatto sulla società dello sviluppo di sistemi informatici che rispettino i valori etici fin dalle fasi di ideazione e progettazione.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITÀ DI COMPRENSIONE

Al termine dell'insegnamento si dovranno conoscere:

- Cosa sia l'etica applicata alla tecnologia
- Comprensione degli eventi più recenti legati
- I rischi sollevati dalle tecnologie dell'informazione nelle loro diverse dimensioni: economici, politici, tecnologici, sociali, ecc.
- l'impatto delle tecnologie dell'informazione sulla società contemporanea

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE

Al termine dell'insegnamento si dovrà essere in grado di:

- adottare un approccio etico all'informatica
- valutare l'impatto sulla società di una tecnologia informatica

AUTONOMIA DI GIUDIZIO

Alla fine di questo insegnamento si saprà:

- sfruttare la migliorata consapevolezza della necessità di un approccio etico all'informatica nella ideazione, progettazione e sviluppo dei sistemi informatici.
- interpretare correttamente le comunicazioni relativa alle novità tecnologiche epurando le finalità di marketing che le avvolgono.

ABILITÀ COMUNICATIVE

Al termine dell'insegnamento si sarà in grado di:

- elaborare, in forma scritta e/o orale, le problematiche legate all'etica e all'impatto delle tecnologie informatiche

CAPACITÀ DI APPRENDIMENTO

Al termine dell'insegnamento, si saranno acquisite capacità autonome di apprendimento e di autovalutazione della propria preparazione, atte a intraprendere gli studi successivi con un alto grado di autonomia.

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Programma

Etica e Società

La prima parte dell'insegnamento mira a presentare e discutere i problemi relativi alla società e all'etica sollevati dai sistemi informativi nella società contemporanea, dal possibile abuso dei social network ai rischi nell'uso dei "big data", dall'impatto dell'informatica sulle libertà fondamentali alla diffusione dell'AI generativa. Si considereranno anche le relazioni tra l'IT e le leggi e i regolamenti.

- Cosa è l'Etica e come è applicata al mondo dell'Informatica
- L'economia dei mercanti dell'attenzione
- Il capitalismo della sorveglianza
- I social media e connessioni con la politica
- L'intreccio Silicon Valley – Washington
- Google e la nascita della pubblicità personalizzata
- Il dispossetion cycle
- Economia digitale, competizione e lavoro
- Le tendenze monopolistiche delle imprese dell'ICT
- Il Panopticon: Sorveglianza vs Libertà
- Aspetti giuridici, epistemologici, etici e sociali dei big data
- Guerra cibernetica
- Opportunità e rischi delle tecnologie blockchain
- Neutralità della Rete
- AI generativa, manipolazione e disinformazione
- Irrisolvibilità tecnologica dei problem di bias e categorizzazione
- Problemi etici dell'emotion recognition
- Deep Learning e l'impossibilità dell'explainability
- Lo scandalo Cambridge Analitica e Edward Snowden

Modalità di insegnamento

L'insegnamento è strutturato in 48 ore di didattica frontale che prevedono una forte componente interattiva tra docenti e studentesse/studenti. Le ore sono suddivise in lezioni da 2 ore in base al calendario accademico. Sono previsti seminari tenuti da esperti sugli argomenti trattati. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

Modalità di verifica dell'apprendimento

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove si potranno trovare forum di discussione, materiale didattico, compiti assegnati e approfondimenti sugli argomenti delle lezioni preparati dai docenti. L'esame consiste in un colloquio atto a valutare la comprensione generale del corso e di argomenti specifici concordati con i docenti. Durante il colloquio potrà essere richiesto lo svolgimento di esercizi. La valutazione è in trentesimi.

Testi consigliati e bibliografia: -

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

~~Programmazione non lineare~~ Algoritmi per ottimizzazione non lineare (6 CFU, 48 h)

Collocazione: laurea magistrale, probabilmente primo anno (?).
Le tematiche sono probabilmente di interesse per l'indirizzo intelligenza artificiale ma potenzialmente anche per altri indirizzi.

Obiettivi formativi

Il corso si propone di fornire ai partecipanti conoscenze relative ai principali modelli e metodi per la massimizzazione o minimizzazione di funzioni non lineari, sia in assenza di vincoli che in presenza di vincoli (lineari o no), analizzando sia problematiche classiche consolidate sia problematiche di particolare interesse per l'area in forte sviluppo del machine learning.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITA' di COMPRENSIONE: dimestichezza con la teoria dell'ottimizzazione non lineare nel continuo, nei casi vincolati e non vincolati, dualità, e conoscenza dei principali algoritmi del primo ordine e superiori.

CAPACITA' DI APPLICARE CONOSCENZA E COMPRENSIONE: capacità di valutare la difficoltà di un problema di ottimizzazione non lineare, capacità valutare e selezionare gli opportuni algoritmi per il problema in esame, declinandoli nelle versioni più adeguate al contesto applicativo.

AUTONOMIA DI GIUDIZIO: acquisizione di autonomia di giudizio nel valutare l'adeguatezza di varie tecniche ed algoritmi noti in vari contesti computazionali (dimensione del problema, convessità o meno di funzione obiettivo e vincoli, smoothness delle funzioni coinvolte).

ABILITA' COMUNICATIVE: capacità di discutere ed illustrare i principali algoritmi per ottimizzazione vincolata e non vincolata.

CAPACITA' DI APPRENDIMENTO: acquisizione di autonome di capacità di apprendimento in campo modellistico ed algoritmico.

Programma

- Preliminari, richiami di analisi multivariata (?)
- Ottimizzazione non vincolata
  - condizioni di ottimalità
  - metodi fondamentali
  - line search, ottimizzazione univariata
  - steepest descent e varianti
  - Metodo di Newton
  - Metodi quasi-Newton

    - metodi particolari: algoritmi a batch (stochastic/batch gradient descent),
    - metodi per minimi quadrati (?)
- Ottimizzazione vincolata
    - dualità lagrangiana, teoria dei moltiplicatori di Lagrange
    - Metodi a punto interno
        - caso lineare
        - caso quadratico
        - caso convesso e generale non convesso (cenni?)

Modalità di insegnamento
Lezioni ed esercitazioni in aula o laboratorio.

Modalità di verifica dell'apprendimento
Esame scritto e/o orale, sviluppo di progetti didattici.

Apprendimento Automatico Responsabile e Affidabile / Responsible & Trustworthy Machine Learning

6 CFU (48 ore)

Prerequisiti
Conoscenze elementari di probabilità e statistica, algoritmi, basi di dati, apprendimento automatico e/o reti neurali.
Insegnamenti propedeutici (forniscono le competenze attese in ingresso): Dalla laurea triennale: Sistemi intelligenti, Basi di dati, Algoritmi. Dalla magistrale: Apprendimento Automatico o Reti Neurali e Deep Learning.

Sommario insegnamento

- Obiettivi formativi
- Risultati dell'apprendimento attesi
- Modalità di insegnamento
- Modalità di verifica dell'apprendimento
- Programma
- Testi consigliati e bibliografia
- Strumenti didattici

Obiettivi formativi

L'insegnamento concorre al raggiungimento degli obiettivi formativi specifici del Corso di Laurea Magistrale in Informatica (LM18) fornendo un'introduzione ai principali metodi per la gestione e l'analisi affidabile e responsabile dei dati. L'insegnamento fornisce le

competenze teoriche e pratiche necessarie alla ideazione, progettazione e sviluppo di algoritmi di apprendimento automatico aperti, affidabili, interpretabili e trasparenti, nel rispetto dell'etica e della privacy dell'individuo.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITÀ DI COMPRENSIONE
Al termine dell'insegnamento si dovranno conoscere:

- i rischi per i diritti fondamentali delle persone (come libertà, non discriminazione) sollevati dalla gestione e analisi dei dati e dall'apprendimento di modelli a partire dai dati.
- i principali metodi di protezione dei dati,
- i principali metodi di per la progettazione di algoritmi di apprendimento automatico etici, giusti, affidabili, interpretabili e trasparenti.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE
Al termine dell'insegnamento si dovrà essere in grado di:

- adottare un approccio responsabile alla progettazione dei sistemi intelligenti
- ™¡

AUTONOMIA DI GIUDIZIO
Alla fine di questo insegnamento si saprà:

- analizzare, progettare e sviluppare soluzioni affidabili, eque, trasparenti e rispettose della privacy nei sistemi informatici in semplici casi di studio,
- sfruttare la migliorata consapevolezza della necessità di un approccio etico all'utilizzo dei dati nella ideazione, progettazione e sviluppo dei sistemi intelligenti basati sull'apprendimento automatico.

