Ingegneria e Scienze Informatiche Magistrale (2023)

87474 - Distributed Systems

PDF, ADOC.

Learning outcomes

At the end of the course, students got acquainted with the fundamental issues of distributed systems, the computational models capturing their essence, and the technologies currently helping facing them in the most systematic and effective way. In particular, students become familiar with the fittest solutions, technologies, architectures, and methodologies to design distributed systems, and is capable of

devising out the most critical aspects of distributed systems coming from physical distribution

determining the most proper methodological approaches

selecting the fittest technologies for implementing the solutions detected

Course contents

Case Studies

The CAP Theorem. Availability, Consistency, Failure in Distributed Systems

The Problem of Consensus in Distributed Systems

Distributed Ledger Technology. Blockchain as Middleware

Representational State Transfer (ReST)

Logical Clocks

Simple Agents in JADE

Coordination in Linda

Code Mobility

General Issues of Distributed Systems

Why Distributed Systems?

Replication & Consistency in Distributed Systems

Dependability in Distributed Systems

Roots of Distributed Systems. Computation in Space & Time

Definitions & Goals for Distributed Systems

Sorts of Distributed Systems

Modelling Distributed Systems. Software & System Architectures

Modelling Distributed Systems. Process Algebra

Computing with Time

Agents for Distributed Systems

Coordination of Distributed Systems

Computing with Space

Technologies for Distributed Systems

Build Automation

Containers

Asynchronous Programming

Sockets

Presentation

Web Services

Consensus

Queues

81609 - Languages, Compilers and Computational Models

PDF, ADOC.

Learning outcomes

At the end of the course the student knows:

- foundational aspects of logics, automata, regular expressions and grammars
- techniques for the design and implementation, via compilers, of programming languages
- languages, models and techniques for the description and property verification of concurrent and distributed software systems.

Course contents

Automata and formal languages:

- finite state automata
- regular expressions and their relationship with finite state automata
- context free grammars
- pushdown automata and their relationship with context free grammars
- Turing machines
- recursive and recursively enumerable languages and their relationship with Turing machines

Basic logic:

- propositional logic
- predicate logic

Compilers:

- lexical analyzers
- syntactic analyzers
- static program checking
- code generation

Computational models:

- transition systems for the modeling of system behavior
- temporal logics for property specification and verification
- concurrent finite state processes with synchronization and interleaving
- Petri nets

Rea

81610 - Machine Learning

PDF, ADOC.

Learning outcomes

Providing the student with the concepts necessary: - to understand and apply machine learning approaches; - implement classification, regression and clustering algorithms to solve problems in different applicative fields; use neural networks and other deep learning techniques.

Course contents

Artificial Intelligence and Machine Learning

Supervided and Unsupervised Learning

Classification and Regression

Classifiers: Bayes, k-Nearest Neighbor, Support Vector Machines, Multiclassifiers

Clustering (K-means, EM) and Dimensionality Reduction (PCA, DA)

Neural Networks (NN)

Introduction to Deep Learning

Convolutional Neural Networks (CNN)

Recurrent Neural Networks (RNN)

Reinforcement Learning (RL)

09679 - Data Base Sistems

PDF, ADOC.

Learning outcomes

The student

- has a in depth knowledge about managing, organizing and planning an Information System. these organizational and managerial competences are complemental to the technical ones in the database area;
- knows the techniques for analyze and classify software modules in the information system contex;
- carries out feasibility studies and Business Process Reengineering project;
- carries out database integration and master data projects;
- acquires practical skills on the previous topics through interactive business cases and seminars.

Course contents

Introduction to Information Systems

IS classification

Il portafoglio applicativo aziendale

CIMsystems

ERP systems

CRMsystems

Seminar: Sage - ERP

Methods for IS design

Resources and processes

IS planning

Business Process Reengineering

Feasibility study

Risk analysis

cost-Benefit Analysis

Architectural analysis of IS

Scope statement

ICT project management

Database integration and Master Data Management

93470 - Cybersecurity

PDF, ADOC.

Learning outcomes

At the end of the course, the student knows the basic principles of computer security and he/she is able to identify the main problems of computer and network security. He/she gets to understand and explain the main protocols and mechanisms used for securing communications and data transfer. He/she is able to perform a critical evaluation of the security of a computing infrastructure and to suggest the best countermeasures to mitigate the vulnerabilities, reduce the risk and increase the resilience to attacks. He/she is also capable of contributing to the design of systems that are secure by design and understanding the basic problems of computer forensics. Finally, he/she is able to design and contribute to the enhancement of the security of devices exposed to the Internet.

