REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

OFFRE DE FORMATION
INGENIEUR D'ETAT

Intégration du module Lean Startup

Domaine	Filière	Spécialité
Mathématiques et Informatique	Informatique	Intelligence Artificielle

Le 16/09/2026

رئيس اللجنة البيداغوجية الوطنية
لميدان الرياضيات والإعلام الآلي
أ.د بابا حنيني محمد شوقي

Année universitaire : 2026/2027

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

عرض تكوين مسار مهندس دولة

إضافة مقياس المؤسسات الناشئة اللينة

الميدان	الشعبة	التخصص
رياضيات و إعلام آلي	الإعلام الآلي	الذكاء الاصطناعي

رئيس اللجنة البيداغوجية الوطنية
لميدان الرياضيات والإعلام الآلي
أ.د بابا حنيني محمد شوقي

Le 16/09/2026

السنة الجامعية: 2027/2026

II – Fiche d'organisation semestrielle des enseignements

(Prière de présenter les fiches des 6 semestres)

1- Semestre 5 :

Unité d'Enseignement	VHS	V.H hebdomadaire				Coeff.	Crédits	Mode d'enseignement		Mode d'évaluation	
	14 sem	C	TD	TP	Autres			A distance	En présentiel	Continu	Examen
UE fondamentales											
UEF1(O/P)											
Architecture et administration des BDD	63H	03H00		01H30		4	6		X	40	60
Compilation	63H	01H30	01H30	01H30		4	6		X	40	60
UEF2(O/P)											
Programmation Linéaire et Dynamique	63H	01H30	01H30	01H30		3	4		X	40	60
Analyse numérique 1	63H	01H30	01H30	01H30		2	4		X	40	60
Fondements de l'IA	42H	01H30		01H30		3	5		X	40	60
UE découverte											
UED1(O/P)											
Génie logiciel	63H	01H30	01H30	01H30		2	3		X	40	60
UE transversales											
UET1(O/P)											
Module optionnel : Développement mobile Interface Homme Machine Conception de jeux vidéo	42H	01H30		01H30		2	2		X	40	60
Total Semestre 5	399H	12H00	06H00	10H30		**20**	**30**				

- Volume hebdomadaire = 28H30 / semaine
- 7 Matières par semestre.

2- Semestre 6 :

Unité d'Enseignement	VHS	V.H hebdomadaire				Coeff.	Crédits	Mode d'enseignement		Mode d'évaluation	
	14 sem	C	TD	TP	Autres			A distance	En présentiel	Continu	Examen
UE fondamentales											
UEF1(O/P)											
Réseaux avancés	63H	01H30	01H30	01H30		3	6		X	40	60
Système d'Exploitation : Synchronisation et Communication	84H	03H00	01H30	01H30		4	6		X	40	60
Gestion de projets	63H	01H30	01H30	01H30		4	4		X	40	60
UEF2(O/P)											
Programmation WEB	42H	01H30		01H30		3	4		X	40	60
Analyse numérique 2	63H	01H30	01H30	01H30		2	4		X	40	60
UE découverte											
UED1(O/P)											
Introduction à la sécurité Informatique	63H	01H30	01H30	01H30		2	4		X	40	60
UE transversales											
UET1(O/P)											
Ethique de l'IA	21H	01H30				2	2		X		100
Total Semestre 6	399H	12H00	07H30	09H00		**20**	**30**				

- Volume hebdomadaire = 28H30 / semaine
- 7 Matières par semestre.

3- Semestre 7 :

Unité d'Enseignement	VHS	V.H hebdomadaire				Coeff.	Crédits	Mode d'enseignement		Mode d'évaluation	
	14 sem	C	TD	TP	Autres			A distance	En présentiel	Continu	Examen
UE fondamentales											
UEF1(O/P)											
Représentation des connaissances et raisonnement	63H	01H30	01H30	01H30		4	6		X	40	60
Calcul haute performance	63H	01H30	01H30	01H30		3	5		X	40	60
UEF2(O/P)											
Machine Learning	63H	03H00		01H30		4	6		X	40	60
Modélisation et simulation	63H	01H30	01H30	01H30		3	5		X	40	60
UE méthodologie											
UEM1(O/P)											
Business Intelligence	42H	01H30		01H30		2	3		X	40	60
Recherche opérationnelle	63H	01H30	01H30	01H30		2	3		X	40	60
UE transversales											
UET1(O/P)											
Techniques de rédaction	42H	01H30		01H30		2	2		X	40	60
Total Semestre 7	399H	12H00	06H00	10H30		**20**	**30**				

- Volume hebdomadaire = 28.5H / semaine
- 7 Matières par semestre.

4- Semestre 8 :

Unité d'Enseignement	VHS	V.H hebdomadaire				Coeff.	Crédits	Mode d'enseignement		Moded'évaluation	
	14sem	C	TD	TP	Autres			A Distance	En présentiel	Continu	Examen
UE Fondamentales											
UEF1(O/P)		**6h00**		**4h30**		**6**	**10**				
Traitement Automatique du Langage Naturel	63h	03h00		01h30		3	5		X	40	60
DeepLearning	84h	03h00		03h00		3	5		X	40	60
UEF2(O/P)		**03h00**	**03h00**	**03h00**		**6**	**10**				
Sécurité des données	63h	01h30	01h30	01h30		3	5		X	40	60
Analyse et Traitement d'image	63h	01h30	01h30	01h30		3	5		X	40	60
UE Méthodologie											
UEM1(O/P)		**3h00**		**6h00**		**6**	**8**				
Technologies de Calcul Distribué	42h	01h30		01h30		2	3		X	40	60
Traitement de données massives	42h	01h30		01h30		2	3		X	40	60
Projet pluridisciplinaire				3h		2	2		X	100	
UE Transversale											
UET1(O/P)		**1h30**		**1h30**		**2**	**2**				
Lean Startup	42h	01h30		01h30		2	2			80	20
TotalSemestre8	441h	13h30	03h00	15h00		**20**	**30**				

- Volume hebdomadaire = 28H30 / semaine
- 7 Matières par semestre.

5- Semestre 9 :

Unité d'Enseignement	VHS	V.H hebdomadaire				Coeff.	Crédits	Mode d'enseignement		Mode d'évaluation	
	14 sem	C	TD	TP	Autres			A distance	En présentiel	Continu	Examen
UE fondamentales											
UEF1(O/P)											
Visualisation de données	63H	01H30		03H00		4	6		X	40	60
Vision par ordinateur	42H	01H30		01h30		3	5		X	40	60
UEF2(O/P)											
Generative AI	63H	01H30		03h00		4	6				
Méthodes bio-inspirées	63H	03H00		01H30		3	5		X	40	60
UE découverte											
UED1(O/P)											
Architectures et technologies blockchain	42H	01H30		01H30		2	3		X	40	60
Recherche d'information	63H	01H30	01H30	01H30		2	3		X	40	60
UE transversales											
UET1(O/P)											
Séminaire & workshops	21H	01H30				2	2		X		100
Total Semestre 9	357H	12H00	01H30	12H00		**20**	**30**				

- Volume hebdomadaire = 25H30 / semaine
- 7 Matières par semestre.

6- Semestre 10 :

Domaine : Mathématiques/ informatique
Filière : Informatique
Spécialité : Intelligence Artificielle

Sujet de recherche ou Stage en entreprise sanctionné par un mémoire et une soutenance.
L'attribution des sujets doit se faire au début de l'année (Octobre)

	VHS	Coeff	Crédits
Travail Personnel	140	/	/
Stage e entreprise	280	/	/
Séminaires	/	/	/
Autre (préciser)	/	/	/
Total Semestre 10	420	20	**30**

5- Récapitulatif global de la formation : (indiquer le VH global séparé en cours, TD, pour les 06 semestres d'enseignement, pour les différents types d'UE)

UE VH	UEF	UEM	UED	UET	Total
Cours	43.5	6	6	6	61.5
TD	16.5	1.5	4.5		22.5
TP	37.5	6	6	6	55.5
Travail personnel					
Autre (préciser)	30				
Total					**169.5**
Crédits	147	12	13	8	**180**
% en crédits pour chaque UE	81.67%	6.67%	7.22%	4.44%	100%

III Programme détaillé par matière

Semestre 5
Unité d'enseignement : UEF1
Matière : Compilation
Coefficient : 4

Objectifs de l'enseignement

L'objectif de ce cours sont multiples :

- Maitrise des langages de programmation : En effet, il est primordial de comprendre le fonctionnement des compilateurs afin de programmer efficacement,
- Conception et description de langages dédiés à des applications spécifiques. Le processus de compilation est souvent associé et restreint aux langages de programmation (procéduraux, orientés objets) alors qu'il peut être généralisé à une multitude de langages tels que : les langages logiques, fonctionnels, formels….

Connaissances préalables recommandées

Théorie des langages, Architecture des ordinateurs, Algorithmique, Système d'exploitation, Logique Mathématique.

Contenu de la matière :

Chapitre I : Généralité

1. Structure d'un compilateur
   - 1.1. Phases d'analyse
     - a. Analyse lexicale
     - b. Analyse Syntaxique
     - c. Analyse sémantique
   - 1.2. Phases de production
     - a. Génération de code
     - b. Optimisation de code
   - 1.3. Phases parallèles
     - a. Gestion de la table de symboles
     - b. Gestion des erreurs
2. Compilateurs, pré-compilateurs, traducteur et interpréteur
3. Rappels sur la théorie des langages
   - 3.1. Les expressions régulières
   - 3.2. Les automates
   - 3.3. Equivalence entre expressions régulières et automates
   - 3.4. Les grammaires

Chapitre I : Analyse lexicale

1. Introduction
2. Mise en œuvre d'un analyseur lexical
   - 2.1. Spécification des entités lexicales
   - 2.2. Correspondance entre expressions régulières et automates
   - 2.3. Algorithme d'analyse
   - 2.4. Création et gestion de la table des symboles
3. Gestion des erreurs lexicales

Chapitre II : Analyse syntaxique

1. Introduction
2. Les concepts de base
   - 2.1. Les formes normales des Grammaires
   - 2.2. Grammaire ε-libres et sans cycles

2.3. Elimination de la récursivité à gauche directe et indirecte dans une grammaire
2.4. Factorisation d'une Grammaire
3. Les méthodes d'analyse descendantes
3.1. Les méthodes d'analyse non déterministes
3.2. Les méthodes d'analyse déterministes
a. La méthode d'analyse LL(1)
- Calcul des débuts et des suivants
- Construction la table d'analyse LL(1)
- Algorithme d'analyse LL(1)

b. La Descente récursive
4. Les méthodes d'analyse Ascendantes déterministes
4.1. Analyse LR(1**)** par les contextes et les items
a. Construction de l'ensemble des contextes et des items LR(1)
b. Construction de la table d'analyse LR(1)
c. Algorithme d'analyse LR(1)
d. Analyse LR(K)
4.2. Analyse Simple LR(1) par les contextes et les items
a. Construction de l'ensemble des items LR(0)
b. Construction de la table d'analyse SLR(1)
c. Algorithme d'analyse SLR(1)
4.3. Analyse LALR(1) par les contextes et les items
a. Construction de l'ensemble des contextes et des items LR(1)
b. Construction de la table d'analyse LALR(1)
c. Algorithme d'analyse LALR(1)
5. Mise à jour de la table des symboles
6. Gestion des conflits dans le cas des tables d'analyse multi-définies
7. Relations entre les méthodes de l'analyse syntaxique
8. Gestion des erreurs syntaxiques
8.1. Mode panique
8.2. Récupération au niveau syntagme

Chapitre III : Les formes intermédiaires

1. Introduction
2. La forme post-fixée
2.1. Evaluation des expressions arithmétiques et logiques
2.2. L'instruction d'affectation
2.3. Le branchement inconditionnel
2.4. Le branchement conditionnel
2.5. Les Instructions répétitives
2.6. Déclaration d'un tableau multidimensionnel
2.7. Référence à un élément d'un tableau multidimensionnel
3. Les quadruplets
3.1. L'instruction d'affectation
3.2. Le branchement inconditionnel
3.3. L'instruction conditionnelle
3.4. Les Instructions répétitives
3.5. Déclaration d'un tableau multidimensionnel
3.6. Référence à un élément d'un tableau multidimensionnel
4. Les triplets directs et indirects
4.1. L'instruction d'affectation
4.2. Le branchement inconditionnel
4.3. L'instruction conditionnelle

4.4. Les instructions répétitives
4.5. Déclaration d'un tableau multidimensionnel
4.6. Référence à un élément d'un tableau multidimensionnel

5. Les Arbres abstraits
   5.1. L'instruction d'affectation
   5.2. Le branchement inconditionnel
   5.3. L'instruction conditionnelle
   5.4. Les Instructions répétitives
   5.5. Bloc d'instructions BEGIN-END
   5.6. Déclaration d'un tableau multidimensionnel
   5.7. Référence à un élément d'un tableau multidimensionnel

Chapitre IV : Traduction dirigée par la syntaxe

1. Schéma de traduction dirigée par la syntaxe dans le cas de l'analyse descendante
   1.1. Les expressions arithmétiques
   1.2. Les expressions logiques
   1.3. Les structures de contrôle
   1.4. Instruction conditionnelle
   1.5. Instructions répétitives
2. Schéma de traduction dirigée par la syntaxe dans le cas de l'analyse ascendante
   2.1. Découpage de la grammaire
   2.2. Les expressions arithmétiques
   2.3. Les instructions répétitives
   2.4. Instruction du branchement inconditionnel

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Compilateurs: principes, techniques et outils - A. Aho, R. Sethi, J. Ullman – InterEditions.
2. Compilateurs - D. Grune, H. Bal, C. Jacobs, K. Langendoen - Dunod.
3. Principles of Compilers- V. Aho, J.D. Ullman - édition 2006.
4. Modern compiler implementation in ML- A.W.Appel - Cambridge University Press 1998.
5. The theory and practice of compiler writing - P.Q G. Sorenson - McGraw-Hill computer science series 1985.
6. Flex & Bison - J.Levin- O"Reilly Media 2009.