ABILITÀ COMUNICATIVE
Al termine dell'insegnamento si sarà in grado di:

- elaborare, in forma scritta e/o orale, le problematiche legate all'equità, trasparenza, affidabilità e privacy e le possibili soluzioni in casi di studio pratici.

CAPACITÀ DI APPRENDIMENTO
Al termine dell'insegnamento, si saranno acquisite capacità autonome di apprendimento e di autovalutazione della propria preparazione, atte a intraprendere gli studi successivi con un alto grado di autonomia.

Modalità di insegnamento

L'insegnamento è strutturato in 48 ore di didattica frontale e laboratoriale che prevedono una forte componente interattiva tra docenti e studentesse/studenti. Le ore sono suddivise in lezioni da 2 ore in base al calendario accademico. Sono previsti seminari tenuti da esperti sugli argomenti trattati. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

Modalità di verifica dell'apprendimento

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove si potranno trovare forum di discussione, materiale didattico, compiti assegnati e approfondimenti sugli argomenti delle lezioni preparati dai docenti. L'esame consiste in uno scritto atto a valutare la comprensione generale del corso e di argomenti specifici. Nell'esame scritto potrà essere richiesto lo svolgimento di esercizi. La valutazione è in trentesimi.

Programma

Privacy e Protezione del Dato

La prima parte dell'insegnamento mira ad introdurre i principali metodi per la gestione e l'analisi privata dei dati. Questa parte si concentra sui principali metodi di anonimizzazione e modelli di computazione privacy-preserving proposti, con un'attenzione particolare alla progettazione di algoritmi di machine learning rispettosi della privacy.

1. Il concetto di privacy nei sistemi intelligenti e la sua regolamentazione (2 ore)
2. Attacchi e modelli di privacy nelle basi di dati statistiche (4 ore)
    1. k-anonymity
    2. l-diversity
    3. t-closeness
    4. delta-presence
3. Differential privacy (8 ore)
    1. Definizioni e teoremi base
    2. Definizioni avanzate
    3. Machine Learning differenzialmente privato
    4. Esercitazione: progetto di algoritmi differenzialmente privati
4. Data separation (2 ore)
    1. Secure multiparty computation
    2. Federated learning

Equità e Bias nel Machine Learning

La seconda parte dell'insegnamento è volta ad introdurre i principali metodi per l'individuazione e la riduzione o rimozione dei bias negli algoritmi di apprendimento automatico.

1. Il concetto di ~~algorithmic~~ bias e discriminazione nei sistemi decisionali
2. Definizioni statistiche di fairness
3. Metodi di preprocessing
4. Metodi “inprocessing”
5. Metodi di postprocessing
6. Valutazione dei modelli (Fairness vs. Accuracy)
7. Adversarial machine learning

Interpretabilità, Spiegabilità e Trasparenza

La terza parte dell’insegnamento mira ad introdurre i principali metodi per migliorare la interpretabilità dei modelli di machine learning e per la spiegabilità delle loro decisioni.

1. Algorithmic transparency nei sistemi decisionali
2. Modelli di machine learning interpretabili
3. Modelli e metodi per la spiegabilità di modelli black-box
4. Tracciabilità nel processamento dei dati

Testi consigliati e bibliografia

Materiale fornito dai docenti.

Bibliografia:

- The Algorithmic Foundations of Differential Privacy: https://www.nowpublishers.com/article/Details/TCS-042
- Fairness and Machine Learning: https://fairmlbook.org/
- A Survey on Bias and Fairness in Machine Learning: https://dl.acm.org/doi/abs/10.1145/3457607
- Technologies for Trustworthy Machine Learning: A Survey in a Socio-Technical Context: https://research.birmingham.ac.uk/en/publications/technologies-for-trustworthy-machine-learning-a-survey-in-a-socio
- A Survey Of Methods For Explaining Black Box Models: https://dl.acm.org/doi/10.1145/3236009

Security Analytics

Numero di CFU - Credits: 6
SSD attività didattica - Scientific Sector of Activity: INF/01 - INFORMATICA TAF - Type of Activity: B - caratterizzante

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Informazioni generali / General Information
Erogazione - Teaching Modality: Tradizionale
Lingua - Language: Italiano
Frequenza - Attendance: Facoltativa ma consigliata
Numero di ore - Number of hours: 48 (32 in aula / 16 in laboratorio)

1. Prerequisiti - Prerequisites

Italiano

- **Competenze attese in ingresso (richieste all'inizio del corso)**
  - Conoscenza di base di reti e sicurezza. Conoscenza di base di Machine Learning e Reti Neurali.
- **Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso")**
  - Sicurezza delle Reti (Magistrale)
  - Apprendimento automatico (Magistrale)
  - Reti Neurali (Magistrale)

2. Obiettivi formativi - Learning objectives

Italiano

Il corso fornisce agli studenti gli strumenti necessari per affrontare le sfide della security analytics. Concetti fondamentali, come SOC, SIEM e SOAR, verranno presentati per aiutare gli studenti a comprendere il loro ruolo. Il corso offre una panoramica dei vari metodi per raccogliere dati sulla sicurezza (da *logs* a *public feeds*). Inoltre, si concentra sullo sviluppo delle competenze necessarie per l'analisi efficace e l'elaborazione di tali dati, compresa la conduzione di *forensics* e *threat detection*.

Le principali metodologie per l'analisi della sicurezza saranno trattate nel corso. Saranno introdotte tecniche di visualizzazione, analisi e previsione di serie temporali, insieme ad algoritmi per la rilevazione di anomalie e l'elaborazione di *security alert.* Inoltre, il corso illustrerà come gli algoritmi di apprendimento automatico, considerati un prerequisito, vengano applicati nel campo della sicurezza informatica. In particolare, le applicazioni della sicurezza informatica introducono la sfida aggiuntiva degli utenti avversari, spesso attaccanti. Il corso dimostrerà come gli approcci di ML possano essere applicati in tali contesti avversari.

English

The course provides students with the necessary tools to tackle typical security analytics challenges. Fundamental concepts, such as **SOC, SIEM, and SOAR**, will be presented to help students understand their roles within the cybersecurity domain. The course offers an overview of various methods for **gathering security data** from sources like logs and

threat intelligence feeds. Additionally, it focuses on developing skills for the effective analysis and processing of such data at scale, including for conducting **forensics** and **threat detection**. The main methodologies essential for security analytics will be covered in the course. It will introduce techniques for **data series visualization, analysis, and forecasting**, along with algorithms for **anomaly detection,** security **alert processing, and correlation**. Moreover, the course will illustrate how machine learning (ML) algorithms, considered a prerequisite, are applied in the field of cybersecurity. Notably, cybersecurity applications introduce the added challenge of adversarial users, often attackers. The course will demonstrate how **ML approaches can be applied in such adversarial settings**.

3. Risultati dell'apprendimento attesi - Learning outcomes

Italiano

Al termine dell'insegnamento lo studente sarà in grado di:

- Raccogliere, analizzare e elaborare dati di sicurezza da diverse fonti (e.g., per la forensic)
- Applicare l'analisi delle serie temporali, compresa la visualizzazione e la previsione, ai dati di sicurezza
- Implementare metodi di rilevamento delle intrusioni
- Applicare *signature-based method*, *supervised classification*, e *unsupervised ML* per l'analisi delle minacce
- Sviluppare soluzioni che siano resistenti agli attacchi avversari, comprese le soluzioni basate sull'apprendimento automatico

English

At the end of the course students will be able to:

- Collect, analyze, and process security data from various sources (e.g., for forensics) ● Apply data series analysis, including visualization and forecasting, to security data ● Implement intrusion detection methods aimed at reducing false positives and improving alert accuracy
- Employ signature-based methods, supervised classification, and unsupervised machine learning for security threat analysis
- Develop solutions resistant to adversarial users, including ML-based solutions

4. Programma - Course Syllabus

Introduction - 2h

- Definitions of SOC, SIEM, SOAR.