Course contents

Computer and network security basics. Threats, risks, attacks, and assets. Security functional requirements.

Small introduction to cryptography. Symmetric Encryption. Public-Key Encryption. Digital Signatures and Key Management.

User authentication and authentication-related problems.

Access control.

Malicious software.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks.

Intrusion Detection Systems (IDS).

Design and implementation of Firewalls.

Security aspects in design and implementation of software.

Security management and risk assessment. Management. Risk analysis and evaluation. Design and implementation of security policies. Social engineering and human-in-the loop. Auditing.

Basic computer forensics and anti-computer forensics techniques.

Hardening of computing devices.

81932 - Big Data

PDF, ADOC.

Learning outcomes

At the end of the course, the student:- Knows the applications of Big Data technologies and the respective challenges - Knows the available hardware and software architectures to handle Big Data - Knows the techniques to store the data, the programming languages and paradigms generally adopted in this kind of systems - Knows the design methodologies for the different kinds of applications in the area of Big Data - Acquires practical expertise in using the different technologies through laboratory and projects. In particular, the main technologies used in practical exercises will be NoSQL databases and the Hadoop platform: Hive, Spark, Tez, Dremel, Giraph, Storm, Mahout, and Open R

Course contents

Requirements

A prior knowledge of relational databases, Java and Scala programming languages, and Unix-like systems is required to attend the course. Attendance of Business Intelligence and Data Mining courses is encouraged.

Classes and teaching material are in English. A good comprehension of English is thus required to use the material and interact during class. Exam can be given in Italian.

Modules

The course is split into two modules.

The first module (20 hours) is mostly theoretical and it is shared with the Master's Degree in Digital Transformation Management (corresponding to Module 1 of Big Data and Cloud Platforms [https://www.unibo.it/it/didattica/insegnamenti/insegnamento/2022/466768]).

The second module (30 hours) is mostly practical and it is exclusive to the Master's Degree in Computer Science and Engineering.

Course Contents

1. Introduction to the course and to Big Data: what they are and how to use them
2. Cluster computing to handle Big Data

Hardware and software architectures

The Apache Hadoop framework and its modules (HDFS, YARN)

Hadoop-specific data structures (Apache Parquet)

1. The MapReduce paradigm: basic principles, limitations, design of algorithms
2. The Apache Spark system

Architecture, data structures,basic principles

Data partitioning and shuffling

Optimization of the computation

1. SQL on Big Data with Spark SQL
2. Data streaming

The architecture to handle data streaming

Approximated algorithms in the streaming context

1. NoSQL databases
2. Handling Big Data in the Cloud

Cluster on-premises vs in the cloud

The technological stack in the cloud

Deploy of a real case study on a cloud provider

1. Designing non-trivial problems under the MapReduce paradigm

30376 - Business Intelligence

PDF, ADOC.

Learning outcomes

After the course, the student is skilled in business intelligence architectures and functionalities. In particular, the student is capable of designing and administrating enterprise data warehouses.

Course contents

Requirements

A prior knowledge and understanding of database systems, relational model, and SQL language is required to attend with profit this course. These notions are normally achieved by giving an exam of Databases or Information Systems.

Fluent spoken and written Italian is a necessary pre-requisite: all lectures and tutorials, and all study material will be in Italian.

Course Contents

Business intelligence:

the role of BI in the corporate information system;

the BI pyramid.

Data Warehousing:

architectures;

techniques for data analysis: reporting and OLAP;

lifecycle:

data source analysis;

requirement analysis;

conceptual design;

workload and data volume;

logical design;

design of ETL procedures;

physical design.

40720 - Data Mining

PDF, ADOC.

B0010 - Software Architecture and Platforms

PDF, ADOC.

Learning outcomes

The goal of this course is to introduces the essential concepts of software architecture and related abilities expected for a software architect in the practice of software engineering, including the application of modern architectures (e.g. Service-Oriented Architectures and REST, microservices, event-driven Architectures, cloud-computing architectures) to concrete application contexts, including cyber-physical systems and the IoT.

The student will learn to:

- analyze how a software architecture relates to an organization and identify its quality attributes;
- design software architectures using fundamental principles, patterns, and styles;
- implement, deploy and manage software architectures using proper software platforms;
- document and evaluate software architectures, using standard languages (e.g. UML) and tools.