Semestre 5
Unité d'enseignement : UEF1
Matière : Architecture et administration des BDD
Coefficient : 4

Objectifs de l'enseignement

Ce module permet de cerner le niveau de stockage des données relationnelles**.** Seront présentées les contraintes d'intégrités **(**définition et représentation à l'aide d'un langage), l'organisation de données et structures d'accès, et les métadonnées, le mécanisme des vues ainsi que le contrôle d'accès.

Connaissances préalables recommandées

Les concepts généraux de BD ainsi que leur modélisation et interrogation.

Contenu de la matière :

Chapitre I : Généralités sur les SGBD

1. Présentation générale des BD et SGBD : Rappels
   - 1.1. Généralités
     - a. Définitions : BD et SGBD
     - b. Historique des SGBD,
   - 1.2. Niveaux de description et Architecture des SGBD
   - 1.3. Objectifs et fonctionnalités d'un SGBD : Le SGBD ORACLE
2. La mémoire relationnelle
   - 2.1. La mémoire de stockage
   - 2.2. Les différentes approches de stockage des données
   - 2.3. Une approche de stockage : les B-arbres
3. Gestion de l'intégrité
   - 3.1. Vue d'ensemble des contraintes d'intégrité (CI)
     - a. Les CI statiques
     - b. Les CI dynamiques : les Triggers

Chapitre II : Les méta données

1. Introduction : nécessité de description des différents types de relations d'une base de données (relation système, relation de base, relation virtuelle)
2. Les concepts de base
   - 2.1. les principaux catalogues ; Relation, Attribut, Index, Vue etc.
   - 2.2. Gestion dynamique des catalogues
   - 2.3. Le dictionnaire de données ORACLE
3. Mécanisme des vues
   - 3.1. Définition des vues
   - 3.2. Interrogation à travers les vues
   - 3.3. Mise à jour à travers les vues
4. Gestion des privilèges d'accès
   - 4.1. Privilèges système
   - 4.2. Privilèges objets
   - 4.3. Notion de rôle

Chapitre III : Sureté de fonctionnement et reprise après pannes

1. Introduction : rôle du SGBD afférent à la gestion de pannes
2. Classification des pannes
3. Le processus de reprise après panne et états d'une transaction
4. Classification des mécanismes de reprise : mise à jour immédiates ou écriture en place, mise à jour différées ou écriture non en place, etc.

5. Les utilitaires de récupération : mécanismes de sauvegarde, outils de journalisation, utilitaires de points de contrôle (Checkpoint), gestionnaire de récupération
6. Les différentes Techniques de restauration : stratégies Undo, Redo etc.

Chapitre IV : Gestion des accès concurrents

1. Transaction : Définitions et caractéristiques
2. Les problèmes de concurrence
   2.1. Mise à jour perdue
   2.2. Non reproductibilité des lectures
   2.3. Dépendance non valide
3. Gestion de la concurrence
   3.1. Principe de sérialisation
   3.2. Les trois techniques de sérialisation : (états équivalents à ordonnancement série, permutation et/ou séparation des opérations permutables et/ou compatibles, et graphes de précédence non cyclique)
   3.3. Les méthodes de sérialisation
      a. Par estampillage : estampillage simple, à deux estampilles
      b. Par verrouillage : différents types de verrouillage, graphe d'attente, V2P, verrous indéfinis

Chapitre V : Optimisation

1. Typologie de requêtes
2. Les différentes phases de traitement d'une requête
3. Optimisation de requêtes
   3.1. Méthode syntaxique
      a. Règles de transformation des arbres algébriques
      b. Algorithme simple de génération d'un plan de requêtes
   3.2. Estimation de coûts
      a. Utilisation de données statistiques
      b. Techniques d'implémentation des opérateurs : cas de la jointure

Chapitre VI : Bases de données distribuées et NoSQL

1. Les bases de données distribuées
   1.1. Qu'est-ce qu'une base de données distribuée ?
   1.2. Avantages et inconvénients des bases de données distribuées
   1.3. Comparaison avec les bases de données centralisées
2. Systèmes de gestion des bases de données distribuées (SGBDD)
   2.1. Définition et concepts de base
   2.2. Architectures des SGBDD (client-serveur, peer-to-peer, multi-agent)
3. Techniques de répartition des données
   3.1. Fragmentation, allocation et réallocation des données
   3.2. Réplication des données : types et stratégies
4. Fragmentation des données
   4.1. Définition
   4.2. Fragmentation horizontale
   4.3. Fragmentation verticale
   4.4. Fragmentation mixte
5. Architectures de schéma de BDDD
   5.1. Systèmes homogènes vs hétérogènes
6. Évaluation des requêtes distribuées
   6.1. Algorithmes d'optimisation des requêtes
   6.2. Exécution et planification des requêtes
7. Émergence des bases de données NoSQL
   7.1. Origine et besoin des bases de données NoSQL

7.2. Comparaison avec les bases de données relationnelles
8. Quelques exemples des solutions NoSQL
   8.1. Clé-valeur : Riak, Redis
   8.2. Orientée colonne : Cassandra, HBase
   8.3. Orientée document : MongoDB, CouchDB, ElasticSearch
   8.4. Orientée graphe : Neo4j, HypergraphDB
9. Les bases de données MongoDB
   9.1. Définition
   9.2. Format JSON
   9.3. Les opérations CRUD
10. Évaluation des performances
   10.1. Indicateurs de performance (latence, débit, disponibilité)
   10.2. Méthodologies de benchmark et de tests de charge
11. Techniques de scalabilité
   11.1. Scalabilité horizontale vs verticale
   11.2. Techniques de sharding et partitionnement

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. J. Date. Introduction aux bases de données. Thomson publishing France 6ième édition. 1998
2. C. Delobel et M. Adiba : bases de données et systèmes relationnels. Dunod 1982
3. T. Connoly et Corolyn Begg. Systèmes de bases de données : approche pratique de conception de l'implémentation et de l'administration. Eyrolles 2005
4. Frederic Brouard SQL CampusPress 2001
5. Georges Gardarin : Les BD systèmes et leurs langages Eyrolles 2003
6. Nacer Boudjlida : Gestion et Administration des Bases de Données. Application a Sybase et Oracle. Dunod, Collection Sciences Sup. 2003.
7. Heurtel, Olivier. Oracle 12c: administración. Ediciones ENI, 2015.

Semestre 5
Unité d'enseignement : UEF2
Matière : Analyse numérique1
Coefficient : 2

Objectifs de l'enseignement

Le module d'analyse numérique est un module de base dans le domaine du calcul scientifique. Le calcul scientifique étant défini comme la branche qui permet d'implémenter une solution d'un problème sur un calculateur. L'objectif du module est l'acquisition des méthodes de discrétisation, des algorithmes de résolution des problèmes spécifiques posés dans le domaine de l'ingénierie informatique, l'analyse des algorithmes (rapidité, précision et souplesse), et enfin l'estimation des erreurs commises dans les modèles mathématiques employées ainsi que dans leur implémentation.

Connaissances préalables recommandées

Pour suivre un module d'Analyse Numérique, les étudiants doivent maîtriser l'algèbre linéaire, le calcul différentiel et intégral, et les équations différentielles. Des notions d'analyse de Fourier et d'analyse numérique de base sont également recommandées.

Contenu de la matière :

Chapitre I : Analyse d'erreurs

1. Introduction générale
2. Erreurs de Modélisation
3. Représentation des nombres sur un ordinateur :
   - 3.1. Représentation des entiers signés
   - 3.2. Représentation des réels.
   - 3.3. Erreurs dues à la représentation.
4. Arithmétique Flottante
   - 4.1. Opérations élémentaires
   - 4.2. Opérations risquées
   - 4.3. Evaluation des polynômes
5. Erreurs de Troncatures

Chapitre II : Equations non linéaire

1. Introduction générale
2. Méthode de la bissection
3. Méthodes des points fixes
   - 3.1. Convergence de la méthode
   - 3.2. Interprétation géométrique
   - 3.3. Théorème d"Ostrovski.
   - 3.4. Ordre de Convergence
   - 3.5. Extrapolation d"Aitken
4. Méthode de Newton
   - 4.1. Interprétation géométrique
   - 4.2. Analyse de convergence
   - 4.3. Théorème de Convergence Locale
5. Méthode de la sécante.
6. Applications

Chapitre III : Systèmes d'équations linéaires

1. Introduction générale
2. Résolution des Systèmes linéaires triangulaires

Partie I Méthodes Directes

3. Elimination de Gauss
4. Décomposition LU ou l'interprétation matricielle de la méthode de Gauss
   4.1. Décomposition de Crout
   4.2. Décomposition LU et permutation de lignes
5. Décomposition de Cholesky
6. Méthode de GAUSS avec permutation
   6.1. Pivot Partiel
   6.2. Pivot Total
   6.3. Pivot de la Toure (Rook Pivoting)
7. Systèmes tridiagonaux
8. Calcul de la Matrice inverse
9. Effets de l"arithmétique flottante
10. Conditionnement d"une matrice
11. Applications

Partie II Méthodes Itératives

12. Principe
13. Condition générale de Convergence
14. Applications :
    14.1. Méthode de Jacobi
    14.2. Méthode de Gauss-Seidel
    14.3. Evaluation de l"erreur
    14.4. Méthodes de relaxation et condition suffisante de convergence

Chapitre IV : Interpolation

1. Introduction
2. Matrice de Vandermonde
3. Interpolation de Lagrange
4. Polynôme de newton
5. Erreurs d'interpolation
6. Splines Cubiques :
   6.1. Courbes de la forme
   6.2. Splines paramétrées
7. Transformés de Fourier Discrètes.
8. Introduction au Nurbs :
   8.1. B-splines
   8.2. Génération de courbes

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques (Livres et polycopiés, sites internet, etc) :

1. Rappaz, J. et M. Picasso: Introduction a l'analyse numerique. Presses polytechniques et universitaires romandes, Lausanne, 1998.
2. Reddy, J.N.: An Introduction to the Finite Element Method. McGraw-Hill, New York, deuxieme edition, 1993.
3. Asher, U. M. et Petzold L. R.: Computer Methods for Ordinary Differential Equations and Differential-Agebraic Equations. SIAM, Philadelphia, USA,1998.
4. A. Fortin Analyse Numérique pour ingénieurs. Presses Internationales Polytechniques, 4ieme édition, 2011.
5. F.Filbet Analyse Numérique-Algorithme et Etude Mathématique. Sciences Sup. Dunod, 2013.
6. M., Benhamadou, A. Jeribi, Analyse Numérique Matricielle –Méthodes et Algorithmes, exercices et Problèmes corrigés . Edition Ellipses.2020

Semestre 5
Unité d'enseignement : UEF2
Matière : Programmation Linéaire et Dynamique
Coefficient : 3

Objectifs de l'enseignement

Ce module présente les principaux concepts de la programmation linéaire et dynamique (théoriques et algorithmiques) en mettant l'accent sur les problèmes applicatifs et les méthodes de résolution (graphique, simplexe,…). L'objectif est d'apprendre à l'étudiant à modéliser des problèmes combinatoires et de choisir l'algorithme le plus adéquat.

Connaissances préalables recommandées

Algèbre, Théorie des Graphes, Algorithmique 1, 2 & 3.

Contenu de la matière :

Chapitre I : La recherche opérationnelle.

1. Domaine de recherche opérationnelle.
2. Origine de la recherche opérationnelle.
3. Types de problèmes traités par la RO.
4. Brèves notions sur la complexité des algorithmes.
5. Domaines d'application de la recherche opérationnelle.

Chapitre II : Rappels.

1. Notions de base sur l'algèbre linéaire.
2. Matrices : notions et éléments de base.
3. Systèmes linéaires.
4. Graphes : notions de base.
5. Aperçu sur quelques problèmes traités en théories des graphes : chemin optimal, ordonnancement.