Data collection - 10h

- Operating system and server logs
- Packet traces, deep packet inspection and flow records

- Time-series and data series
- Open source threat intelligence feeds
- Scalable tools/platforms for log processing
    - Streaming data processing
    - Time-series databases (e.g.,TSDB)
    - Unstructured logs (e.g., ELK stack)
- Log analysis and forensics

Data series analysis - 12
- Visualization
- Classic time-series models and forecasting
- Multidimensional time-series (correlation and forecasting)
- Generic data series indexing and mining
- Tools for data series analysis

Anomaly detection - 8
- Intrusion Detection and Prevention Systems
- Supervised and unsupervised anomaly detection for security applications
- Alert correlation and false positives

Cyber threat analysis - 8
- Signature-based methods and pattern matching
- Supervised classification of security data (e.g., SPAM and malware classification)
- Drifting
- Unsupervised ML applied to security

Advanced topics on security analytics - 8
- Security of data-driven and AI systems
- Adversarial ML - bypassing ML-based defenses

5. Modalità di verifica dell'apprendimento - Course grade determination

La valutazione dell'esame si comporrà di (i) la consegna delle soluzioni degli esercizi proposti durante il corso; (ii) un esame scritto; (iii) una prova orale di teoria. Tutte le parti dell'esame devono essere superate e contribuiscono a determinare il voto finale secondo una proporzione pre-determinata. La consegna degli esercizi e il superamento della prova scritta sono condizioni necessarie per accedere all'esame orale.

6. Modalità d'insegnamento - Course structure

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 32 ore di lezioni frontali, integrate da esempi e da esercitazioni. Per la parte

di laboratorio sono previste 16 ore di attività in laboratorio in cui si svolgono esercizi. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

7. Attività di supporto - Optional activities

Il materiale didattico di supporto (e.g., lucidi, link, macchine virtuali ecc.) è disponibile presso il supporto on-line ai corsi I-learn. I temi dei progetti e degli esercizi saranno resi disponibili online durante il corso sullo stesso sito.

8. Testi consigliati e bibliografia - Reading materials

- Slides, book chapters, scientific articles and online resources
- (candidate textbooks - partially cover the arguments, must be completed) https://www.amazon.com/Data-Analytics-Cybersecurity-Vandana-Janeja-ebook/dp/B0 9VXQLJYG https://www.amazon.it/Science-Cyber-Security-Security-Technology-English-ebook/d p/B07JPKL8BM

Sicurezza delle Reti / Network Security

Numero di CFU - Credits: 6
SSD attività didattica - Scientific Sector of Activity: INF/01 - INFORMATICA TAF - Type of Activity: B - caratterizzante

Informazioni generali / General Information
Erogazione - Teaching Modality: Tradizionale
Lingua - Language: Italiano
Frequenza - Attendance: Facoltativa ma consigliata
Numero di ore - Number of hours: 48 (32 in aula / 16 in laboratorio)

1. Prerequisiti - Prerequisites

- **Competenze attese in ingresso (richieste all'inizio del corso)**
  - Conoscenza di base di reti, sicurezza e sistemi operativi. Conoscenza di base dei principali linguaggi di script, come il bash, Python ecc.
- **Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso")**
  - Reti di Elaboratori o Reti I (triennale)
  - Sistemi Operativi (triennale)
  - Sicurezza (triennale)

2. Obiettivi formativi - Learning objectives

Il corso copre sia i fondamenti che gli argomenti avanzati sulla sicurezza delle reti. Offre agli studenti gli strumenti per valutare la vulnerabilità dei sistemi connessi, come reagire in caso di incidenti di sicurezza delle reti e come proteggere le reti da comuni tipologie di attacchi. Vengono fornite nozioni sui principali attacchi remoti, tra cui configurazioni errate e rischi derivanti dalla mancanza di meccanismi di protezione di base. Il corso si estende a una vasta gamma di attacchi, dalle reti TCP/IP alle reti mobili. Esempi di attacchi vengono illustrati per offrire una visione d'insieme su come tali minacce possano manifestarsi e come prevenirle. Il corso ha una componente pratica significativa, con lezioni e laboratori che si svolgono in parallelo. Gli studenti avranno l'opportunità di affrontare vari scenari di attacco e parteciperanno allo sviluppo di progetti pratici e a attività hands-on.

3. Risultati dell'apprendimento attesi - Learning outcomes

Alla fine di questo insegnamento, le/gli studentesse/studenti saranno in grado di:

- Comprendere le misure di sicurezza presenti in una vasta gamma di tecnologie e protocolli di rete, dal TCP/IP alle reti mobili.
- Comprendere le vulnerabilità dei sistemi connessi e le loro superfici di attacco.
- Ricercare vulnerabilità nei sistemi connessi utilizzando strumenti per testare la sicurezza.
- Implementare misure di sicurezza per proteggere le reti (come firewall e VPN) e mitigare l'impatto degli attacchi.

4. Programma - Course Syllabus

- Network security basics - 2 h
- Basic MAC layer and TCP/IP attacks - 8 h
    - ARP poisoning
    - Sniffing, spoofing
    - IP fragmentation, ICMP attacks
    - Session Hijacking, Mitnick and TCP RST attacks
    - Flood attacks
    - MiTM attacks
    - Routing and BGP attacks
- UDP and DNS attacks - 8 h
    - DNS basics
    - DDoS and amplification

    - Cache poisoning (local and remote)
    - DoT, DoH and DNSSEC
- NAT, firewall and VPN - 10 h
    - Stateless and stateful firewalls
    - Netfilter
    - NAT and NAT traversal
    - VPNs, tunneling e firewall evasion
    - Intrusion Detection Systems
- Wireless security - 10 h
    - Physical layer, jamming
    - 3G/4G/5G cellular networks and security
    - IEEE 802.11 (wifi) networks, rogue AP, WEP, WPA
    - Bluetooth (pairing, authentication, confidentiality, bluesnarfing, bluejacking ,...)
    - NFC and RFID (security and privacy)
- Network vulnerability assessment - 4 h
    - Scanning for vulnerable services
    - Network penetration testing
- Advanced topics on network security - 6 h
    - Honeypots
    - The darkweb
    - Phishing and squatting

5. Modalità di verifica dell'apprendimento - Course grade determination

La valutazione dell'esame si comporrà di (i) la consegna delle soluzioni degli esercizi proposti durante il corso; (ii) un esame pratico di valutazione della sicurezza di una rete (impostato come una "Capture The Flag"); (iii) una prova orale di teoria. Tutte le parti dell'esame devono essere superate e contribuiscono a determinare il voto finale secondo una proporzione pre-determinata. La consegna degli esercizi e il superamento della prova pratica sono condizioni necessarie per accedere all'esame orale.

6. Modalità d'insegnamento - Course structure

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 32 ore di lezioni frontali, integrate da esempi e da esercitazioni. Per la parte di laboratorio sono previste 16 ore di attività in laboratorio in cui si svolgono esercizi. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

7. Attività di supporto - Optional activities

Il materiale didattico di supporto (e.g., lucidi, link, macchine virtuali ecc.) è disponibile presso il supporto on-line ai corsi I-learn. I temi dei progetti e degli esercizi saranno resi

disponibili online durante il corso sullo stesso sito.

8. Testi consigliati e bibliografia - Reading materials

Du, Wenliang. ***Internet Security: A Hands-on Approach*.** Third edition. 2022.

Sicurezza dei Sistemi / System Security

Numero di CFU - Credits: 6
SSD attività didattica - Scientific Sector of Activity: INF/01 -
INFORMATICA TAF - Type of Activity: B - caratterizzante

Informazioni generali / General Information
Erogazione - Teaching Modality: Tradizionale
Lingua - Language: Italiano
Frequenza - Attendance: Facoltativa ma consigliata
Numero di ore - Number of hours: 48 (32 in aula / 16 in laboratorio)

1. Prerequisiti - Prerequisites

● **Competenze attese in ingresso (richieste all'inizio del corso)** Conoscenza di sicurezza delle reti, sistemi operativi, architettura degli elaboratori.