Course contents

- Software Architecture introduction

— What do we mean for Software Architecture and Architectural Thinking

— The Role of Software Architect

- Software Architecture foundations

— Basic Concepts and Design Principles

— Architectural Styles and Patterns

--- main examples including Layered Architecture, Clean Architecture, Service-Oriented Architecture, Microservices, Event-Driven Architecture

- Software Architecture and the Software Engineering Process

— Architectural Significant Requirements

— Designing, implementing, evaluating, documenting, evolving a software architecture

— Software Architectures and Domain-Driven Design

— Software Architectures and DevOps

- Software Architectures in Practice - selected domains and case studies

— Pervasive Computing, Internet of Things and Web of Things - with a focus on Digital Twins

— Enterprise Applications and Cloud Computing

— Autonomous/Intelligent Systems - with a focus on Intelligent Agents and Cognitive Architectures

B0011 - Software Process Engineering

PDF, ADOC.

Learning outcomes

By the end of the course, the student is expected to master advanced techniques for organising the software development process, and in particular:

- ability to set up, evolve, and maintain an agile development process, including build automation, multi-platform and multi-target testing, continuous integration, and continuous delivery;
- advanced knowledge of modern version control systems
- understanding of the software licensing (with a focus on open source products) and of the existing models of software versioning;
- capacity of performing a domain-first analysis in a technology-independent fashion, leveraging techniques of Domain-Driven (DDD) and Model Driven (MDD) development;
- knowledge of supporting tools for DDD/MDD: creation of domain-specific languages (DSLs) and corresponding code generators, development of language-internal DSLs in modern programming languages.

Course contents

MODULE 1

Introduction to Kotlin

Internal domain-specific languages in Kotlin

Build automation (Gradle as reference tool)

Automated Quality Assurance

Software versioning

Software licensing

Teamwork organization via git

Advanced version control (submodules, rebasing, cherry-picking, squashing, stashing)

Continuous integration (GitHub Actions)

Continuous delivery/deployment

MODULE 2

Domain-Driven Design

Model-Driven Design

External and Internal Domain-Specific Languages (DLSs)

Containerization

Orchestration

Multi-Platform Programming (Kotlin)

Bug hunting and Performance Engineering

29443 - Computer Vision

PDF, ADOC.

Learning outcomes

The course aims at providing the notions and tools necessary for the design and implementation of automatic systems able to analyze digital images for object detection and recognition. In particular, the course focuses mainly on the techniques for feature extraction from digital images and the application of these techniques to typical problems in computer vision such as segmentation, localization, classification and similarity searches. Both traditional approaches as well as deep-learning based solutions will be analyzed, with examples in real-life applications.

Course contents

Basic techniques for digital image processing and filtering

Feature extraction

Color features:

- color histograms and similarity metrics;
- color moments

Texture features:

- gray-level co-occurrence matrix and related measures (enthropy, contrast, homogeneity, etc..);
- Gabor filters: filter banks;
- Haar features: integral image and efficient feature extraction;
- Local Binary Pattern;

Shape features:

- Object countour extraction and one-dimensional shape representations;
- Shape descriptors, Fourier descriptors;
- Invariant moments.

Handcrafted features vs Representation learning

Image stitching, 2D image registration and Visual SLAM

Keypoints and local descriptors:

- Keypoint detection: Harris corner detector;
- Scale invariant detectors: Harris Laplace, Laplacian of Gaussian, Difference of Gaussian;
- Keypoints and descriptors: SIFT, SURF, BRIEF, Histogram of Oriented Gradients.
- Ransac algorithm for feature matching.
- Application to robotics: Visual SLAM (Simultaneous Localization and Mapping)

Semantic segmentation in digital images

Color-based segmentation techniques, Mean Shift algorithm;

Deep learning based techniques with applications in satellite and medical images analysis.

Recognition “in the wild”

Object detection/classification

- Color, texture and shape features for content-based image retrieval;
- Bag of visual Words;
- Feature-based Rigid Template matching and applications to object detection and recognition (e.g. grocery products)
- Hough Transform;
- Deep learning techniques for object detection and recognition (e.g. pedestrian and road sign recognition, object recognition for robotic vision, face detection and recognition).

Video surveillance and video analysis

Basic techniques for frame subtraction and background modeling

Approaches for object/person tracking and crowd analysis;

Human activity detection and recognition.

Readin

B0009 - Advanced Software Modelling and Design

PDF, ADOC.