Chapitre III : Programme linéaire.

1. Introduction par un exemple.
2. Forme générale, forme canonique, forme standard.
3. Modélisation.
4. Résolution graphique.

Chapitre IV : Méthode du simplexe.

1. Algorithme général du Simplexe.
2. Algorithme à deux phases.

Chapitre V : La dualité en programmation linéaire.

1. PL dual.
2. Algorithme dual du simplexe.
3. Analyse post-optimale.

Chapitre VI: Programmation linéaire en nombres entiers.

1. Introduction par un exemple.
2. Résolution graphique.
3. Méthode branch and bound (B&B).

Chapitre VII : Programmation dynamique.

1. Introduction à la programmation dynamique.
2. Problème du sac à dos.

Chapitre VIII : Problème d'affectation

1. Couplage maximal.
2. Algorithme de Hongrois.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. J. Acher et J. Gardelle. *« Programmation linéaire »*. Livre. Editions Bordas. 1978.
2. Kaufmann et A. henry-Labordere. *« Méthodes et modèles de la recherche opérationnelle »*. Livre. Editions Dunod. 1974.
3. M. Sakarovitch. *« Optimisation Combinatoire – Tome 1 : Graphes et programmation Linéaire »*. Livre. Editions Hermann. 1984.
4. Roseaux. *« Exercices et problèmes résolus de recherche opérationnelle. Tome 1. Graphes : leurs usages, leurs algorithmes »*. Livre. Editions Dunod.1998.
5. Bourreau, E., Gondran, M., Lacomme, P., & Vinot, M. (2019). De la programmation linéaire à la programmation par contraintes (pp. 360-p). Ellipses.
6. Haddadi, S. (2021). Programmation linéaire: une approche mathématique et algorithmique. ed. Ellipses (2021) 1-192.

Semestre 5
Unité d'enseignement : UEF2
Matière : Fondements de l'IA
Coefficient : 3

Objectifs de l'enseignement

L'objectif de ce module est de présenter les fondements du domaine de l'intelligence artificielle. Dans ce cours, les principes de base des différents éléments de l'intelligence artificielle seront présentés : Logiques et Raisonnement, Résolution de problèmes, Méta heuristiques, Problème de satisfaction de contraintes, Apprentissage automatique et réseaux de neurones, Systèmes multi-agents.

Connaissances préalables recommandées

Algèbre, Théorie des Graphes, Algorithmique 1, 2 & 3.

Contenu de la matière :

Chapitre I: Introduction à l'intelligence artificielle (IA)

1. Historique et Définitions
2. Importance de l'IA
3. Approches de l'IA
4. Domaines d'applications de l'IA

Chapitre II : Logiques et Raisonnement

1. Logique propositionnelle
2. Logique du 1er ordre
3. Introduction aux logiques non classiques

Chapitre III : Algorithmes de Recherche pour la Résolution de Problèmes

1. Représentation d'un problème par un espace d'états
2. Méthodes de recherche de solution dans les espaces d'états
3. Algorithmes de recherches

Chapitre IV. Théorie de Jeux

1. Formalisations de jeux
2. Stratégies d'optimisation
3. Algorithmes MinMax
4. Algorithme Alpha Beta

Chapitre V. Les méta-heuristiques

1. La recherche locale
2. Méthode Simulated Annealing (Recuit Simulé)
3. Algorithme génétique

Chapitre VI. Problème de satisfaction de contraintes (CSP)

1. Exemples de CSP
2. Recherche en arrière pour les CSPs (BackTracking)
3. Algorithme AC-3

Chapitre VII. Apprentissage automatique et réseaux de neurones

1. Algorithmes d'apprentissage automatique
2. Méthode des K plus proches voisins
3. Les réseaux de neurones

Chapitre VIII : Les systèmes multi-agents

1. Concepts de base
2. Les systèmes multi agents. (SMA) : Interaction, Communication, Négociation et Architecture.
3. Approches d'apprentissage dans les systèmes multi agents
4. Application : planification et ordonnancement

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Principles of Artificial Intelligence par Nils J. Nilson, 1982, Springer-Verlag Berlin Heidelberg.
2. Essentials of Artificial Intelligence par Nils J. Nilson, 1993, Copyright: © Morgan Kaufmann
3. Artificial Intelligence: A new synthesis, Nils Nilson, 1997, Elsevier.
4. Artificial Intelligence: A Modern Approach par Stuart Russell and Peter Norvig, Ed. Pearson, 2016
5. Rapport de synthèse - France Intelligence Artificielle. Ministère de l'Enseignement supérieur et de la Recherche, 2017.
6. Castaño, A. P. (2018). Practical Artificial Intelligence: Machine Learning, Bots, and Agent Solutions Using C. Apress.
7. Barraud, B. (2019). L'intelligence artificielle–Dans toutes ses dimensions. Université Paris 1 Panthéon-Sorbonne-Université Jiao Tong de Shanghai-L'Harmattan.
8. BAUM, H. (2023). Introduction to artificial intelligence. AG Printing & Publishing.

Semestre 5
Unité d'enseignement : UED1
Matière : Génie logiciel
Coefficient : 2

Objectifs de l'enseignement

- Acquérir les concepts de base du génie logiciel en termes d'objectifs de qualité, de coût et de délai.
- Maitriser les étapes d'un cycle de vie de logiciels
- Comprendre le processus de spécification et d'analyse des besoins en utilisant les diagrammes de cas d'utilisation UML
- Maîtriser les aspects de conception de logiciels
- Maîtriser la conception d'un diagramme de classes et de séquences UML.
- Acquérir les bases de la conception de qualité
- Exploiter les patrons de conception
- Maîtrise les techniques de test dans le but de détecter la présence d'erreurs dans un programme vis-à-vis de sa spécification.

Connaissances préalables recommandées

Maîtrise des concepts algorithmiques et de la programmation ✓ Notions de base en Programmation Orientée Objet.

Contenu de la matière :

Chapitre I : Introduction

1. Définitions et Objectifs du génie logiciel
2. Qualités d´un logiciel (critères et facteurs de qualité)
3. Notion de cycle de vie d´un logiciel
4. Les étapes d´un cycle de vie de logiciels
5. Les différents modèles de cycle de vie (cascade, V, prototypage, spirale, incréments)

Chapitre II : Introduction á la modélisation avec UML

1. Concepts de base de la modélisation Orienté- Objet
2. Mécanismes généraux de la modélisation Orienté- Objet
3. Les axes de modélisation d´UML
4. Les diagrammes d´UML

Chapitre III : Modélisation orientée objet à l'aide d'UML / Aspect fonctionnel

1. Définition des besoins à laide du diagramme de cas d´utilisation
2. Description textuelles détaillées des cas d´utilisation

Chapitre IV : Modélisation orientée objet à l'aide d'UML / Aspect Statique

1. Diagrammes de classes
2. Diagramme d´objets

Chapitre V : Modélisation orientée objet à l'aide d'UML / Aspect Dynamique

1. Les diagrammes d‟interactions : de séquences et communication
2. Les diagrammes d‟états-transitions et d‟activités.

Chapitre VI : Les patrons de conception

Introduction aux patrons de conception

Chapitre VII : Concepts avancés

Conception Orientée Composants : Diagrammes de composants …

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Olivier Englender, Sophie Fernandes. Manager un projet informatique, Troisième édition, EYROLLES, 2012
2. Valéry Guilhem Frémaux. Le projet informatique de A à Z, Approche pragmatique de la gestion de projet, ellipses, 2006
3. Stanley Portny & Sandrine Sage. La gestion de projets, , © Éditions First-Gründ, Paris, 2011 pour l'édition française. Publié en accord avec Wiley Publishing, Inc.
4. Roger Aïm. Les fondamentaux de la gestion de projet, Afnor édition 2011,
5. Christine Peterson. Le Guide Pratique à la Gestion de Projet, PMP &bookboon.com, 2014, ISBN 978-87-403-0615-6

Semestre : 5
Unité d'enseignement : UET1
Matière : Développement mobile (Module optionnel)
Coefficient : 2

Objectifs de l'enseignement

Ce cours fournit une base de référence pour créer des applications mobiles à l'aide de Flutter et Dart. Les participants apprendront les bases du langage de programmation Dart et comment développer des applications mobiles multiplateformes avec Flutter.

Connaissances préalables recommandées

- Compréhension de base des concepts de programmation
- La connaissance des langages de programmation orientés objet est bénéfique mais pas obligatoire

Contenu de la matière :

Chapitre I : Introduction à Flutter

1. Présentation du framework Flutter
2. Avantages de Flutter pour le développement d'applications mobiles
3. Comprendre l'architecture de Flutter
4. Mise en place de l'environnement de développement Flutter

Chapitre II : Commencer avec Dart

1. Introduction au langage de programmation Dart
2. Variables, types de données et opérateurs
3. Flux de contrôle : instructions if-else, boucles
4. Fonctions et paramètres

Chapitre III : Créer des interfaces utilisateur avec Flutter

1. Widgets et mise en page Flutter
2. Gestion des entrées et des gestes de l'utilisateur
3. Styliser et thématiser les applications Flutter
4. Navigation et routage dans les applications Flutter

Chapitre IV : Gestion de l'état dans Flutter

1. Comprendre les widgets avec et sans état
2. Gestion de l'état avec setState()
3. Utilisation d'un fournisseur pour la gestion de l'état
4. Introduction au modèle BLoC

Chapitre V : Travailler avec des données dans Flutter

1. Récupération de données à partir des API REST
2. Analyse des données JSON dans Flutter
3. Stockage des données localement à l'aide de SQLite ou SharedPreferences
4. Travail avec les formulaires et validation

Projet final

• Les participants travaillent sur un projet final pour appliquer les concepts appris tout au long du cours. Le projet consistera à concevoir et développer une application mobile multiplateforme fonctionnelle utilisant Flutter et Dart.

Références

TEXT BOOKS:

1. T1. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011)

REFERENCE BOOKS:

1. R1. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd
2. R2. Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
3. R3. Android Application Development All in one for Dummies by Barry Burd, Edition: I

Semestre : 5
Unité d'enseignement : UET1
Matière : Interfaces Homme-Machine (Module optionnel)
Coefficient : 2

Objectifs de l'enseignement :

- Introduire les concepts de base de l'interaction Homme-Machine et donner une vision complète de tous les aspects liés.
- Maîtriser la conception/la modélisation, le développement et l'évaluation des IHMs
- Acquérir les compétences nécessaires pour le développement des IHMs

Connaissances préalables recommandées

Pour aborder efficacement le module Interfaces Homme-Machine (IHM), il est recommandé de posséder des connaissances de base en informatique et en programmation, en particulier dans des langages tels que Python, JavaScript, ou HTML/CSS. Enfin, une expérience préalable avec des systèmes interactifs ou des projets de développement web ou mobile vous aidera à appréhender plus facilement les concepts avancés du module.

Contenu de la matière :

Chapitre I : Introduction à l'IHM (4h)

1. Historique et définitions
   - 1.1. Pourquoi IHM ?
   - 1.2. Interface, Interaction, Communication et Dialogue
   - 1.3. Facteurs humains : ergonomie et psychologie cognitive
   - 1.4. Design : design industriel, arts graphiques, typographie
2. Évolution des IHMs
   - 2.1. Navigation, indexation et annotation
   - 2.2. Interfaces graphiques
   - 2.3. What you see is what you get
   - 2.4. Interface multimodale
   - 2.5. Informatique ubiquitaire

Chapitre II : Ergonomie et sciences cognitives pour l'IHM (4h)

1. Notions de base sur les sciences cognitives
   - 1.1. Le sous-système sensoriel
   - 1.2. Le sous-système cognitif
   - 1.3. Le sous-système moteur
   - 1.4. Les limites
   - 1.5. L'approche cognitive pour les IHMs
   - 1.6. L'approche pour les utilisateurs
2. Ergonomie et guide de style
   - 2.1. Définition
   - 2.2. Critères ergonomiques
     a. Compatibilité
     b. Guidage
     c. Homogénéité
     d. Souplesse
     e. Contrôle explicite
     f. Gestion des erreurs
     g. Concision

Chapitre III : Modélisation des IHMs (3h)

1. Les besoins de modélisation en IHM

2. Les modèles en IHM
   2.1. Modèles d'architecture
   2.2. Modèles linguistiques
   2.3. Modèles à agent
   2.4. Modèles d'interaction

Chapitre IV : Développement et évaluation des IHMs (4h)

1. Programmation événementielle
   1.1. Définition
   1.2. Bibliothèque et environnements (JavaFX, QT, etc.)
2. Boîtes à Outils
   2.1. Interface web (Ink, TCL/tk, etc.)
   2.2. Interface mobile (Android, iOS, etc.)
   2.3. Limitations
3. Évaluation des IHMs
   3.1. Objectifs de l'évaluation
   3.2. Les outils de l'évaluation
   3.3. Limitations

CONTROLE DE CONNAISSANCES

- Examen écrit sur le cours
- Contrôle continu des TD/TP

BIBLIOGRAPHIE

1. Ludovie Cinquin, Erika Duriot, EricGroise, Olivier Mallassi, André Nedelcoux, David Rousselie, Vanessa
2. Vimond « les dossiers de l'écran ; Utilisabilité et technologie IHM », Edition OCTO, Technologiue 2010
3. J. Coutaz, « Interface Homme-Ordinateur » Dunod, 19990.
4. N. Elouali, « Approche à base des modèles pour construction d'applications mobiles multimodales », PHD (2014).