● **Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso")** ○ Architettura degli Elaboratori II (Magistrale)
- ○ Sicurezza delle Reti (Magistrale)
- ○ Identity and Access Management (Magistrale)

2. Obiettivi formativi - Learning objectives

In questo corso verranno affrontati argomenti avanzati di sicurezza dei sistemi, inclusi i sistemi web, sistemi operativi e sicurezza hardware. Il corso fornirà le competenze per valutare le potenziali vulnerabilità ad attacchi dei sistemi informatici, per sapere come reagire nel caso in cui si verifichi un incidente di sicurezza, e come proteggere sistemi dagli attacchi più comuni.

Il corso fornisce una panoramica delle vulnerabilità più comuni, tra cui errori di programmazione, configurazioni errate o parziali e rischi legati alla mancanza di meccanismi di protezione di base. Verranno inoltre presentati come casi di studio diverse tipologie di attacchi, le opportune contromisure per mitigarne gli effetti e gli strumenti/tool utilizzati nella realizzazione di tali attacchi.

Il corso prevede lo svolgimento di attività di laboratorio in cui gli studenti potranno testare nella pratica i diversi scenari di attacco. Inoltre, le attività in laboratorio e lo sviluppo di un progetto saranno parte integrante della valutazione finale.

3. Risultati dell'apprendimento attesi - Learning outcomes

Al termine dell'insegnamento lo studente sarà in grado di:

- Comprendere le vulnerabilità di un sistema e valutarne le superfici di attacco

- Identificare vulnerabilità in sistemi realistici mediante strumenti/tools atti alla valutazione della sicurezza
- Implementare misure atte ad evitare gli attacchi e/o a mitigarne gli effetti

4. Programma - Course Syllabus

- Main software security problems - 2 h
- Software security - 18 h
    - Intro: x86, ELF, calling conventions, memory management
    - Static vs dynamic analysis, debuggers
    - Reverse engineering, disassembly and decompilation
    - Control hijacking
        - Reverse shell
        - Integer overflows, buffer overflow, use after free
        - ASLR, execution space vs data space
        - Return-to-libc, ROP, JOP, code gadgets
    - Format string vulnerability
    - Shellcode
    - Principles of racing conditions
- Web security - 12 h
    - HTTP security basics, cookies and sessions
    - File disclosure and path traversal
    - Code and commands injection
    - SQL injection
    - Cross-site request forgery
    - Cross-site scripting and CORS
    - Clickjacking, baiting
    - Security of mobile applications
- Hardware and OS security - 12 h
    - Linux security basics

    - Privilege escalation
    - Racing conditions: The dirty cow example
    - Processor security
    - Speculative execution, Spectre e Meltdown and others
- Software testing and fuzzing - 2h
- Information gathering and OSINT - 2h

5. Modalità di verifica dell'apprendimento - Course grade determination

La valutazione dell'esame si comporrà di (i) la consegna delle soluzioni degli esercizi proposti durante il corso; (ii) un esame pratico ("Capture The Flag"); (iii) una prova orale di teoria. Tutte le parti dell'esame devono essere superate e contribuiscono a determinare il voto finale secondo una proporzione pre-determinata. La consegna degli esercizi e il superamento della prova pratica sono condizioni necessarie per accedere all'esame orale.

Esercizi

Durante lo svolgimento del corso verranno assegnati degli esercizi di laboratorio che dovranno essere consegnati entro le scadenze previste. La consegna degli esercizi al di fuori delle scadenze determina una riduzione del punteggio ottenibile.

CTF

I contenuti del corso vengono sintetizzati nella esecuzione di una prova che consiste nell'analisi di un sistema (applicazione/servizio/ecc.) volta all'individuazione ed all'exploitation delle vulnerabilità e degli errori di configurazione presenti utilizzando le tecniche apprese durante il corso.

6. Modalità d'insegnamento - Course structure

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 32 ore di lezioni frontali, integrate da esempi e da esercitazioni. Per la parte di laboratorio sono previste 16 ore di attività in laboratorio in cui si svolgono esercizi. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

7. Attività di supporto - Optional activities

Il materiale didattico di supporto (e.g., lucidi, link, macchine virtuali ecc.) è disponibile

presso il supporto on-line ai corsi I-learn. I temi dei progetti e degli esercizi saranno resi disponibili online durante il corso sullo stesso sito.

8. Testi consigliati e bibliografia - Reading materials

Du, Wenliang. ***Computer Security: A Hands-on Approach***. Third Edition. 2022.

TLN-1: Trattamento del Linguaggio Naturale (1)

6 CFU (48 ore)
Prerequisiti
Comprensione dei concetti di base di programmazione e algoritmi.
Insegnamenti propedeutici (forniscono le competenze attese in ingresso)
Dalla laurea triennale: Sistemi intelligenti, Basi di dati, Algoritmi. Dalla magistrale:

Apprendimento Automatico o Reti Neurali e Deep Learning.

Sommario insegnamento
- Obiettivi formativi
- Risultati dell'apprendimento attesi
- Modalità di insegnamento
- Modalità di verifica dell'apprendimento
- Programma
- Testi consigliati e bibliografia
- Strumenti didattici

Obiettivi formativi
Questo corso mira a fornire una comprensione approfondita e competenze avanzate nel

trattamento del linguaggio naturale. Gli studenti acquisiranno conoscenze teoriche e

pratiche necessarie per affrontare sfide complesse nella comprensione, generazione e

analisi automatica del testo.

Risultati dell'apprendimento attesi
CONOSCENZA E CAPACITÀ DI COMPRENSIONE
- Comprendere a fondo i livelli linguistici e le loro interazioni.
- Conoscere le tecniche avanzate di PoS (Part-of-Speech) e NER (Named Entity Recognition) tagging.
- Comprendere i principi della sintassi computazionale.
- Avere una conoscenza avanzata delle rappresentazioni vettoriali per il TLN. ● Conoscere approcci avanzati per il topic modeling.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE
- Applicare tecniche avanzate di PoS e NER tagging su testi reali.

- Utilizzare analisi sintattiche avanzate per l'analisi del testo.
- Utilizzare risorse lessico-semantiche per task quali la Word Sense Disambiguation e la Semantic Similarity
- Scegliere e applicare modelli di rappresentazione vettoriale
- Sviluppare applicazioni di TLN avanzate.

AUTONOMIA DI GIUDIZIO

- Valutare in modo critico l'efficacia delle tecniche di elaborazione del linguaggio naturale. ● Scegliere le metodologie appropriate per specifici compiti di TLN.

Modalità di insegnamento

Il corso prevede 48 ore di lezioni frontali, esercitazioni pratiche e discussioni interattive. Gli studenti saranno coinvolti in progetti pratici legati al trattamento del linguaggio naturale e parteciperanno a seminari tenuti da esperti del settore.

Modalità di verifica dell'apprendimento

Gli studenti sono tenuti a partecipare attivamente alle esercitazioni pratiche e ai progetti assegnati durante il corso. La valutazione avverrà attraverso esami scritti e/o orali in cui gli studenti dimostreranno la loro comprensione delle materie trattate e la loro capacità di applicare le conoscenze acquisite.

Programma

- Introduzione ai livelli linguistici.
- PoS/NER tagging avanzato.
- Analisi sintattica avanzata.
- Semantica computazionale e rappresentazioni vettoriali.
- Natural Language Generation.
- Sistemi di dialogo.
- Lexical Semantics e risorse lessicali.
- Topic modeling.
- Applicazioni avanzate nel trattamento del linguaggio naturale.

Testi consigliati e bibliografia

- Materiale fornito dai docenti.
- Jurafsky, D., & Martin, J. H. (2020). "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition."

Strumenti didattici

L'ambiente di e-learning Moodle sarà utilizzato per la distribuzione di materiale didattico, discussioni online e consegna di progetti. Gli studenti avranno accesso a risorse e dataset per l'elaborazione del linguaggio naturale avanzata.

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

TLN-2: Trattamento Avanzato del Linguaggio Naturale con Deep Learning e Large Language Models

6 CFU (48 ore)

Prerequisiti

Conoscenza avanzata di trattamento del linguaggio naturale (TLN) e deep learning, comprensione dei principi fondamentali delle reti neurali artificiali.