Learning outcomes

The goal of this course is to enhance the abilities of prospective software architects to construct models of software systems in a variety of contexts (including cyber-physical systems and the IoT), and turn them into concrete designs of reliable and effective systems and applications.

The student will learn to:

- model and design computational systems featuring non-determinism, stochasticity, large-scaleness, and intelligence;
- adopt advanced programming language constructs, techniques and design patterns to address complex software system development;
- rigorously address system requirements adopting techniques of software testing, simulation and verification.

Course contents

The content of the course is organised around a selected set of topics that concern sound modelling and design (and implementation thereof) of modern, complex software systems that include:

large-scale distributed systems

software systems featuring complex domains

software systems with intelligent components

simulation of distributed cyber-physical systems

Such topics are covered by the following didactic modules:

high-level patterns of system programming in Java and Scala: component programming, monads, effects

full test-driven system development: testing coverage, TDD, Acceptance TDD and Behavioural-Driven Development, property-based testing

systematic validation: model-checking (MC), probabilistic MC, approximate MC, simulation

large-scale system modelling: Petri-nets, non-determinism, networks of devices, chemical-oriented models

probability and adaptiveness: discrete-time markov chains (DTMC), continuous-time markov-chains (CTMC)

self-organisation and macro-programming: patterns and aggregate computing

decision processes and learning: markov decision processes, reinforcement learning (RL), deep RL, multiagent RL

program synthesis with AI: LLMs, role in testing, role in program completion

91411 - Intelligent Robotic Systems

PDF, ADOC.

Learning outcomes

At the end of the course, students have acquired knowledge and competencies for designing a system composed of one or more robots (here, we call "robot" an autonomous system which exists in the physical world, can sense its environment and can act on it to achieve some goals). In particular, students know the main models, methods, architectures and tools for programming robots equipped with nontrivial computational and cognitive capabilities.

Course contents

Introduction to robotics

Brief history of robotics

Notions of robot and its behavior in a physical environment

Main issues in intelligent robotic systems design

Behavior-based robotics

Overview of main paradigms for coordinating behaviours

The Subsumption Architecture

Motor schemas

Fuzzy logic and fuzzy systems

Behaviour trees

Experimental evaluation and parameter tuning of control software for robots: practical guidelines

Artificial Evolution and Evolutionary Robotics

Automatic design of robot programs

Swarm robotics

Robot learning

Associative learning

Reinforcement learning

Value-based learning and intrinsic motivation

Deliberative control

Informed search algorithms. A* and its variants for path planning problems

Robot planning: main definitions and principles

STRIPS and its extensions

Nonlinear planning

Navigation problems and main solution approaches

Lab activities

Experiments with robotic simulators with the aim of experiencing with the various kinds of control and testing the knowledge acquired.

R

93669 - Intelligent Systems Engineering

PDF, ADOC.

Learning outcomes

At the end of the course, students get acquainted with the fundamental issues of intelligent systems, the most relevant computational models and technologies, and the most effective methods. In particular, students become familiar with the fittest solutions, languages, technologies, architectures, and methodologies to design intelligent systems, and are capable of

- devising the problems requiring artificial intelligence techniques for their solution;
- determining the most proper conceptual and methodological approaches;
- selecting and integrating the fittest technologies for implementing the solutions detected.

Course contents

Case Studies

ChatGPT—Beyond the Turing Test

Autonomy in Biology

Programming Intentional Agents in AgentSpeak(L) & Jason

Natural Language Processing

General Issues of Intelligent Systems

Drivers for Intelligent Systems

On Autonomy. Concepts & Definitions

Agents for Intelligent Systems Engineering

Artificial Intelligence. A Bird’s Eye View

Automated Reasoning

Reasoning Agents

Logic & Computation

Planning for Intelligent Agents

Self-Organising Systems

Nature-Inspired Coordination & Self-Organisation

Interacting with Autonomous Systems: Conversational Informatics

Simulation & Multi-Agent Systems: An Introduction

Scientific Competences

Sources of Scientific Literature for Intelligent & Autonomous Systems

Systematic Literature Review. A Methodology for Scientific Surveys

Technologies for Intelligent Systems

Knowledge Representation

Inference

Perception and Actuation

Planning with STRIPS

Programming Intentional Agents: Exercises in Jason

eXplainable Artificial Intelligence (XAI): A Gentle Introduction

Re

95638 - Operational Analytics

PDF, ADOC.