Semestre : 5
Unité d'enseignement : UET1
Matière : Conception de jeux vidéo (Module optionnel)
Coefficient : 2

Objectifs de l'enseignement

L'objectif de ce cours et de fournir à l'étudiant les principes de base de design et de conception d'un jeu vidéo allant du 2D au 3D. Plusieurs plateformes seront présentées et étudiés en pratique pour offrir un choix de développement.

Connaissances préalables recommandées

Pour le module Conception de jeux vidéo, des connaissances en programmation (C++, C#, Python), mathématiques (géométrie, algèbre linéaire), et utilisation de moteurs de jeu (Unity, Unreal Engine) sont recommandées. Des compétences en design graphique et en game design enrichiront également votre expérience.

Contenu de la matière :

Chapitre I : Conception et développement de jeux : Principes de base

1. Concepteur de jeux : Définition et tâches
2. Notion de jouabilité de jeu
3. Besoins du concepteur de jeux
4. Exigences des joueurs
5. Outils de conception
6. Document de conception (GDD)

Chapitre II : Gestion des Lutins (Sprites) et tuiles (Tiles)

1. Génération et visualisation de sprites
2. Génération et visualisation de de tuiles
3. Atlas de tuiles (spritesheet)
4. Défilement de scènes de tuiles (Scrolling tilemaps)
5. Tuiles isométriques
6. Exemple d'un jeux 2D

Chapitre III : Détection de collision

1. Physique des objets rigides
2. Le moteur physique
3. Calcul de la collision
4. Moteur physique box2dweb

Chapitre IV : Webgl et son intégration dans les jeux 3D

1. Le WEBGL
2. Three.js
3. Editeur Three.js
4. Intégration de WebGL dans les jeux 3D
5. Exemple de jeux 3D basé sur Three.js

Chapitre V : Applications : Le moteur de jeux Unity

6. Principes de base de Unity
7. Navigation dans les scènes et physique
8. Importation de ressources statiques
9. Création de scripts avec C#
10. Animation
11. Armement et effets spéciaux.

Mode d'évaluation : (type d'évaluation et pondération)

Contrôle continu (40%), Examen pratique (60%)

Références

1. Game design : theory and practice; Richard Rousse, 2005
2. Beginning game level design, John Feil & Marc scattergood, 2005
3. Game Programming, All in One, Jonathan s. Harbour, 2004
4. Android game programming for dummies, Derek James, John Wiley & Son Inc. 2013
5. Unity for Absolute Beginners, Sue Blackman, 2014, Apress, www.it-ebooks.info

Semestre 6
Unité d'enseignement : UEF1
Matière : Réseaux Avancés
Coefficient : 3

Objectifs de l'enseignement

L'objectif de ce cours est d'initier l'étudiant aux réseaux longues distances et les technologies associées. L'étudiant apprendra également le routage dynamique, la notion de qualité de service dans les réseaux et les réseaux mobiles. Le cours attache un intérêt particulier à la couche transport ; réseau et à certains protocoles de la couche application, le DNS notamment.

Connaissances préalables recommandées

Réseau 1, Notions de base de l'électricité et l'électronique, Systèmes d'exploitation de base

Contenu de la matière :

Chapitre I : Protocoles de transport

1. Rôle et position dans le modèle OSI - TCP/IP
2. Notion de contrôle de flux et de récupération sur erreur
3. Notion de port
4. Protocole TCP (mode connecté)
5. Protocole UDP (mode non connecté)
6. Interface de programmation réseaux : Les sockets
7. TP (Utilisation de Telnet, FTP, WireShark)

Chapitre II : Adressage et routage dynamique

1. Rappels sur l'adressage IPV4
2. La communication multicast dans les réseaux IP
3. Le routage dynamique et le routage sur Internet (RIP, OSPF, BGP)
4. Étude avancée de l'adressage IPV6: mécanismes d'auto configuration, gestion de la mobilité
5. TP (Configuration de routage dynamique (RIP, OSPF et BGP))

Chapitre III : La qualité de service (QoS) dans les réseaux IP

1. Définitions et problématique
2. Mécanismes pour gérer la Qualité de Service (QoS)
3. Architectures de la QoS: best effort, IntServ, DiffServ; Service à charge contrôlée
4. Le protocole de signalisation RSVP
5. Contrôle de congestion et contrôle de flux
6. IPv6 et la QoS
7. TP (Mise en œuvre d'un mécanisme de QoS sur les routeurs)

Chapitre IV : Les réseaux étendus (haut débit)

1. Réseaux à haut débit : architecture, techniques, commutation et routage
2. Les technologies grande distance (PDH, SDH)
3. Réseaux optiques (SONET/SDH): les techniques de multiplexage WDM, C-WDM, DWDM
4. Les accès opérateurs : Types d'interface, Niveau de disponibilité, Les contraintes, Frame relay, ATM
5. Technologie MPLS et GMPLS: techniques de commutation et de signalisation

Chapitre V : Introduction aux réseaux mobiles

1. Réseaux mobiles radio de télécommunication : GSM, GPRS, UMTS
2. Normes (3G et dérivées): architecture et protocoles
3. Déploiement et administration des technologies de téléphonie mobile

Mode d'évaluation : (type d'évaluation et pondération)

Contrôle continu (40%), Examen pratique (60%)

Références :

1. Mühlethaler, P., "802.11 et les réseaux sans fil", Eyrolles, 2002.
2. Cisco system (Paris), Christian Soubrier, "Architecture de réseaux et études de cas", Campus Press, 1999.
3. Tanenbaum, A., "Réseaux: Architectures, protocoles, applications". Ed.: InterEditions, 3ème édition, 1997.
4. Kurose, J.F. and Ross, K.W., "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson, 3ème édition, 2004.
5. Ferguson, P., Huston, G., "Quality of Service: Delivering QoS on the Internet and in Corporate Network", Wiley, 1st edition, 1998.
6. Jain, R., "The art of computer systems performance analysis", John Wiley & Sons, 2008.

Semestre 6
Unité d'enseignement : UEF1
Matière : Système d'Exploitation : Synchronisation et Communication
Coefficient : 4

Objectifs de l'enseignement

L'objectif de ce cours sont multiples :

- Introduire la problématique du parallélisme dans les systèmes d'exploitation, étudier la mise en œuvre de la coopération à travers les outils de synchronisation et de communication et les problèmes associés.
- *Recommandation* : Illustrer les différents concepts à l'aide d'outils (création de processus, signaux, sémaphores, mémoire partagée, files de messages, tubes) du système UNIX notamment dans les séances de TP.

Connaissances préalables recommandées

Des connaissances en programmation (C, C++), structures de données, et concepts de base des systèmes d'exploitation (processus, threads, mémoire) sont recommandées. Une compréhension des mécanismes de synchronisation et des protocoles de communication inter-processus est également essentielle.

Contenu de la matière :

Chapitre I : NOTION DE PARALLELISME/CONCURRENCE

1. Systèmes de tâches
2. Parallélisme et langages évolués
3. Déterminisme d'un système de tâches
4. Parallélisme maximal

Chapitre II : SYNCHRONISATION DES PROCESSUS

1. Partage de ressources et des données
2. Exclusion mutuelle
3. Mécanismes d'exclusion mutuelle : variables d'états, solutions matérielles, les sémaphores, autres mécanismes de synchronisation

Chapitre III : COMMUNICATION ENTRE PROCESSUS

1. Communication par partage de variables : Modèle du producteur/ consommateur, modèle des lecteurs/ rédacteurs.
2. Communication par échanges de messages : Désignation (directe : CSP, indirecte : les boites aux lettres)

Chapitre IV : MONITEURS ET REGIONS CRITIQUES

1. Les moniteurs : Structure, implémentation, illustrations, variantes de moniteurs
2. Régions critiques (Simples, Conditionnelles)

Chapitre V : INTERBLOCAGE

1. Caractéristiques, Modèles, Formalisation, Représentations graphiques
2. Traitement de l'interblocage : Prévention, Détection- Guérison, Evitement

Chapitre VI : INTRODUCTION AUX SYSTEMES REPARTIS

1. Définition
   1.1. Système reparti
   1.2. Systèmes de communication
2. Caractéristiques des Systèmes répartis
3. Etape de conception des systèmes d'exploitation répartis
4. Algorithmes répartis
   4.1. Définition

4.2. Qualité d'un algorithme réparti
4.3. Quelques algorithmes de contrôle réparti
   a. Exclusion mutuelle
   b. Election
   c. Diffusion
   d. Terminaison
5. Exemples de systèmes répartis

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Thomas Anderson and Michael Dahlin Operating Systems: Principles and Practice (Second Edition) Volume II: Concurrency ISBN : 978-0-9856735-4-3 Publisher: Recursive Books, Ltd., 2011-2015.
2. M. J. Bach, traduit par G. Fellah, „"Conception du Système UNIX,"" Masson et Prentice Hall,1990.
3. J. Beauquier, B. Berard,"" Systèmes d'exploitation : concepts et algorithmes,"" McGraw-Hill, 1990.
4. Crocus,"" Systèmes d"exploitation des ordinateurs"", Dunod 1975.
5. Patrick Cegielski, Conception de systèmes d'exploitation : le cas linux, Éditions Eyrolles, 16 Octobre 2003.
6. N. B. Fontaine, P. Hammes, „"UNIX système V: Système et environnement"" Masson 1989.
7. S. Krakowiak, „"Principes des systèmes d'exploitation des ordinateurs,"" Dunod 1987.
8. J-L.Peterson, A.Silbershartz „"Operating Systems Concepts,""Addison-Wesly Publishing
9. Company,Inc, 1983.
10. A. Silberschatz, P. B. Galvin „" Principes des systèmes d"exploitation,"" 4 e Edition, Addison Wesley, 1994.
11. William Stallings, Operating Systems: Internals and Design Principles (8th Edition),
12. Publisher : Pearson; 8th edition (February 2, 2014).
13. A. Tanenbaum, „" Modern Operating Systems, "" Third Edition Prentice Hall, 2009.
14. A. Tananbaun, „"Systèmes d"exploitation,"" 3°édition, Pearson Edition. 2008.
15. J-P. Verjus et al,""Synchronisation des programmes parallèles : Expression et mise en oeuvre dans les systèmes centralisés, édition 1983 Dunod.
16. Andrew S. Tanenbaum , Maarten Van Steen (2006): Distributed Systems: Principles and Paradigms.

Semestre 6
Unité d'enseignement : UEF1
Matière : Gestion de projets
Coefficient : 4

Objectifs de l'enseignement

- Permettre à l'étudiant de comprendre l'enjeu majeur de la gestion de projet.
- Initier l'étudiant au processus d'organisation et de planification.
- Entrainer l'étudiant à l'application de processus, méthodes et outils de planification.
- Initier l'étudiant aux environnements de gestion projet.

Connaissances préalables recommandées

Pour le module Gestion de projets, des connaissances en méthodes de gestion (Agile, Waterfall), outils de planification (Gantt, PERT), et logiciels de gestion de projet (Trello, MS Project) sont recommandées. Des compétences en communication et en leadership sont également essentielles pour coordonner efficacement les équipes et les ressources.

Contenu de la matière :

Chapitre I : Qu'est-ce qu'un projet ?

1. Les caractéristiques d'un projet : les fondamentaux
2. Les différences entre opérations et projets
3. Maîtrise d'ouvrage (MOA) et Maîtrise d'œuvre (MOE)
4. Le rôle et les compétences du chef de projet
5. Les aspects humains du travail en mode projet
6. Le cycle de vie d'un projet (introduction)
   6.1. La phase d´avant-projet
   6.2. La phase d'initialisation
   6.3. La phase de réalisation et de contrôle
   6.4. La phase de clôture

Chapitre II : La phase d'avant-projet

1. L'analyse du besoin
2. L'étude d'opportunité
3. L'étude de faisabilité
4. Le Cahier des Charges
5. L'analyse des risques

Chapitre III : La phase d'initialisation du projet

1. La définition de l'équipe projet
2. La Work Breakdown Structure (WBS)
3. La planification du projet
   3.1. diagramme de Gantt,
   3.2. les liaisons entre tâches dans le diagramme de Gantt
   3.3. Réseau PERT
4. L'identification des ressources nécessaires
5. Le calcul du coût du projet
6. La planification des mécanismes de contrôle de la qualité

Chapitre IV : Le lancement du projet

1. La réunion de lancement
2. Construire un ordre du jour d'une réunion de lancement de projet
3. Animer une réunion de lancement de projet

4. Faire le compte rendu d'une réunion de lancement

Chapitre V : La phase de réalisation et de contrôle

1. Le suivi du projet
2. La gestion du triangle Qualité-Coûts-Délais
3. Le management d'équipe projet
4. Le reporting et les ajustements en cours de projet

Chapitre VI : La phase de clôture

1. La recette d'un projet
2. Le bilan d'un projet et la réunion de clôture
3. La capitalisation en fin de projet
4. Le redéploiement des ressources humaines en fin de projet

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Principles of software engineering management by Tom GILB Edition Lavoisier.
2. Software Engineering: A Practitioner's Approach by Roger S Pressman.
3. Software Project Management in Practice by Pankaj Jalote.
4. Génie logiciel: principes, méthodes et techniques by Alfred Strohmeier et Didier Buchs.