Insegnamenti propedeutici (forniscono le competenze attese in ingresso)

Dalla laurea triennale: Sistemi intelligenti, Basi di dati, Algoritmi. Dalla magistrale: Apprendimento Automatico, Reti Neurali e Deep Learning, Trattamento del Linguaggio Naturale 1 (TLN-1)

Sommario insegnamento

- Obiettivi formativi
- Risultati dell'apprendimento attesi
- Modalità di insegnamento
- Modalità di verifica dell'apprendimento
- Programma
- Testi consigliati e bibliografia
- Strumenti didattici

Obiettivi formativi

Questo corso mira a fornire una conoscenza avanzata e competenze specialistiche nel trattamento del linguaggio naturale utilizzando tecniche di deep learning e large language models (LLM). Gli studenti acquisiranno conoscenze teoriche e pratiche necessarie per affrontare sfide avanzate nell'elaborazione del linguaggio naturale, con un focus particolare sui LLM.

Risultati dell'apprendimento attesi
CONOSCENZA E CAPACITÀ DI COMPRENSIONE

- Comprendere approfonditamente il meccanismo dell'attenzione nei modelli di LLM.
- Conoscere i concetti di self-supervision e la loro applicazione nel TLN. ●

Avere una conoscenza avanzata delle architetture transformer-based. ● Comprendere il funzionamento delle AI generative nel contesto del TLN multimodale.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE

- Sviluppare sistemi self-supervised per il TLN.
- Utilizzare architetture transformer-based per compiti specifici del TLN. ● Creare applicazioni multimodali avanzate che coinvolgono testo, audio e immagini.

AUTONOMIA DI GIUDIZIO

- Valutare criticamente l'uso dei LLM in vari contesti e comprenderne i limiti e le potenzialità.
- Scegliere e adattare LLM per compiti specifici di TLN.

Modalità di insegnamento

Il corso prevede 48 ore di lezioni avanzate, laboratori pratici e discussione interattiva. Gli studenti avranno l'opportunità di lavorare su progetti pratici avanzati che coinvolgono large language models.

Modalità di verifica dell'apprendimento

Gli studenti saranno valutati attraverso esami scritti e/o orali, nonché tramite progetti pratici avanzati. Gli esami metteranno alla prova la comprensione teorica e la capacità di applicare le conoscenze acquisite nell'ambito del TLN avanzato. Sono previsti seminari tenuti da esperti sugli argomenti trattati.

Programma

- Deep Learning e NLP: Meccanismo dell'attenzione e self-supervision.
- Architetture transformer-based per il TLN.
- AI generativa e Multimodal NLP: Trasformazione del testo in altre modalità (audio, immagini) e viceversa.
- Large Language Models: raw language models, pre-training, fine-tuning, instruction-tuned models, prompting, limiti e sicurezza.
- Explainability nel contesto dei LLM.

Testi consigliati e bibliografia

- Materiale fornito dai docenti.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention Is All You Need."
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). "Improving Language Understanding by Generative Pre-training."
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). "Language Models are Few-Shot Learners."

Strumenti didattici

L'ambiente di e-learning Moodle verrà utilizzato per la distribuzione di materiale

didattico, discussioni online, consegna di progetti e laboratori pratici. Gli studenti avranno accesso a risorse e dataset avanzati per l'elaborazione del linguaggio naturale e per l'applicazione dei large language models.

********** L31**********

Problem Solving e Programmazione Competitiva (syllabus sostituito con nuova proposta inserita nell'Allegato al verbale della riunione del 18 ottobre 2023)

Docenti di riferimento: Giorgio Audrito , Elvio Amparore
Corso di studio: laurea triennale
Anno: 2º anno (?)
Periodo: preferibilmente IIº semestre per sinergia con le date in cui va comunicata la squadra del dipartimento alle SWERC-ICPC (ottobre-novembre)
Tipologia: Non caratterizzante
Crediti/Valenza: crediti liberi, 3 CFU (ore aula: 24) [eventualmente aumentabili a 6 CFU (48 ore) con opportuna estensione della proposta]
SSD attività didattica: INF/01 - informatica
Erogazione: Tradizionale
Lingua: Italiano
Frequenza: richiesta almeno al 50%
Tipologia esame: Scritto

Obiettivi formativi

In riferimento agli obiettivi formativi specifici del CdS in Informatica (LM18), il corso ha i seguenti obiettivi formativi:

1. Fornire gli strumenti e le metodologie per il problem solving in forma competitiva, sotto limiti di tempo. La forma competitiva è tipicamente utilizzata in gare di programmazione o durante le fasi di recruiting in aziende ICT di alto profilo.
2. Acquisire la padronanza di linguaggi di programmazione e pattern di coding adatti allo sviluppo di codice ad alte performance in poco tempo.
3. Introdurre tecniche algoritmiche avanzate, non trattate negli insegnamenti di *Algoritmi e Strutture Dati* (MFN0597) e *Algoritmi e Complessità* (INF0097).
4. Esercitare tecniche e metodi per il lavoro di squadra, nell'ambito della programmazione ad alte performance e competitiva.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITÀ DI COMPRENSIONE
Conoscere algoritmi e strutture dati utili per la programmazione ad alte performance e competitiva. Saper comprendere e interpretare un problema algoritmico per avviare la progettazione della soluzione.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE
Capacità di utilizzare in maniera adeguata gli algoritmi e le strutture dati necessarie per risolvere problemi di programmazione non banali e con alte performance. Saper applicare tecniche avanzate di programmazione per ridurre la complessità in tempo e memoria, e realizzare soluzioni efficienti.
AUTONOMIA DI GIUDIZIO
Capacità di autovalutazione e stima della complessità algoritmica di una soluzione, sfruttando anche il supporto di strumenti come piattaforme online di valutazione su use-case difficili e/o su grandi moli di dati,

ABILITÀ COMUNICATIVE
Capacità di comunicare idee e soluzioni per problemi algoritmici, applicate in particolare al lavoro di squadra, utilizzando un linguaggio scientifico adeguato ed efficace.

CAPACITÀ DI APPRENDIMENTO
Acquisizione di capacità autonome di apprendimento, tramite identificazione di risorse e materiali rilevanti, incluse piattaforme di valutazione del codice, e definizione di obiettivi chiari nella progettazione e analisi di algoritmi avanzati.

Programma

1. Introduzione alla programmazione competitiva e ad alte performance: come affrontare problemi algoritmici con limiti di tempo e di memoria.
2. Problemi che si possono risolvere tramite libreria standard di Java o C++. 3. Problemi con soluzione greedy, ottimizzazione e finestre scorrevoli. Capire quando una soluzione greedy è sufficiente per risolvere il problema. 4. Ricorsione esaustiva, backtracking e strategie per ridurre lo spazio da esplorare. 5. Catalogo di strategie risolutive e programmazione dinamica.
6. Algoritmi su grafi, visite e cammini, componenti fortemente connesse.
7. Alberi minimi ricoprenti e struttura dati Union-Find.
8. Ordinamento topologico e algoritmi su DAG.
9. Strutture dati per operazioni su intervalli: somme prefisse, range tree, treap.
10. Algoritmi di flusso su grafi e matching perfetto.
11. Tecniche di decomposizione: square-root, binary lifting, centroid e heavy-light. 12. Geometria computazionale e discrete Fourier transform.

Modalità di insegnamento

Il corso si articola in 12 moduli da 2 ore ciascuno. Ogni modulo affronta un argomento alternando parti teoriche e di esercitazione.
Le parti teoriche sono effettuate come lezioni frontali con il supporto di lucidi e altro materiale di approfondimento. Le nozioni teoriche vengono declinate in particolare riferimento a dei problemi presentati, e a come possono essere applicate ad essi.
Le esercitazioni tematiche si svolgono esponendo un insieme di problemi che gli studenti devono analizzare, trovando soluzioni efficienti e determinandone la complessità e potenziali problematiche, ed infine sviluppando codice efficiente.

Modalità di verifica dell'apprendimento

L'esame si terrà sotto forma di prova di programmazione al computer. La prova prevederà la risoluzione di 3 problemi algoritmici in un tempo massimo di 3 ore. Ogni problema prevede un punteggio parziale a seconda della complessità della soluzione prodotta. I problemi saranno proposti in lingua inglese, come nelle competizioni internazionali esistenti.
Nei mesi da settembre a marzo saranno proposti 5 appelli di esame o esonero in occasione delle competizioni nazionali delle olimpiadi di informatica individuali e a squadre, basati ciascuno su una selezione dei problemi proposti a quelle gare. Il primo di questi appelli sarà anche valevole come selezione della squadra che rappresenti l'Università di Torino alla competizione internazionale SWERC dell'International Collegiate Programming Contest.