Learning outcomes

Operational analytics, a specific type of business analytics, is focused on the analysis of business processes to the end of creating competitive advantage by means of operational data analysis and of the application of analytical algorithms. The course concentrates on the algorithmic side,

specifically presenting predictive analytical techniques, forecasting future data on the basis of available time series, and prescriptive analytical techniques, defining optimized usage of available resources. Real world cases will be studied and used as testbed for self-developed systems.

Course contents

The course is part of a data science curriculum and provides some tools for predicting short/medium-term management data (predictive analytics) and optimizing scarce resource allocation processes based on predicted data (prescriptive analytics). Elements of predictive analytics and heuristic optimization will be presented and integrated.

For the predictive part, the course proposes methodologies and techniques for analyzing, modeling and predicting univariate time series, with hints of multivariate ones.

It will be shown, compatibly with the available time, that forecast data can be elements of mathematical models of management processes in which to optimize the allocation of scarce resources.

The proposed tools are meant to be used in real business application cases, actual case studies will be shown compatibly with time, and/or demanded to final projects.

The scientific content of the course relates to the knowledge needed to develop an operational analysis module on data obtained from a business information system. Specifically, the following will be presented

- brief summary of stochastic models, random variables. probability distributions
- predictive models: statistical (ARMA, ARIMA, SARIMA), neural (MLP, LSTM, perhaps SVR if time permits) and machine learning/decision tree models (ensemble, random forest, boosting)
- performance indicators, descriptive statistics, statistical significance tests.
- introductory integer programming models
- hints to meta/math-heuristic solving techniques

The technological contents will be functional to the practical implementation of the mentioned module, which will be done standalone in python, although other environments and architectures are acceptable.

A full solution will be set up in the classroom and completed independently by each student, and may constitute the project for the exam.

Read

73435 - Project Management

PDF, ADOC.

Learning outcomes

At the end of the course students:

- know the basic principles of Project Management and can manage different approaches used in software production, from the most traditional ones to the iterative and adaptive approaches typical of the “agile” project management;
- know some tools used in project management and have gained a better understanding of some practical aspects also through practical exercises;
- can participate in or manage a software development project.

Course contents

Introduction to project management.

Definition of project, program, portfolio.

Definition of scope, quality, resources and their interrelationship.

Definition of project management.

Definition of processes and knowledge areas involved in project management according to the Project Management Body of Knowledge (PMBOK).

Management of integration, scope, time, cost, quality, human resources, communications, risk, procurement, and stakeholders.

The life cycle of a project and its management: traditional, incremental, iterative, and adaptive.

Scoping process group.

Planning process group.

Launching/executing process group.

Monitoring and controlling process group

Closing process group.

Exercises.

Seminars.

Re

42500 - Semantic Web

PDF, ADOC.

Learning outcomes

At the end of the course, the student is familiar with the concepts, standards and languages that constitute the architecture of the Semantic Web, the technologies and models for the representation of metadata and ontologies, and the ways in which different resources can be integrated and utilised (e.g. according to the Linked Data approach). The course explores the new ways of organising, integrating, managing and retrieving resources that the Semantic Web and the Web of Data make available today.

Course contents

This course is aimed at providing both a solid conceptual framework on the subject of knowledge representation in the Web of Data, and the basis for the development and use of Knowledge Graphs (KGs) by means of the most widespread tools in computer science and shared standards, highlighting issues of interest and analysing some specific topics. The growth of Web contents requires the study of theories, methodologies and techniques for the conceptualisation of these contents. This course aims to provide students with an understanding of the formal assumptions, languages and technologies that allow applications to be realised, as well as specific skills in the use and development of computer systems for the semantic management of knowledge. In particular, the following topics will be addressed:

KG in the Web of Data (Semantic Web, Linked Open Data)

Semantic technologies (RDF, RDFS, logical inference, RDFa, microformats)

Querying RDF (DBPedia KG, SPARQL)

Knowledge representation using ontologies (OWL)

KG applications (ontologies, KG programming, visualisation and analytics)

Advanced KG applications (KG embeddings, KG completion, KG mappings and alignment, semantic search)

Graph machine learning

72521 - Internet Routing and Trasport: Protocols and Performance

PDF, ADOC.

90074 - Smart Vehicular Systems

PDF, ADOC.

Learning outcomes

Students taking this class: - Will An extended knowledge of challenges and opportunites in the area of vehicular communications including V2V and V2I. The class material will include real world examples and deployments. - Will understand impact of mobility and propagation on Vehicular Systems performances - Will learn vehicular application scenarios and their evolution - Will learn how design and model for connected/autonomous vehicular systems.