Semestre : 6
Unité d'enseignement : UEF2
Matière : Programmation WEB
Coefficient : 3

Objectifs de l'enseignement

- Apprendre à créer des sites et applications Web modernes et réactifs.
- Maîtriser les standards du Web et les bonnes pratiques de développement.
- Maîtriser la programmation Web côté client et côté serveur avec les Framework actuels.

Connaissances préalables recommandées

- Algorithmique et programmation en C, Bases de données et langage SQL.

Contenu de la matière :

Chapitre I : Introduction au Web

1. Historique et évolution du Web.
2. Notions de bases du Web (HTTP, Client, Serveur, etc.).
3. Rappel sur les principaux protocoles du Web.

Chapitre II : Langage HTML 5

1. Historique et différentes versions.
2. Syntaxe du Langage : document bien formé, document conforme, outils de validation.
3. Présentation des éléments de base du Langage :
   3.1. Sauts de ligne et séparateurs, titres, paragraphes, listes, tableaux, etc.
   3.2. Sections sémantiques (section, article, nav, etc.)
   3.3. Les formulaires.
4. Les éléments multimédias (audio, vidéo, etc.) et les graphiques : Canvas et SVG.

Chapitre III : Les feuilles de style CSS

1. Séparation entre contenu et mise en forme et avantages.
2. Syntaxe de base de CSS : Sélecteurs, Propriétés.
3. Media Queries et responsive design.
4. Gestion des conflits.
5. Introduction aux préprocesseurs CSS (Sass, LESS).

Chapitre IV : JavaScript et Frameworks modernes

1. Principe de la programmation web côté client.
2. Syntaxe de JavaScript.
3. Introduction aux frameworks et librairies JavaScript.
   3.1. React.js : Concepts de base, composants, état et props, cycle de vie.
   3.2. Vue.js : Concepts de base, directives, composants, réactivité.
   3.3. Comparaison des frameworks (React, Vue, Angular).

Chapitre V : Programmation côté serveur avec PHP et Node.js

1. Aperçu sur les langages de programmation côté serveur.
2. Syntaxe du langage PHP.
3. Fonctions PHP prédéfinies.
4. Traitement des formulaires en PHP.
5. Interaction avec les bases de données (MySQL).
6. Sessions et cookies.
7. Gestion des fichiers.

8. Introduction à Node.js : Installation, modules, npm, création de serveur basique.

Chapitre VI : AJAX Présentation.

1. Mode synchrone vs asynchrone.
2. L'objet XMLHTTPRequest.
3. Structure d'un code AJAX.
4. Interaction avec PHP et Node.js.
5. Utilisation de Fetch API pour les requêtes asynchrones.

Chapitre VII : Utilisation des Frameworks

1. Limites de la programmation traditionnelle.
2. L'utilité des Frameworks.
3. Exemples de Frameworks CSS (ex. Bootstrap, Tailwind CSS).
4. Le Framework Laravel.

Chapitre VIII : Web Intelligence

1. Concepts de base de l'intelligence web.
2. Utilisation des données pour améliorer les expériences utilisateur.
3. Outils et techniques de collecte de données sur le web.
4. Applications de l'intelligence web dans le développement moderne.

Chapitre X : Bonnes pratiques de développement Web

1. Accessibilité web.
2. SEO (Search Engine Optimization).
3. Performance et optimisation.
4. Tests et débogage.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. World Wide Web Consortium, standards for web Design and applications
2. Site Web partenaire du consortium W3C destiné aux développeurs Web
3. Site Web officiel : Manuel du PHP
4. Rodolphe Rimelé, "HTML 5 Une référence pour le développeur web", éditions Eyrolles, 2017.
5. J. Engels, "PHP 7 Cours et exercices.", éditions Eyrolles, 2017.
6. Mat Marquis, "JavaScript pour les web designers", éditions Eyrolles, 2017.
7. Michel Plasse, "Développez en Ajax", éditions Eyrolles, 2006.
8. React Documentation
9. Vue.js Documentation
10. Node.js Documentation

Semestre 6
Unité d'enseignement : UEF2
Matière : Analyse numérique2
Coefficient : 2

Objectifs de l'enseignement

Le module d'analyse numérique est un module de base dans le domaine du calcul scientifique. Le calcul scientifique étant défini comme la branche qui permet d'implémenter une solution d'un problème sur un calculateur. L'objectif du module est l'acquisition des méthodes de discrétisation, des algorithmes de résolution des problèmes spécifiques posés dans le domaine de l'ingénierie informatique, l'analyse des algorithmes (rapidité, précision et souplesse), et enfin l'estimation des erreurs commises dans les modèles mathématiques employées ainsi que dans leur implémentation.

Connaissances préalables recommandées

Les étudiants doivent maîtriser l'algèbre linéaire, le calcul différentiel et intégral, et les équations différentielles. Ils doivent savoir programmer en Python ou MATLAB, comprendre les structures de données et la complexité algorithmique, et avoir des bases en probabilités et statistiques. Des notions d'analyse de Fourier et d'analyse numérique de base sont également recommandées.

Contenu de la matière :

Chapitre I : Différentiation et Intégration Numériques

1. Introduction
2. Différentiation Numérique
   2.1. Différentiation d‟ordre 1
   2.2. Différentiation d‟ordre supérieur
3. Intégration Numérique :
   3.1. Formules de Newton -côtes simples et Composées
   3.2. Méthode de Romberg
   3.3. Quadratures de Gauss-Legendre
   3.4. Intégration en utilisant les Splines
   3.5. Applications

Chapitre II : Calcul numérique des valeurs et vecteurs propres

1. Introduction
2. Localisation des valeurs propres : Théorème de Gershgorin
3. Méthode de la puissance
4. Méthode de la puissance Inverse
5. Méthode QRApplications

Chapitre III : Equations différentielles

1. Introduction
2. Méthode d‟Euler Explicite
3. Méthode de Taylor
4. Méthode de Runge-Kutta
   4.1. Méthode de Runge-Kutta d‟ordre2
   4.2. Méthode de Runge-Kutta d‟ordre 4
   4.3. Analyse de l‟erreur
5. Méthodes à pas multiples

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Rappaz, J. et M. Picasso: *Introduction a l'analyse numerique*. Presses polytechniques et universitaires romandes, Lausanne, 1998.
2. Reddy, J.N.: *An Introduction to the Finite Element Method*. McGraw-Hill, New York, deuxieme edition, 1993.
3. Asher, U. M. et Petzold L. R.: Computer Methods for Ordinary Differential Equations and Differential-Agebraic Equations. SIAM, Philadelphia, USA,1998.
4. A. Fortin Analyse Numérique pour ingénieurs. Presses Internationales Polytechniques, 4ieme édition, 2011.
5. F.Filbet Analyse Numérique-Algorithme et Etude Mathématique. Sciences Sup. Dunod, 2013.
6. M., Benhamadou, A. Jeribi, Analyse Numérique Matricielle –Méthodes et Algorithmes, exercices et Problèmes corrigés . Edition Ellipses.2020

Semestre : 6
Unité d'enseignement : UED1
Matière : Introduction à la sécurité Informatique
Coefficient : 2

Objectifs de l'enseignement

Ce module permet d'initier les étudiants aux concepts de base de la sécurité informatique. En particulier, ce module consiste à :

- Présenter les aspects fondamentaux de la sécurité informatique.
- Identifier les principaux enjeux de la sécurité informatique.
- Distinguer et d'expliquer les différents types d'attaques auxquels un réseau ou un système informatique peut faire face.
- Etudier les principes et les techniques de la cryptographie.
- Sécuriser un site web par SSL.
- Sécuriser la messagerie par PGP.
- Protéger un hôte ou un réseau par un firewall et IDS.

Connaissances préalables recommandées

Les étudiants doivent maîtriser les concepts de base des systèmes d'exploitation, des réseaux informatiques, et de la programmation. Ils doivent comprendre les principes des protocoles réseau, de la cryptographie, et des modèles de sécurité. Une connaissance des langages de programmation comme Python ou C++ est nécessaire. Des notions en gestion des systèmes et administration réseau sont également recommandées.

Contenu de la matière :

Chapitre I : Introduction

1. Définition
2. Buts de la sécurité
   - 2.1. Confidentialité
   - 2.2. Intégrité
   - 2.3. Disponibilité
   - 2.4. Non répudiation
3. Domaines d'application de la sécurité physique
   - 3.1. Sécurité d'exploitation
   - 3.2. Sécurité logique
   - 3.3. Sécurité applicative
   - 3.4. Sécurité des télécommunications
4. Attaques
   - 4.1. Types d"Attaques
     a. Script kiddie
     b. Hacktivist
     c. Insider
     d. Organized crime
     e. Competitors
     f. Open-source intelligence
     g. Advanced persistent threat (APT)
   - 4.2. Les types de malware
     a. Viruses
     b. Worms

c. Logic Bombs
d. Backdoor
e. Trojans
f. Remote access Trojan (RAT)
g. Ransomware
h. Keylogger
i. Spyware
j. Bots and Botnets
k. Rootkit

4.3. Reconnaître les attaques courantes

a. Social Engineering
b. Shoulder Surfing
c. Tricking Users with Hoaxes
d. Tailgating and Mantraps
e. Dumpster Diving
f. Watering Hole Attacks
g. Attacks via Email and Phone (spam, phishing, spear phishing, and whaling)

5. Politique de sécurité

Chapitre II : Cryptographie classique (chiffrement)

1. Introduction
2. Chiffre de César
3. Chiffrement monoalphabetique
4. Chiffrement par permutation par bloc
5. Chiffrement de Vigénaire
6. Chiffrement de HILL
7. Chiffre affine

Chapitre III : cryptographie moderne

1. Introduction
2. Classes de chiffrement moderne
   2.1. Chiffrement symétrique
   2.2. Chiffrement asymétrique
3. Chiffrement symétrique (AES)
4. Chiffrement asymétrique (à clé publique)RSA: (Rivest - Shamir – Adleman)
   4.1. Exponentiation modulaire
   4.2. Exponentiation modulaire rapide
5. Notion de clé de session
   5.1. Protocole d'échange de clé
6. Enveloppe digitale
7. Condensé de message (haché)
8. Signature numérique

Chapitre IV : Gestion des clés

1. Introduction
2. Système basé sur CA (Certification Authority)
3. Principales fonctions
4. Architecture de l‟infrastructure de gestion des clés
5. Certificat numérique
6. Installation d‟autorité de certification sur windows serveur

Chapitre V : Protocoles sécurisés

1. Protocole SSL (Secure Socket Layer)
   1.1. Fonctions de SSL

1.2. Fonctionnement de SSL
1.3. Surveillez les alertes SSL
1.4. Configurer un Service HTTPS dans IIS
2. SSH (Secure Shell)
3. IPSec (IP Secure)
3.1. Architecture
3.2. SA : Security Association
3.3. SPD : Security Policy Database

Chapitre VI : Protocole PGP

1. Introduction
2. PGP: (Pretty Good Privacy)
3. PGP: Algorithme
4. PGP: Authentification
5. PGP: Confidentialité
6. PGP: Compression
7. PGP: chiffrer ses mails sur windows

Chapitre VII : Pare-feux et IDS

1. Introduction
2. Les Firewall
2.1. Caractéristiques des Firewall
2.2. Les types de Firewall
a. Le filtrage statique
b. Le filtrage dynamique (filtrage de paquet avec état)
c. Le filtrage applicatif
d. Le filtrage statique
- Les règles de filtrage des ACL
- Filtrage statique par iptables
- Gestion des règles par iptables

e. Le filtrage dynamique (filtrage de paquet avec état)
f. Le filtrage applicatif
2.3. Les limites d‟un firewall
3. Système de détection d‟intrusions (IDS)
3.1. Fonctions d‟un IDS
3.2. Classification des IDS
3.3. Exemple d‟IDS

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

Citer au moins 3 à 4 références classiques et importantes.