Testi consigliati e bibliografia

Tutto il materiale di studio necessario verrà caricato nella pagina moodle del corso durante le lezioni, con riferimenti anche ad alcune attività accessibili online. Tutto il materiale sarà fornito in digitale e sarà possibile consultarlo online e, se si desidera, scaricarlo e stamparlo.

Un testo consigliato (non obbligatorio) sull'argomento è:

Competitive Programming 4: The Lower Bound of Programming Contest in the 2020s. Steven Halim, Felix Halim, Suhendry Effendy
Libri 1 e 2. Lulu Press, Incorporated, 2018 e 2020.
ISBN: 1716745527 e 1716745519.

.....
.....

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

mercoledì 18 ottobre 2023 ore 16.30
Collegamento alla riunione:
Giunta di CCL-LM

https://unito.webex.com/unito/j.php?MTID=mdcb1367ad2569e47df491747df69bc10

per trattare il seguente Ordine del Giorno:

1. Aggiornamento analisi delle proposte di apertura di nuovi insegnamenti L31 e LM18
2. Approvazione verbale seduta precedente aggiornato alla data odierna

La Presidente della Giunta di CCL-LM
(prof.ssa Liliana Ardissono)

ELENCO DEI COMPONENTI della Giunta di CCL-LM in Informatica:
Ardissono Liliana, Cardone Felice, Esposito Roberto, Gaeta Rossano, Petrone Giovanna, Pozzato Gian Luca, Sapino Maria Luisa, Sirovich Roberta, Sproston Jeremy James

IN CONGEDO: Pensa Ruggero Gaetano

SONO PRESENTI: Ardissono Liliana, Cardone Felice, Esposito Roberto, Pozzato Gian Luca

ASSENTI GIUSTIFICATI: Gaeta Rossano, Petrone Giovanna, Sapino Maria Luisa, Sproston Jeremy James

OSPITI:
Matteo Baldoni, Paola Gatti

La seduta ha inizio alle ore 16:30.

1. *Aggiornamento analisi delle proposte di apertura di nuovi insegnamenti L31 e LM18*

La seduta riprende dopo la riunione del giorno 11 ottobre 2023.

Si analizzano le risposte dei docenti che hanno proposto gli insegnamenti.
Laddove le modifiche alle proposte sono circoscritte, si riportano i dati aggiornati, o le spiegazioni, localmente alla parte di verbale dell'11 ottobre: per evidenziare le modifiche apportate, queste sono **in giallo**. Le altre modifiche vengono riportate nel seguito. Le bozze di syllabus riviste rispetto alla versione dell'11 ottobre 2023 si trovano nell'**Allegato al Verbale della Giunta di CCL-LM del 18 ottobre 2023.**

Dopo ampia e approfondita discussione, la Giunta esprime le seguenti proposte:

- I seguenti insegnamenti potrebbero essere istituiti nella nuova offerta didattica, se la Commissione Didattica di Dipartimento riterrà che ci siano sufficienti risorse per la loro copertura:
  - LM18:
    - Sdoppiamento dell'insegnamento Sicurezza delle reti e dei sistemi (6 CFU) in (i) Sicurezza delle reti (6 CFU) e (ii) Sicurezza dei sistemi (6 CFU)
    - Algoritmi per ottimizzazione non lineare (6 CFU)
    - Computazione quantistica (6 CFU)
  - L31:
    - Advanced Problem Solving (3 CFU). La Giunta segnala che l'insegnamento dovrebbe essere previsto per il terzo anno di corso in quanto a scelta.
    - Blockchain, sistemi distribuiti e decentralizzati (6 CFU)

- I restanti insegnamenti meritano ulteriori approfondimenti e si suggerisce pertanto di discuterli nell'ambito della prevista ristrutturazione della LM18.

La Giunta ricorda ai proponenti che i syllabus riportati in questo verbale sono bozze da migliorare.

2. ***Approvazione verbale seduta precedente aggiornato alla data odierna***

Salvo il punto 4.1 del verbale del giorno 11 ottobre 2023, il presente verbale completo di seduta aggiornata alla data odierna, viene **approvato seduta stante all'unanimità**.

La seduta è tolta alle ore 18:20.

La Presidente
Prof.ssa Liliana Ardissono

Il Segretario verbalizzante
Prof. Roberto Esposito

Allegato al Verbale della Giunta di CCL-LM del 18 ottobre 2023

Etica e l'impatto sociale dell'IA / *Ethics and the social impact of AI*

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Corso di studio

[008515] Laurea magistrale in informatica

Anno

1° anno, 2° anno

Tipologia

Caratterizzante

Mettere il corso come a scelta libera impedirebbe di laureare persone consapevoli degli effetti del lavoro che andranno a fare nello sviluppo di sistemi AI, e cittadini consapevoli degli impatti sulla società delle tecnologie che useranno. Non si tratta di un corso di deontologia professionale che spiega regole da seguire ma di un corso che permette una consapevolezza più generale dei problemi legati all'AI partendo dagli aspetti tecnici.
La differenza rispetto al corso di Apprendimento Automatico Responsabile e Affidabile sta nel fatto che Etica e impatto sociale dell'IA è focalizzato sul come e perché le tecnologie AI impattano sulla società, sul perché sia necessario un approccio etico e sui rischi inerenti all'AI che non possono essere affrontati solamente con le metodologie proposte nel primo corso ma richiedono un'azione non solo di tipo tecnologico.

Crediti/Valenza

6 CFU - Numero di ore - Number of hours: 48 (in aula)

Crediti percorso 24 CFU

2

SSD attività didattica

INF/01 – informatica:

Non si tratta di un corso di carattere filosofico ma interdisciplinare, che richiede sia per lo studente che per il docente una conoscenza approfondita di come funzionino i sistemi autonomi e il machine learning e delle sue applicazioni quali NLP o Computer Vision.
Non si tratta di un corso di filosofia morale ma un corso che parte dall'analisi dell'impatto dell'AI nella società e sui valori etici dal punto di vista tecnico e mette in luce in particolare dove la tecnologia non può arrivare a risolvere i problemi che crea ma siano inerenti alla tecnologia stessa.
E' focalizzato a spiegare come la tecnologia sviluppata è sempre parte di un sistema economico e di potere e sociale.
Non è un corso che può essere insegnato da un umanista né è diretto a umanisti: il Direttore del Dipartimento di Filosofia e Scienze dell'Educazione, Prof. Graziano Lingua, che è pure ordinario di Filosofia Morale e quindi la persona di riferimento per questi temi, conosce il corso e la proposta di estensione e condivide questa posizione.
Esempi di corsi simili sono quello di Responsible AI a Polito:
https://didattica.polito.it/pls/portal30/gap.pkg_guide.viewGap?p_cod_ins=01DTEOV&p_a_acc=2023&p_header=S&p_lang=IT&multi=N
Settore ING-INF/05 che però ha un focus primario sulla deontologia professionale e un carattere meno interdisciplinare (60 ore).
Mentre a UniBo il corso è obbligatorio ma ha una connotazione più giuridica perché tenuto dal Prof. Giovanni Sartor, giurista esperto e riferimento della materia in quella università:
https://www.unibo.it/it/didattica/insegnamenti/insegnamento/2023/446601 (6 CFU)

Anche nella triennale di Pavia su AI il corso ha più carattere giuridico: http://www-4.unipv.it/offertaformativa/portale/corso.php?lingua=2&idAttivitaFormativa=348007&modulo=0&anno=2021/2022 (12 crediti)
Cosi come a IULM ma in una laurea AI di comunicazione: https://www.iulm.it/it/offerta-formativa/corsi-di-lauree-magistrali/intelligenza-artificiale-impresa-societa/piano-studi
Anche la laurea online dell'eCampus include temi etici e giuridici: https://www2.uniecampus.it/facolta/facolta-ingegneria/corso-laurea-magistrale-ingegneria-informatica-automazione-artificial-intelligence.asp#form
Laurea Triennale in Inglese di Bocconi, però con orientamento matematico economico ha Seminars in Digital Ethics and in Behavioural Skills di stampo giuridico
https://www.unibocconi.eu/wps/wcm/connect/bocconi/sitopubblico_en/navigation+tree/home/programs/bachelor+of+science/mathematical+and+computing+sciences+for+artificial+intelligence/program+structure/
L'unico corso in settore filosofico è quello della Statale di Milano che però è in una laurea AI per umanisti e non in informatica: https://www.unimi.it/en/education/degree-programme-courses/2024/ai-ethics-and-law

Erogazione
Tradizionale
Lingua
Italiano
Frequenza
Facoltativa
Tipologia esame
Orale
Tipologia unità didattica
corso
Prerequisiti
Conoscenze elementari di probabilità e statistica, algoritmi, basi di dati, apprendimento automatico e/o reti neurali.
Conoscenza di sistemi AI autonomi.
Conoscenza dei sistemi AI basati su reti neurali e su Deep Learning.
Conoscenza dei sistemi di AI generativa come i LLM.