Course contents

Autonomy, Mobility and AI are the biggest innovations posed to affect our life in the years to come. Th

This course will provide a survey of mobile and autonomous systems with the goal of understanding the building blocks for autonomous mobile systems and their principle of operation. The course will explore a wide range scenarios to outline how in the current stage of technology is in its infancy and there is not yet a general solution.

Pre-requisites for the class is an excellent understanding of algorithms, AI concepts, and a good understanding of systems. The class is designed to be heavy on projects therefore good programming skills and ability to work with Linux and RT os are a must; furthermore a good knowledge of AI models, tools and techniques is preferential (i.e. Yolo, Tensor RT, etc). Knowledge of robot operating system (ROS), Nvidia CUDA/Jetson programming, and introductory level understanding of control systems are a plus.

The goal is to teach autonomous mobile systems fundamental tradeoffs and techniques to equip students with a deep understanding of how the autonomous vehicular systems have developed, and how they will evolve in the near future.

A successful student in this class will gain skills on:

Understand the challenges and opportunities offered by advances in autonomous vehicles and robotics.

Gain in-depth knowledge of advanced computer networking and telecommunications issues applied to autonomous connected vehicles including new network architectures beyond the Internet.

Analyze the requirements for a given autonomous system and select the most appropriate system architecture and technologies.

Gain an introductory level familiarity with tools for autonomous systems including simulation and practical systems.

93668 - 3D Image Analysis and Computer Vision Systems

PDF, ADOC.

Learning outcomes

The objective of this Course is to form an engineer able to design computer vision systems working in the real world, even in real time, for industrial, scientific and play purposes. The systems exploit automatic analysis of 3D image sequences in a number of application fields including machine vision, automotive, automatic quality control, zero-defect analysis, predictive maintenance, security, medical and biomedical imaging, aerospace imaging, precision agriculture, cultural heritage. At the end of the Course, students are able to apply the acquired skills:

To apply computer vision in critical multidisciplinary fields, including medical imaging

To use cameras to perform high-accuracy 3D measurements

To realize augmented and mixed reality applications

To use vision sensors in IoT applications

To perform 3D scene reconstruction from moving aerial or terrestrial vehicle

To perform 3D scanning, also for 3D printing

To perform 3D defect analysis in industry

To design real time computer vision systems on parallel/distributed architectures

Course contents

- Computer vision: examples of systems and applications
- From sensor to image: basics of optics
- Real time image sequence and video analysis
- Semantic extraction of multi-dimensional features
- Perspective: a 2D projection of the real world
- From perspective to 3D: camera calibration, stereoscopy, triangulation
- From motion to 3D: the cloud points
- Cloud points and 3D geometric object representation
- Principles of real time processing (high throughput)
- Effective GPU programming

Case studies: the ongoing projects

91250 - Deep Learning

PDF, ADOC.

Learning outcomes

The course aims at providing advanced skills (both theoretical and practical) on machine learning and, in particular, on deep learning.

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

in-depth train and optimize deep learning approaches;

choose and customize the most appropriate techniques to be used in real application scenarios;

use advanced deep learning techniques.

Course contents

Introduction to deep learning

Linear algebra, calculus and automatic differentiation

Artificial neural networks

Backpropagation

Optimization algorithms

Convolutional Neural Networks (CNN)

Recurrent Neural Networks (RNN)

Transformers

AutoEncoders (AE)

Generative models

Reinforcement Learning (RL)

Natural Language Processing (NLP) (a practical example)

93667 - Laboratory of Network Programmability and Automation

PDF, ADOC.

Learning outcomes

The student will learn the modern network programming and automation methodologies and the

related enabling technologies. The student will learn how to use a platform for virtual network

programming and automation in a cloud computing environment and a control plane for software defined networks. Moreover the student will learn how to use specific data models for the design and for the programming of network services, according to the current standards.

Course contents

Network Virtualization and cloud computing

Network virtualization: Linux namespaces, virtual bridges and virtual switches

Virtual networking in cloud computing: examples with Docker and OpenStack

Software Defined Networking

SDN architecture

The OpenFlow protocol

Examples of SDN programming with OpenFlow

Network Function Virtualization

The ETSI NFV-MANO architecture and standards

An example of orchestration platform: OpenSource Mano (OSM)

Implementation of network service descriptors with OSM

Programming the data plane

The P4 language

Example of programming switches behavior with P4