1. Practical Malware Analysis - Andrew Honig and Michael Sikorski. Published: February 29, 2012 by No Starch Press
2. The Hacker Playbook 3: Practical Guide To Penetration Testing - Peter Kim. Published: May 1, 2018 by Secure Planet
3. Sécurité informatique Ethical Hacking -Apprendre l'attaque pour mieux se défendre », Editions ENI - Octobre 2009 ISBN: 978-2-7460-5105-8
4. ETHICAL HACKING AND PENETRATION TESTING GUIDE, CRC Press Taylor & Francis Group ISBN-2015 : 13: 978-1-4822-3162-5
5. Ethical Hacking & Countermeasures- Threats & Defense Mechanisms, EC-Council | Press 2010 ISBN- 13 978-1-4354-8361-3
6. Mastering Hacking (The Art of Information Gathering & Scanning) - Harsh Bothra. Published: August 28, 2019
7. Stéphane Natkin, les protocoles de sécurité de l‟‟internet, Dunod 2002
8. Bruce Schneier, cryptographie appliquée, Thomson Publishing, 1995
9. Bruno Martin, Codage, cryptologie et applications, Presses polytechniques et universitaires romandes, 2004
10. Stallings Williams, sécurité des réseaux, applications et standards, Vuibert Informatique, 2002
11. Solange Ghernaouti, Sécurité informatique et réseaux, 4ème édition Dunod, 2013
12. Gildas Avoine, Pascal Junod, Philippe Deschslin, sécurité informatique: cours et exercices corrigés, 2ème édition Vuibert 2009

Semestre 6
Unité d'enseignement : UET1
Matière : Éthique de l'IA
Coefficient : 2

Objectifs de l'enseignement

L'étudiant doit être préparé au monde réel du travail qualité d'ingénieur, et en qualité de décideur tant sur le plan organisationnel que sur le plan éthique.

Connaissances préalables recommandées

Les étudiants doivent avoir des bases en intelligence artificielle, en informatique, et en philosophie morale. Ils doivent comprendre les concepts fondamentaux de l'IA, les algorithmes de machine learning, et les enjeux éthiques liés à l'utilisation des technologies. Une connaissance des principes de la vie privée, de la transparence des algorithmes, et des implications sociétales de l'IA est également recommandée.

Contenu de la matière :

1. Introduction : enjeux de la révolution numérique
2. Étude critique des problèmes d'éthique associés à l'IA.
3. Les aspects légaux de l'IA : régulations, lois, règles.
4. Procédés de prise de décision éthique de la vie privée, de la confidentialité, de crimes informatiques, responsabilités professionnelles, piratage de logiciel, et l'impact et conséquence de l'informatique sur la société.
5. Responsabilité des réseaux sociaux
6. L'éthique dans l'IoT, cas du véhicule autonome

Mode d'évaluation : Examen (100%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Bahu-Leyser, D., "Une éthique à construire", Hermès, La Revue, n° 53, p. 161-166, 2009, URL :https://www.cairn.info/revue-hermes-la-revue-2009-1-page-161.htm.
2. Davis, K., *"Ethics of Big Data: Balancing risk and innovation"*, O'Reilly Media, Inc.", 2012.
3. CNIL : « Comment permettre à l'homme de garder la main », synthèse de débat public, décembre 2017
4. OPECST : « Pour une intelligence artificielle maîtrisée, utile et démystifiée », rapport, mars 2017
5. CERNA / Allistene : « Ethique de la recherche en apprentissage machine », juin 2017 ;
6. Le Marketing éthique : Les sens du commerce' ; Elisabeth Pastore-Reiss et Hervé Naillon
4. 'Business plan: Concevoir un business plan efficace' de Richard Stutely
5. La communication informatique en toute liberté : Histoire et éthique de l'information numérique Broché ', de Daniel Moatti

Semestre 7
Unité d'enseignement : UEF1
Matière : Représentation des connaissances et raisonnement
Coefficient : 4

Objectifs de l'enseignement

Faire une correspondance entre le monde extérieur et un système souvent symbolique manipulable par un ordinateur,
Raisonner sur ces connaissances. Ainsi, les objectifs de ce module s'insèrent dans cette optique. Différentes théories et formalismes (logiques et graphiques) seront étudiés afin de représenter et de raisonner sur des connaissances de natures multiples.

Connaissances préalables recommandées

Logique mathématique, les fondements de l'Intelligence Artificielle.

Contenu de la matière :

Chapitre I : Logiques non classiques

1. Logique modale
   - 1.1. Notion de modalité
   - 1.2. Syntaxe
   - 1.3. Règles de déduction
   - 1.4. Sémantique : mondes possibles, sémantique de Kripke.
   - 1.5. Représentation des connaissances
   - 1.6. Logique modale et les systèmes multi-agents
2. La logique des défauts
   - 2.1. Notion de non monotonie
   - 2.2. Concepts de base de la logique des défauts
   - 2.3. Calcul des extensions
   - 2.4. La logique des défauts et le data-mining
3. Logiques de description
   - 3.1. Les bases des logiques de description
   - 3.2. Mécanismes de raisonnement
   - 3.3. Logiques de description et le traitement des images
   - 3.4. Logiques de description et les ontologies

Chapitre II : Le raisonnement incertain

1. Gestion de l'incertitude
2. La théorie des fonctions de croyance
   - 2.1. Concepts fondamentaux
   - 2.2. Règles de combinaisons
   - 2.3. Applications
3. Les réseaux bayésiens
   - 3.1. Rappel probabilités (probabilité, probabilité conditionnelle, ..)
   - 3.2. Le théorème de Base
   - 3.3. Définition
   - 3.4. Concepts de base (les dépendances conditionnelles, distribution conjointe de probabilités, probabilité marginale, Independence conditionnelle, …)
4. Utilisations avancées (Apprentissage, graphe de décision,. ;)

Chapitre III : La théorie des sous-ensembles flous

1. Concepts de base
2. Contrôleur flou
3. Domaines d'application de la théorie des sous-ensembles flous

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Bloch, R. Clouard, M. Revenu and O. Sigaud, in A Guided Tour of Artificial Intelligence Research (P. Marquis,
2. O. Papini, H. Prade, Eds), Springer, chap. Artifical Intelligence and Pattern Recognition, Vision, Learning, 2017.
3. Yifan Yang, Jamal Atif, Isabelle Bloch: Raisonnement abductif en logique de description exploitant les domaines concrets spatiaux pour l'interprétation d'images. Revue d' Intelligence Artificielle 31(1- 2) 11-39, 2017.
4. Van Benthem J., Bezhanishvili G. Modal Logics of Space. In: Aiello M., Pratt-Hartmann I., Van Benthem J. (eds) Handbook of Spatial Logics. Springer, Dordrecht, 2007.
5. F. Baader. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.
6. B.Bouchon-Meunier et C.Marsala, Logique floue, principes, aide à la décision, Hermès Sciences, 2002.
7. D. Kayser : La représentation des connaissances. Hermes 1997.
8. F. V. Jensen. An Introduction to Bayesian Networks. University College London: UCL Press, 1996.
9. D.Poole, Default logic. In Handbook of Logic in Artificial Intelligence and Logic Programming: Nonmonotonic Reasoning and Uncertain Reasoning (vol. 3), D. M. Gabbay, C, 1994.
10. J. Hogger, and J. A. Robinson, Eds. Oxford University Press, Inc., New York, NY, 189–215
11. M. Alliot et T.Schiex, Intelligence Artificielle et Informatique Théorique, Cépaduès Editions, 1993.
12. L.Sombe. Raisonnement sur des informations incomplètes en intelligence artificielle. Teknea, Marseille, 1989.
13. P. Besnard, An Introduction to Default Logic. Springer-Verlag, Vienna, Austria, 1989.
14. J.Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publications, 1988.
15. P.Smets, E.H.Mamdani, D.Dubois and H.Prade: Non-standard logics for automated reasoning. 1988.
16. D. Dubois and H. Prade. Possibility Theory. An Approach to Computerized Processing of Uncertainty. Plenum Press, New York and London, 1988.
17. R.Reiter, A logic for default reasoning. Artificial . Intelligence. 13, 81–132, 1980.
18. D. Dubois and H. Prade. Fuzzy Sets and Systems - Theory and Applications. Academic Press, New York, 1980.
19. L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 1978.

Semestre 7
Unité d'enseignement : UEF1
Matière : Calcul haute performance
Coefficient : 3

Objectifs de l'enseignement

Ce cours fournit une base solide en calcul haute performance (HPC) et son rôle dans la science visées d'apprentissage :

- Familiarisez avec les paradigmes de programmation parallèle.
- Etudier les techniques fondamentales de développement d'applications HPC.
- Les plates-formes HPC couramment utilisées.
- Les méthodes de mesure, d'évaluation et d'analyse des performances des applications HPC.

Les étudiants seront initiés aux enjeux liés à l'utilisation des techniques HPC dans la résolution de grands problèmes scientifiques.

Connaissances préalables recommandées

Architectures Evoluées des Ordinateurs, Système d'exploitation I & II.

Contenu de la matière :

Chapitre I : Introduction au calcul parallèle

1. Comprendre la nécessité du calcul parallèle.
2. Les lois fondamentales du calcul parallèle (avantages et limites)
3. Vue d'ensemble des systèmes parallèles et distribués pour le calcul haute performance

Chapitre II : Modèle et structure d'application pour des interfaces de programmation parallèle :

1. Paradigmes pour les applications parallèles.
2. Message Passing Interface (MPI).
3. Programmation parallèle avec OpenMP.
4. Programmation parallèle avec PThreads.

Chapitre III : Calcul générique sur processeur graphique

1. Architectures et concepts GPU
2. Modèle de programmation GPU
3. Programmation des GPUs (CUDA & OpenCL).
4. Approches hybrides.

Chapitre IV : Nouvelles technologies en HPC

1. Informatique quantique
2. Les TPU (Tensor Processing Units)
3. Conception de puces spécialisées pour l'IA
4. Refroidissement liquide et centres de données éco-énergétiques…..

Chapitre V: Techniques d'optimisation et meilleures pratiques dans divers domaines scientifiques.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Georg Hager , Gerhard Wellein (2010) Introduction to High Performance Computing for Scientists and Engineers
2. J. R. PRASAD , P. P. JORVEKAR, S. N. BHOSALE , M. P. NAVALE (2018): HIGH PERFORMANCE COMPUTING.
3. Thomas Rauber et Gudula Rünger (2023): Parallel Programming for Multicore and Cluster Systems"
4. Jordan M.Petkov ( ): High Performance Parallelized Computing
5. C. Bischof, C. Bischof, M. Bucker, P. Gibbon, G. Joubert, T. Lippert (2008): Parallel Computing: Architectures, Algorithms and Applications

Semestre 7
Unité d'enseignement : UEF2
Matière : Machine Learning
Coefficient : 4

Objectifs de l'enseignement

This course aims to provide an understanding of fundamental concepts and learning types, while enabling students to master and implement key algorithms such as linear regression, random forests, and neural networks. It also seeks to develop practical skills for data analysis and building effective predictive models.

Connaissances préalables recommandées

Students should master the basics of mathematics (linear algebra, differential calculus) and programming (especially in Python). An understanding of statistics and fundamental concepts of data systems is also recommended.