Insegnamenti propedeutici (forniscono le competenze attese in ingresso):

Dalla laurea triennale: Sistemi intelligenti, Basi di dati, Algoritmi.

Dalla magistrale:

Intelligenza Artificiale e Laboratorio
Istituzioni di Sistemi Intelligenti
Apprendimento Automatico
Reti Neurali e Deep Learning
Tecnologie del Linguaggio Naturale

Obiettivi formativi

L'insegnamento concorre al raggiungimento degli obiettivi formativi specifici del Corso di Laurea Magistrale in Informatica (LM18) creando competenze tecniche che permettono ai laureati di comprendere se e come si possono risolvere i problemi relativi all'impatto dell'AI sulla società e come creare sistemi AI allineati ai principi etici condivisi nella società contemporanea in particolare alla luce dei recenti sviluppi dell'AI generativa. Si propone perciò di fornire le competenze informatiche e interdisciplinari necessarie a comprendere come gli algoritmi AI e il loro uso nei diversi contesti abbiano un impatto sulla società, sugli individui, sulle categorie più vulnerabili (donne, giovani, poveri, migranti, ecc.). Se da un lato promuovono sistemi AI che rispettino i valori etici fin dalle fasi di ideazione e progettazione, dall'altro le tecnologie non sempre possono evitare tramite sviluppi tecnologici l'impatto negativo sulla società o sui valori etici condivisi. Il corso permette di avere laureati che siano consapevoli del loro ruolo nel mondo lavoro che faranno e come cittadini e in grado di evitare ove possibile il disallineamento etico e di difendere la loro posizione nei confronti di datori di lavoro, clienti ed altri stakeholders.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITÀ DI COMPRENSIONE

Al termine dell'insegnamento si dovranno conoscere:

- La necessità di un approccio etico all'AI e di minimizzare l'impatto negativo sulla società.
- Valutazione dell'impatto delle diverse scelte di design di sistemi AI e capacità di scegliere quelle che minimizzano l'impatto.
- Cosa sia l'etica applicata alla tecnologia.
- Capacità di comprensione degli eventi più recenti legati alle novità tecnologiche nell'ambito dell'AI e oltre.
- I rischi sollevati dalle tecnologie dell'informazione nelle loro diverse dimensioni: economici, politici, tecnologici, sociali, ecc.

- l'impatto delle tecnologie dell'informazione sulla società contemporanea.
- I limiti delle tecnologie AI nel ridurre i loro impatti.
- I fattori economici, sociali, geopolitici che influenzano lo sviluppo dell'AI.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE

Al termine dell'insegnamento si dovrà essere in grado di:

- prevedere gli impatti dei sistemi AI su cui lavora al di là delle prospettive tecnosoluzioniste.
- adottare un approccio etico all'AI andando oltre le puramente soluzioni deontologiche tipiche degli approcci "by design".
- valutare l'impatto sulla società di una tecnologia informatica che stanno sviluppando.

AUTONOMIA DI GIUDIZIO
Alla fine di questo insegnamento si saprà:

- ideare, progettare e programmare sistemi AI evitando ricadute negative sulla società e di violare principi etici.
- Decidere quando è necessario fare notare a datori di lavoro e committenti di sistemi AI gli effetti negativi che possono avere sulla società.
- Consapevolezza delle motivazioni per cui i sistemi AI che stanno sviluppando hanno un impatto negativo sulla società o violano principi etici condivisi.

ABILITÀ COMUNICATIVE
Al termine dell'insegnamento si sarà in grado di:

- Argomentare, in forma scritta e/o orale, le decisioni prese durante lo sviluppo di sistemi AI per spiegare come questi siano allineati a valori etici condivisi e minimizzino l'impatto negativo sulla società.
- Spiegare al datore di lavoro quando i sistemi AI che gli viene chiesto di sviluppare hanno un impatto negativo sulla società o violano principi etici condivisi.
- Spiegare ai clienti committenti dei sistemi AI sviluppati quali sono i limiti da rispettare per un uso etico degli stessi.
- Interpretare correttamente le informazioni sui media relative alle novità tecnologiche epurando le finalità di marketing che le avvolgono e le forme di retorica e propaganda.

CAPACITÀ DI APPRENDIMENTO
Al termine dell'insegnamento, si saranno acquisite capacità autonome di apprendimento e di autovalutazione della propria preparazione, atte a integrarsi nel mondo del lavoro con un alto grado di coscienza critica e autonomia di giudizio.

Programma

Etica e impatto sociale dell'IA

L'insegnamento mira a presentare e discutere i problemi relativi all' impatto e all'etica sollevati dai sistemi AI nella società contemporanea, dal possibile abuso dei social network in politica ai rischi nell'uso dei "big data" per la personalizzazine, dall'impatto dell'informatica sulle libertà fondamentali alla diffusione dell'AI generativa.

Si adotterà una prospettiva non puramente deontologica e non tecnosoluzionista: alcuni problemi non sono risolvibili dall'evoluzione tecnologia.

Si andrà oltre i requisiti di trustworhiness, accountability, fairness e transparency per comprendere come l'AI sia parte integrante di un sistema politico ed economico globale. Si considereranno anche le relazioni tra l'AI e le leggi e i regolamenti.

- AI generativa, manipolazione e disinformazione

1. Le ragioni tecniche dei problemi di ChaGPT legate al design dei Trasnformers: le allucinazioni.
2. Manipolazione cognitiva, synthetic relationships.
3. Problematiche relative alla gestione di training set per i LLM
4. Il sovvertimento delle regole del copyright e l'espropriazione della conoscenza per l'addestramento degli LLM.
5. L'hacking del sistema operativo della società.
6. Impatto della generazione di immagini sul discorso pubblico.
7. Rischi di discriminazione di genere causati da AI generativa: deepfake e stereotipi.

- Irrisolvibilità tecnologica dei problem di bias e categorizzazione
  1. Limit dell'uso di ImageNet e Wordnet nell'addestramento di sistemi di visione artificiale.
  2. Teorie della categorizzazione e limiti della categorizzazione come semplificazione del mondo e atto politico.
  3. Inevitabilità dei bias nel Machine Learning.
  4. Bias come problema politico e non tecnologico.
  5. Contraddittorietà dei principi trustworhiness, accountability, fairness e transparency
- Deep Learning e l'impossibilità dell'explainability
  1. Cosa è una spiegazione?
  2. Deep Learning, conoscenza tacita e teoria del caos
- Cosa è l'Etica e come è applicata al mondo dell'AI
  1. AI in medicina
  2. AI nel mondo sociale
  3. AI e guerra
  4. AI e altri domini applicativi
  5. I limiti delle ethical guidelines e degli approcci deontologici
- Problemi etici dell'emotion recognition
  1. Teorie usate nell'emotion recognitio e epidermizzazione delle emozioni
  2. Rischi per la società: manipolazione e controllo
- Il capitalismo della sorveglianza
  1. L'algoritmo di Google e la nascita della pubblicità personalizzata
  2. Il modello di business basato sulla espropriazione dei dati.
  3. Algoritmi per la raccolta di dati personali in nuovi domini
  4. Impatto sulla privacy dei cittadini
- L'economia dei mercanti dell'attenzione
  1. Evoluzione del mercato dell'advertisement digitale
  2. Modelli di business alternativi allo sfruttamento dei dati

- I social media e connessioni con la politica
  1. Dai social network ai social media: l'algoritmo di Tik Tok
  2. Gli algoritmi di personalizzazione e come vengono sfruttati nelle campagne politiche
  3. Impatto degli algoritmi di personalizzazione sulle elezioni politiche
  4. Lo scandalo Cambridge Analitica e Edward Snowden
  5. L'intreccio Silicon Valley – Washington