Contenu de la matière :

Chapter I: Introduction and Fundamentals

1. Introduction to Machine Learning
   - 1.1. Overview of Machine Learning
   - 1.2. Types of Machine Learning: Supervised, Unsupervised, Semi-supervised, Reinforcement Learning, Self-learning
   - 1.3. Applications of Machine Learning
   - 1.4. **PW**: Set up the development environment and Simple linear regression implementation
2. Probability and Statistics Refresher
   - 2.1. Probability theory basics
   - 2.2. Random variables and distributions
   - 2.3. Bayesian thinking

Chapter II: Supervised Learning

1. Linear Models and Regularization
   - 1.1. Linear regression
   - 1.2. Logistic regression
   - 1.3. Regularization techniques: Ridge (L2), Lasso (L1), Elastic Net
   - 1.4. Advanced topics: Polynomial regression, interaction terms
   - 1.5. **PW**: Implement and compare linear and logistic regression models with regularization
2. Decision Trees and Ensemble Methods
   - 2.1. Decision Trees
   - 2.2. Bagging: Random Forests
   - 2.3. Boosting: Gradient Boosting Machines (GBM), AdaBoost, XGBoost, LightGBM, CatBoost
   - 2.4. Stacking and Blending
   - 2.5. **PW**: Implement and analyze the performance of decision trees and ensemble methods
3. Support Vector Machines and Advanced Optimization
   - 3.1. Introduction to Support Vector Machines (SVM)
   - 3.2. Kernel trick: linear, polynomial, RBF, sigmoid
   - 3.3. Optimization techniques: Stochastic Gradient Descent (SGD), Adam, RMSprop

3.4. **PW**: Implement SVM using a library and experiment with different kernels and optimization techniques

Chapter III: Unsupervised Learning

1. Clustering
   1.1. K-means clustering
   1.2. Hierarchical clustering (Agglomerative, Divisive)
   1.3. Density-Based Spatial Clustering (DBSCAN, OPTICS)
   1.4. Evaluation metrics: Silhouette score, Davies-Bouldin index, Adjusted Rand Index, Mutual Information
   1.5. **PW**: Apply clustering algorithms on a dataset and analyze the results
2. Dimensionality Reduction
   2.1. Principal Component Analysis (PCA)
   2.2. t-SNE
   2.3. Linear Discriminant Analysis (LDA)
   2.4. Independent Component Analysis (ICA)
   2.5. Advanced techniques: UMAP, Factor Analysis
   2.6. **PW**: Implement and apply dimensionality reduction techniques on high-dimensional data

Chapter IV: Specialized Techniques

1. Bayesian Networks and Gaussian Processes
   1.1. Introduction to Bayesian Networks
   1.2. Structure learning and parameter learning in Bayesian Networks
   1.3. Gaussian Processes: Regression and classification
   1.4. Applications and examples
   1.5. **PW**: Implement a Bayesian Network for a given dataset
2. Genetic Algorithms and Belief Functions
   2.1. Introduction to Genetic Algorithms
   2.2. Fitness function, selection, crossover, mutation
   2.3. Introduction to Belief Functions and Dempster-Shafer Theory
   2.4. Applications in optimization and uncertainty modeling
   2.5. **PW**: Solve an optimization problem using a genetic algorithm

Chapter V: Semi-Supervised and Reinforcement Learning

1. Semi-Supervised Learning
   1.1. Introduction to Semi-Supervised Learning
   1.2. Self-training and Co-training
   1.3. Generative models for semi-supervised learning
   1.4. Semi-supervised SVMs
   1.5. **PW**: Implement a semi-supervised learning algorithm on a partially labeled dataset
2. Reinforcement Learning
   2.1. Fundamentals of Reinforcement Learning
   2.2. Markov Decision Processes (MDPs)
   2.3. Q-Learning, SARSA
   2.4. Policy Gradient Methods
   2.5. Applications: Game playing, Robotics
   2.6. **PW**: Implement a Q-learning algorithm for a simple game environment

Chapter VI: Feature Engineering and Anomaly Detection

1. Feature Engineering and Preprocessing
   1.1. Data preprocessing techniques
   1.2. Feature scaling and normalization
   1.3. Handling missing values and categorical data

1.4. Feature selection methods: Recursive Feature Elimination, SelectKBest, L1-based feature selection
1.5. Feature extraction: PCA, LDA, t-SNE
1.6. **PW**: Perform data cleaning and preprocessing on a real-world dataset

2. Anomaly Detection
2.1. Types of anomalies: point, contextual, collective
2.2. Statistical methods: Z-score, Grubbs' test
2.3. Machine learning-based methods: Isolation Forest, One-Class SVM, Local Outlier Factor (LOF), Autoencoders for anomaly detection
2.4. Evaluation metrics: Precision at N, Receiver Operating Characteristic (ROC) for anomaly detection, Area Under Curve (AUC-ROC), F1 score for imbalanced data
2.5. **PW**: Detect anomalies in a given dataset using different methods

Chapter VII: Recommender Systems and Association Rules

1. Recommender Systems
1.1. Content-based filtering
1.2. Collaborative filtering: User-based, Item-based
1.3. Matrix Factorization techniques: SVD, ALS, NMF
1.4. Hybrid methods
1.5. Evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Precision and Recall at K, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG)
1.6. **PW**: Build a simple recommender system using collaborative filtering and matrix factorization

2. Association Rule Learning
2.1. Market basket analysis
2.2. Apriori algorithm
2.3. Eclat algorithm
2.4. FP-Growth
2.5. Evaluation metrics: Support, Confidence, Lift, Conviction
2.6. **PW**: Implement and analyze association rules on transactional data

Chapter VIII: Privacy and Ethics

1. Federated Learning and Ethical Considerations
1.1. Introduction to Federated Learning
1.2. Federated averaging algorithm
1.3. Privacy-preserving techniques: Differential Privacy, Secure Multiparty Computation
1.4. Ethical issues in Machine Learning: bias, fairness, accountability
1.5. Model interpretability and fairness: SHAP values, LIME, Fairness-aware ML
1.6. Deployment and scalability of ML models: Model serving, A/B testing, Continuous Integration/Continuous Deployment (CI/CD) for ML
1.7. **PW**: Implement a simple federated learning system

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. "Pattern Recognition and Machine Learning", Christopher M. Bishop, Cambridge University Press, 2021 (Reprint)
2. "Deep Learning", Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 2016 (Updated Edition)
3. "Machine Learning: A Probabilistic Perspective", Kevin P. Murphy, MIT Press, 2012 (Updated Edition)
4. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron, O'Reilly Media, 2022 (2nd Edition)
5. "Introduction to Machine Learning with Python: A Guide for Data Scientists", Andreas C. Müller, Sarah Guido, O'Reilly Media, 2016
6. "Machine Learning Yearning", Andrew Ng, (Available online as a free ebook)

Semestre 7
Unité d'enseignement : UEF2
Matière : Modélisation & Simulation
Coefficient : 3

Objectifs de l'enseignement

Ce module est destiné à approfondir les connaissances de l'étudiant dans le domaine de la modélisation et la simulation. De plus, il initie aux techniques d'évaluation des performances.

Connaissances préalables recommandées

Les étudiants doivent avoir des connaissances préalables en mathématiques (notamment en calcul différentiel et intégral, algèbre linéaire et statistiques). Ils doivent également être familiers avec les concepts fondamentaux des systèmes dynamiques et des équations différentielles. Une expérience en programmation, de préférence dans des langages comme Python, MATLAB ou R, est également recommandée pour la mise en œuvre de modèles et la réalisation de simulations.

Contenu de la matière :

Chapitre I : Modélisation des systèmes

1. Types de systèmes (discret, continu, déterministe, ...)
2. Outils de modélisation (machine d'états finis, réseaux de Petri, modèles des files d'attentes)

Chapitre II : Techniques d'évaluation des performances

1. Présentation des techniques
2. Les méthodes mathématiques
3. La simulation (introduction)

Chapitre III : La simulation

1. Types de simulation
2. Simulation de systèmes dynamiques
3. Simulation continue
4. Simulation des systèmes discrets
5. Echantillonnage
6. Génération de nombres pseudo-aléatoires
7. Les tests de générateurs de nombres aléatoires
8. Analyse et validation des résultats d'une simulation

Chapitre IV : Les outils de simulation

1. Logiciels
2. Langages
3. Le graphisme et la simulation

Chapitre V : Etude d'un langage de simulation

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. S. S. Lavenberg, Computer systems performance evaluation, Academic press 1983
2. Mitrani, Modeling of computer and communication systems, Cambridge university press 1987
3. M. Pidd, Computer simulation and management science, J. Wiley and Sons Ed. 1984
4. K. S. Trivedi, Probability and statistics with reliability, queing and computer science applications, Prentice Hall 1982

Semestre 7
Unité d'enseignement : UEM1
Matière : Business Intelligence
Coefficient : 2

Objectifs de l'enseignement

Ce cours vise la découverte des différentes facettes du processus décisionnel et des modèles et outils associés et la maîtrise de la technologie des Data Warehouses comme application d'un système décisionnel**.**

Connaissances préalables recommandées

Bases de données et système d'information

Contenu de la matière :

Chapitre I : Introduction aux systèmes d'aide à la décision

1. Concepts de base
2. Historique
3. Architectures décisionnelles

Chapitre II : Modélisation multidimensionnelle :

1. OLAP : concepts et règles
2. Niveau conceptuel
3. Niveau logique
4. Niveau physique
5. Méthode de conception multidimensionnelle

Chapitre III : Intégration et administration

1. Extraction
2. Transformation
3. Chargement

Chapitre IV : Exploitation des données décisionnelles

4. Manipulation des données multidimensionnelles
5. SQL analytique
6. Tableaux de bord

Chapitre V : Optimisation

1. Classification des techniques d'optimisation
2. Les vues matérialisées
3. Les index
4. La fragmentation.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. W. Inmon. *Building the Data Warehouse*. John Wiley & Sons Publishers, 2002
2. Thomas Connolly, Carolyn Begg: Systèmes de bases de données : approche pratique de la conception, de l'implémentation et de l'administration, Eyrolles 2005
3. R. Kimball, L. Reeves, M. Ross, W. Thornthwaite, "*Concevoir et déployer un data warehouse"*, Editions Eyrolles, 2000.
4. Elzbieta Malinowski, Esteban Zimányi: "Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications" (Data-Centric Systems and applications), 2008.
5. Vassil Guliashki, Edjola Naka: "Optimization Techniques in Data Management: A Survey." ICDDE, 2021.

Semestre 7
Unité d'enseignement : UEM1
Matière : Recherche opérationnelle
Coefficient : 2.

Objectifs de l'enseignement

Le module vise à fournir aux étudiants une compréhension approfondie des méthodes d'optimisation et de modélisation des problèmes complexes. Les objectifs incluent l'apprentissage des techniques d'optimisation linéaire et non linéaire, la formulation de modèles mathématiques pour divers problèmes réels, et l'utilisation d'outils logiciels pour résoudre ces modèles et interpréter les résultats.

Connaissances préalables recommandées

Les étudiants doivent avoir des bases solides en mathématiques, notamment en algèbre linéaire et en calcul différentiel. Une compréhension des statistiques et des probabilités, ainsi qu'une expérience en programmation pour utiliser des logiciels d'optimisation, sont également recommandées.

Contenu de la matière :

Chapitre I : Introduction à la recherche opérationnelle

1. Définition et objectifs de la recherche opérationnelle
2. Introduction à la programmation mathématique classique
3. Formulation d'un programme linéaire
4. Algorithme et méthode du simplexe
5. Théorème de dualité

Chapitre II : Programmation mathématique classique : application aux problèmes posés en variables entières et/ou booléennes

1. Introduction à la programmation en nombres entiers
2. Problèmes de programmation en nombres entiers et booléens
3. Techniques de résolution (méthode des branches et des bornes, programmation dynamique, etc.)

Chapitre III : Ordonnancement

1. Méthodes MPM (Méthode du Plus Tôt / Plus Tard)
2. Méthode PERT (Program Evaluation and Review Technique)

Chapitre IV : Eléments de programmation convexe

1. Introduction à la programmation convexe
2. Méthode de Franck et Wolfe
3. Méthode des plans sécants de Kelley

Chapitre V : Eléments de programmation sans contraintes

1. Introduction à la programmation sans contraintes
2. Méthodes de gradient
3. Méthodes directes
4. Méthodes par essaims particulaires et autres méthodes modernes (heuristiques et méta-heuristiques)

Chapitre VI : Modélisation et résolution de problèmes réels

1. Étude de cas et projets pratiques
2. Utilisation de logiciels de modélisation et d'optimisation (comme AMPL, GAMS, MATLAB, etc.)
3. Analyse et interprétation des résultats

Chapitre VII : Avancées en recherche opérationnelle

1. Techniques émergentes et tendances actuelles en recherche opérationnelle

2. Recherche opérationnelle et intelligence artificielle : applications des méthodes d'optimisation dans l'IA
3. Impacts de la recherche opérationnelle sur les industries et les secteurs économiques

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. "Introduction to Operations Research", Frederick S. Hillier, Gerald J. Lieberman, McGraw-Hill Education, 2022 (10th Edition)
2. "Operations Research: An Introduction", Taha H. operations, Pearson, 2017 (10th Edition)
3. "Linear Programming and Network Flows", Gerald B. Dantzig, John S. Santa, Wiley, 2018 (2nd Edition)
4. "Optimization in Operations Research", David L. Poole, Alan K. Mackworth, Cambridge University Press, 2019
5. "Introduction to Linear Optimization", Dimitris Bertsimas, Robert Weismantel, MIT Press, 2021.

Semestre 7
Unité d'enseignement : UET1
Matière : Techniques de rédaction
Coefficient : 2

Objectifs de l'enseignement

Cet enseignement porte sur les Techniques d'Expression et de Communication Scientifique et Technique (TEC) liées aux études (Communications écrites et orales : élaboration et présentation de documents écrits et visuels pour la soutenance du projet de fin d'études et la présentation orale. L'enseignement TEC vise à accompagner l'étudiant tout au long de son cursus de chercheur et à développer et renforcer ses compétences cognitives, attitudinales, comportementales inhérentes à la recherche scientifique, à la valorisation des travaux réalisés et aux modalités de la diffusion des informations scientifiques.
Pour le TP, l'objectif est de pratiquer les concepts de base du latex pour préparer des documents de grande qualité.

Connaissances préalables recommandées

Les étudiants doivent maîtriser les règles de grammaire, de syntaxe, et de style, ainsi que savoir structurer et organiser leurs idées de manière cohérente. Une connaissance des normes de rédaction académique et une expérience en recherche documentaire sont également recommandées.