- AI Act della EU: potenzialità e limiti dell'approccio risk based
- Splinternet:
    1. La separazione di internet a livello di stati nazionali
    2. Implicazioni di una AI nazionalista
- Economia dell'AI, competizione e lavoro
    1. Le tendenze monopolistiche delle imprese dell'ICT
    2. Il futuro del lavoro
- Le ideologie di Silicon Valley e la retorica dietro il marketing di nuove tecnologie AI

Modalità di insegnamento

L'insegnamento è strutturato in 48 ore di didattica frontale che prevedono una forte componente interattiva tra docenti e studentesse/studenti. Le ore sono suddivise in lezioni da 2 ore in base al calendario accademico. Sono previsti seminari tenuti da esperti sugli argomenti trattati. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non. Le lezioni verranno registrate e messe a disposizione su Moodle

Modalità di verifica dell'apprendimento

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove si potranno trovare forum di discussione, materiale didattico, e approfondimenti sugli argomenti delle lezioni preparati dai docenti. L'esame consiste in un colloquio atto a valutare la comprensione generale del corso e di argomenti specifici concordati con i docenti. Durante il colloquio la candidata o il candidato presenteranno un loro approfondimento fatto su temi di attualità legati all'etica dell'AI e all'impatto dell'AI. La valutazione è in trentesimi.

Testi consigliati e bibliografia

I principali testi usati durante il corso sono i seguenti, ma il programma di esame prevede una selezione di due libri concordata con il docente:

- The Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence. Kate Crawford 2021
- The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power Shoshana Zuboff 2018
- Weapons of Math Destruction. Cathy O'Neil 2016
- Everyday Chaos Technology, Complexity, and How We're Thriving in a New World of Possibility. David Weinberger 2019
- The Master Switch: The Rise and Fall of Information Empires. Tim Wu 2010
- The Attention Merchants: How Our Time and Attention Are Gathered and Sold. Tim Wu 2017
- The Curse of Bigness - Antitrust in the new Gilded Age. Tim Wu 2018

Università degli Studi di Torino
Dipartimento di Informatica
Corso di Laurea e Laurea Magistrale in Informatica

Problem Solving Avanzato
(Advanced Problem Solving)

Docenti di riferimento: Giorgio Audrito, Elvio Amparore
Corso di studio: L31
Anno: 2º anno (?)
Periodo: preferibilmente IIº semestre per sinergia con le date in cui va comunicata la squadra del dipartimento alle SWERC-ICPC (ottobre-novembre)
Tipologia: Non caratterizzante
Crediti/Valenza: crediti liberi, 3 CFU (ore aula: 24) [eventualmente aumentabili a 6 CFU (48 ore) con opportuna estensione della proposta]
SSD attività didattica: INF/01 - informatica
Erogazione: Tradizionale
Lingua: Italiano
Frequenza: richiesta almeno al 50%
Tipologia esame: Scritto

Obiettivi formativi

In riferimento agli obiettivi formativi specifici del CdS in Informatica (LM18), il corso ha i seguenti obiettivi formativi:

1. Acquisire la padronanza di linguaggi di programmazione e pattern di coding adatti al rapido sviluppo di algoritmi efficienti, nonché nella capacità di selezionare velocemente l'algoritmo giusto per una data situazione.
2. Introdurre tecniche algoritmiche avanzate, non trattate negli insegnamenti di *Algoritmi e Strutture Dati* (MFN0597) e *Algoritmi e Complessità* (INF0097).
3. Sviluppare la capacità di lavorare in squadra per risolvere problemi algoritmici, e a comunicare in modo efficace. Il corso segue un format tipicamente utilizzato in gare di programmazione o durante le fasi di recruiting in aziende ICT di alto profilo.

Risultati dell'apprendimento attesi

CONOSCENZA E CAPACITÀ DI COMPRENSIONE

Conoscere algoritmi e strutture dati utili per la programmazione ad alte performance. Saper comprendere e interpretare un problema algoritmico per avviare una rapida progettazione e realizzazione della soluzione.

CAPACITÀ DI APPLICARE CONOSCENZA E COMPRENSIONE

Capacità di utilizzare in maniera adeguata gli algoritmi e le strutture dati necessarie per risolvere problemi di programmazione non banali e con alte performance. Saper applicare tecniche avanzate di programmazione per ridurre la complessità in tempo e memoria, e realizzare soluzioni efficienti.

AUTONOMIA DI GIUDIZIO

Capacità di autovalutazione e stima della complessità algoritmica di una soluzione, sfruttando anche il supporto di strumenti come piattaforme online di valutazione su use-case difficili e/o su grandi moli di dati.

ABILITÀ COMUNICATIVE

Capacità di comunicare idee e soluzioni per problemi algoritmici, applicate in particolare al lavoro di squadra, utilizzando un linguaggio scientifico adeguato ed efficace.

CAPACITÀ DI APPRENDIMENTO

Acquisizione di capacità autonome di apprendimento, tramite identificazione di risorse e materiali rilevanti, incluse piattaforme di valutazione del codice, e definizione di obiettivi chiari nella progettazione e analisi di algoritmi avanzati.

Programma

1. Introduzione alla programmazione ad alte performance e competitiva: come affrontare problemi algoritmici con limiti di tempo e di memoria.
2. Problemi che si possono risolvere tramite libreria standard di Java o C++.
3. Problemi con soluzione greedy, ottimizzazione e finestre scorrevoli. Capire quando una soluzione greedy è sufficiente per risolvere il problema.
4. Ricorsione esaustiva, backtracking e strategie per ridurre lo spazio da esplorare.
5. Catalogo di strategie risolutive e programmazione dinamica.
6. Algoritmi su grafi, visite e cammini, componenti fortemente connesse.
7. Alberi minimi ricoprenti e struttura dati Union-Find.
8. Ordinamento topologico e algoritmi su DAG.
9. Strutture dati per operazioni su intervalli: somme prefisse, range tree, treap.
10. Algoritmi di flusso su grafi e matching perfetto.
11. Tecniche di decomposizione: square-root, binary lifting, centroid e heavy-light.
12. Geometria computazionale e discrete Fourier transform.

Modalità di insegnamento

Il corso si articola in 12 moduli da 2 ore ciascuno. Ogni modulo affronta un argomento alternando parti teoriche e di esercitazione.
Le parti teoriche sono effettuate come lezioni frontali con il supporto di lucidi e altro materiale di approfondimento. Le nozioni teoriche vengono declinate in particolare riferimento a dei problemi presentati, e a come possono essere applicate ad essi.
Le esercitazioni tematiche si svolgono esponendo un insieme di problemi che gli studenti devono analizzare, trovando soluzioni efficienti e determinandone la complessità e potenziali problematiche, ed infine sviluppando codice efficiente.

Modalità di verifica dell'apprendimento

L'esame si terrà sotto forma di prova di programmazione al computer. La prova prevederà la risoluzione di 3 problemi algoritmici in un tempo massimo di 3 ore. Ogni problema prevede un punteggio parziale a seconda della complessità della soluzione prodotta. I problemi saranno proposti in lingua inglese, come nelle competizioni internazionali esistenti.

Nei mesi da settembre a marzo saranno proposti 5 appelli di esame o esonero in occasione delle competizioni nazionali delle olimpiadi di informatica individuali e a squadre, basati ciascuno su una selezione dei problemi proposti a quelle gare. Il primo di questi appelli sarà anche valevole come selezione della squadra che rappresenti l'Università di Torino alla competizione internazionale SWERC dell'International Collegiate Programming Contest.

Testi consigliati e bibliografia

Tutto il materiale di studio necessario verrà caricato nella pagina moodle del corso durante le lezioni, con riferimenti anche ad alcune attività accessibili online. Tutto il materiale sarà fornito in digitale e sarà possibile consultarlo online e, se si desidera, scaricarlo e stamparlo.

Un testo consigliato (non obbligatorio) sull'argomento è:

Competitive Programming 4: The Lower Bound of Programming Contest in the 2020s.
Steven Halim, Felix Halim, Suhendry Effendy
Libri 1 e 2. Lulu Press, Incorporated, 2018 e 2020.
ISBN: 1716745527 e 1716745519.