Contenu de la matière :

Chapitre I : Expression & communication

1. Introduction

Présentation des outils de communication.

2. Communications écrites :
   - 2.1. Rédaction d'écrits scientifiques et techniques.
   - 2.2. Rédaction et présentation d'un cadre théorique pour un mémoire de recherche scientifique.
   - 2.3. Rédaction et présentation d'un cadre méthodologique pour un mémoire de recherche scientifique.
   - 2.4. Rédaction, explication et interprétation de données pour un mémoire de recherche scientifique.
   - 2.5. Rédaction d'une introduction et d'une conclusion d'un mémoire de recherche scientifique.
   - 2.6. Présentation de la bibliographie d'un mémoire de recherche scientifique.
   - 2.7. Présentation finale d'un mémoire de recherche scientifique.
3. 2. Communications orales :
   - 3.1. Préparation de communications orales (Exposés, table ronde, colloque, séminaires, congrès, missions de radios et de télévision…).
   - 3.2. Présentation d'une communication orale publique (Exposés, table ronde, colloque, séminaires, congrès, missions de radios et de télévision…).
   - 3.3. Préparation aux aspects oraux spécifiques à la soutenance d'un PFE.
4. 3. Préparation au devenir professionnel :
   - 4.1. Rédaction d'un curriculum vitae.
   - 4.2. Elaboration et présentation d'un dossier de candidature à un emploi.
   - 4.3. Préparation à un entretien d'embauche public et privé.
   - 4.4. Elaboration d'un projet professionnel ; d'un projet et plan de carrière.

Chapitre II : Latex

5. Les bases de latex
6. Documents latex
7. Les tableaux
8. Les images
9. Les équations
10. Bibliographie et références

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. ECKENSCHWILLER, Michèle.- L'écrit universitaire, Paris, Editions d'Organisation, 1994.
2. FINTZ, Claude ; RISPAIL, Marielle.-Le Français dans le supérieur ; fiches méthodologiques.- Paris : Ellipses, 1997.
3. REOL, Gilles; FLAGEUL, Noë1.- Méthodes et techniques de l'expression écrite et orale, Paris, A. Colin, 1996.

Semestre 8
Unité d'enseignement : UEF1
Matière : Traitement Automatique du Langage Naturel
Coefficient :3

Objectifs de l'enseignement

The course covers many topics, including preprocessing, text representation and classification, syntactic and semantic analysis, sentiment analysis, and language modeling. It allows students to apply different NLP techniques to deal with practical problems such as machine translation, text summarization, information extraction, information retrieval and question answering.

Connaissances préalables recommandées

Machine Learning.

Contenu de la matière :

Chapter I: Introduction to Natural Language Processing

1. What is NLP?
2. History of NLP
3. Applications of NLP

Chapter II: Text preprocessing

1. Text cleaning
2. Text tokenization
3. Text normalization

Chapter III: Text classification

1. Learning algorithms
   1.1. Naïve Bayes
   1.2. Logistic regression
   1.3. Neural networks
2. 2. Classification
   2.1. Sentiments and analysis opinions
   2.2. Word sense disambiguation

Chapter IV: Text representation

1. Bag-of-words
2. Term frequency-Inverse document frequency
3. Word embeddings
4. Contextualized word embeddings

Chapter V: Sequence Labeling

1. Viterbi algorithm
2. Hidden Markov Models
3. Conditional Random Fields
4. Named entity recognition
5. Part-of-speech tagging

Chapter VI : Language models

1. N-gram language models
2. Neural networks language models
3. Feedforward neural networks as language models
4. Recurrent neural networks as language models

5. Large language models
    5.1. Pretraining and fine-tuning large language models
    5.2. T5 as a translation large language model
    5.3. ChatGPT as generative large model.
6. Evaluation of language models

Chapter VII : Parsing and meaning

1. Context-Free Grammars
2. Dependency Parsing
3. Logical representation of meaning
4. Semantic parsing

Chapter VIII : Applications of NLP

1. Machine translation
2. Text summarization
3. Text generation
4. Information extraction
5. Information retrieval
6. Question Answering

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Jurafsky, D & Martin, J.H. (2023). Speech & language processing (3rd ed. draft).
2. Eisenstein, J. (2018). Natural language processing. Jacob Eisenstein.
3. Tunstall, L., Von Werra, L., and Wolf, T. (2022). Natural language processing with transformers. O'Reilly Media, Inc.
4. Goldberg, Y. (2016). A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research, 57, 345-420.

Semestre 8
Unité d'enseignement : UEF1
Matière : Deep Learning
Coefficient : 4

Objectifs de l'enseignement

The Deep Learning course aims to provide a thorough understanding of deep neural network architectures and advanced machine learning techniques. Objectives include learning key concepts such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and the implementation and optimization of these models to solve complex problems in image, text, and sequential data processing.

Connaissances préalables recommandées

Students should master the basic concepts of machine learning, optimization algorithms, and have programming skills with libraries such as TensorFlow or PyTorch. A solid understanding of mathematics, particularly in linear algebra, differential calculus, and statistics, is also required..

Contenu de la matière :

Chapter 1: Introduction and Fundamentals

1. Introduction to Deep Learning
   - 1.1. Overview of Deep Learning
   - 1.2. History and evolution of Deep Learning
   - 1.3. Key differences between Deep Learning and traditional Machine Learning
   - 1.4. Applications of Deep Learning
   - 1.5. **PW**: Set up the development environment (Python, TensorFlow, PyTorch) and Simple neural network implementation using a deep learning framework
2. Mathematics for Deep Learning
   - 2.1. Linear algebra for Deep Learning: matrices, vectors, eigenvalues, and eigenvectors
   - 2.2. Calculus for optimization: gradients, partial derivatives
   - 2.3. Probability and statistics refresher for Deep Learning

Chapter 2: Feedforward Neural Networks

1. Fundamentals of Neural Networks
   - 1.1. Perceptron and multi-layer perceptron (MLP)
   - 1.2. Activation functions: sigmoid, tanh, ReLU, Leaky ReLU
   - 1.3. Forward and backward propagation
   - 1.4. Loss functions: MSE, cross-entropy
   - 1.5. **PW**: Implement a simple feedforward neural network from scratch
2. Training Neural Networks
   - 2.1. Gradient descent and its variants: SGD, Momentum, Nesterov, Adagrad, RMSprop, Adam
   - 2.2. Overfitting and regularization: L2, L1, Dropout, Batch Normalization
   - 2.3. Hyperparameter tuning and optimization
   - 2.4. **PW**: Train a neural network on a dataset and experiment with different optimization techniques and regularization methods

Chapter 3: Convolutional Neural Networks (CNNs)

1. Introduction to CNNs
   - 1.1. Convolution operations and filters
   - 1.2. Pooling layers: max pooling, average pooling
   - 1.3. CNN architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet

1.4. **PW**: Implement a simple CNN for image classification
2. Advanced CNN Architectures and Techniques
   2.1. Transfer learning and fine-tuning
   2.2. Data augmentation techniques
   2.3. Advanced architectures: DenseNet, MobileNet, EfficientNet
   2.4. **PW**: Fine-tune a pre-trained CNN on a new dataset

Chapter 4: Recurrent Neural Networks (RNNs)

1. Fundamentals of RNNs
   1.1. Introduction to Recurrent Neural Networks
   1.2. Backpropagation through time (BPTT)
   1.3. Vanishing and exploding gradients
   1.4. **PW**: Implement a simple RNN for sequence prediction
2. Advanced RNN Architectures
   2.1. Long Short-Term Memory (LSTM) networks
   2.2. Gated Recurrent Units (GRUs)
   2.3. Bidirectional RNNs
   2.4. Applications: time series forecasting, language modeling, speech recognition
   2.5. **PW**: Implement an LSTM or GRU for a sequence-based task

Chapter 5: Unsupervised Deep Learning

1. Autoencoders and Variational Autoencoders
   1.1. Introduction to autoencoders
   1.2. Variational Autoencoders (VAEs)
   1.3. Applications: anomaly detection, image denoising, dimensionality reduction
   1.4. **PW**: Implement an autoencoder for image reconstruction
2. Generative Models
   2.1. Generative Adversarial Networks (GANs)
   2.2. Conditional GANs (cGANs)
   2.3. Advanced GAN architectures: DCGAN, CycleGAN, StyleGAN
   2.4. Applications: image generation, style transfer
   2.5. **PW**: Implement a simple GAN for image generation

Chapter 6: Advanced Topics in Deep Learning

1. Attention Mechanisms and Transformers
   1.1. Introduction to attention mechanisms
   1.2. Sequence-to-sequence models
   1.3. Transformers: architecture, self-attention, multi-head attention
   1.4. Applications: machine translation, text summarization
   1.5. **PW**: Implement a transformer model for a natural language processing task
2. Reinforcement Learning with Deep Learning
   2.1. Deep Q-Learning
   2.2. Policy gradients and actor-critic methods
   2.3. Advanced techniques: Double DQN, Dueling DQN, A3C, PPO
   2.4. **PW**: Implement a Deep Q-Network for a simple game environment

Chapter 7: Specialized Deep Learning Applications

1. Natural Language Processing (NLP) with Deep Learning
   1.1. Word embeddings: Word2Vec, GloVe, FastText
   1.2. Recurrent models for NLP: RNNs, LSTMs, GRUs
   1.3. Transformers for NLP: BERT, GPT, T5
   1.4. **PW**: Implement a text classification model using a pre-trained transformer
2. Computer Vision with Deep Learning
   2.1. Object detection: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD

2.2. Semantic segmentation: Fully Convolutional Networks (FCNs), U-Net, SegNet
2.3. Image captioning and visual question answering
2.4. **PW**: Implement an object detection model on a custom dataset

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. "Deep Learning", Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 2016 (Updated Edition)
2. "Deep Learning for Computer Vision", Rajalingappaa Shanmugamani, Packt Publishing, 2021
3. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron, O'Reilly Media, 2022 (2nd Edition)
4. "Deep Learning with Python", François Chollet, Manning Publications, 2021 (Updated Edition)
5. "Neural Networks and Deep Learning: A Textbook", Charu C. Aggarwal, Springer, 2018

Semestre 8
Unité d'enseignement : UEF2
Matière : Analyse et traitement d'image
Coefficient : 3

Objectifs de l'enseignement

Ce cours traite de l'acquisition, du traitement et de l'analyse d'images numériques réelles. Il s'agit de modéliser les processus nécessaires à la transformation de ces images discrètes (ensemble de pixels) en donnés perceptuelles et sémantiques (scène d'objets). Ce cours traite en particulier les problèmes d'acquisition.

Connaissances préalables recommandées

Les étudiants doivent avoir des connaissances en mathématiques, notamment en algèbre linéaire, calcul différentiel, et statistiques. Une maîtrise des concepts de base en traitement du signal et en programmation, idéalement avec des outils comme Python et des bibliothèques telles que OpenCV ou scikit-image, est également recommandée.

Contenu de la matière :

Chapitre I: Généralités sur l'image

1. Image numérique.
2. Caractéristiques de l'image numérique.
3. Types d'images et fichiers graphiques.
4. Les grands domaines de l'imagerie numérique

Chapitre II : Acquisition des Images

1. Système visuel humain
2. Outils d'acquisitions
3. Caractéristiques d'un capteur.
4. Les formats d'images.
5. Les bruits.

Chapitre III : Numérisation d'une image

1. Codage d'image.
2. Quantification, échantillonnage d'image.
3. Rehaussement, restauration d'image.
4. Transformations ponctuelles et globales.
5. Convolution.
6. Détection des points et des contours.

Chapitre IV : Amélioration d'images par histogrammes

1. Histogramme des images en niveau de gris.
2. Histogramme des images en couleurs.
3. Informations issues de l'histogramme (pics, luminosités, contrast…etc.)
4. Transformations de l'histogramme : amélioration du contraste, expansion de la dynamique, seuillage, accentuation de contraste, correction gamma, égalisation,…etc.

Chapitre V : Techniques d'analyses et de traitement d'images

1. Qu'est-ce que l'analyse et le traitement d'image ?
2. Le processus de traitement d'image.
3. Technique d'analyses d'images
4. Techniques de traitement d'images

Chapitre VI : Les techniques de filtrage

1. Introduction au filtrage (dé-bruitage).
2. Filtre moyenneur Vs filtre médian.
3. Filtre moyenneur pondérée Vs filtre gaussien.
4. Filtre adaptatif.
5. Récapitulatif sur les filtres: linéaire et non linaire
6. Quelques filtres supplémentaires : amélioration des bords, gaufrage, effet peinture, filtre marr, filtre de kirsch, filtre de Nagao, filtrage par maximum

Chapitre VII : Segmentation

1. Définition
2. Types de segmentations
   2.1. Contour
   2.2. Régions
   2.3. Hybrides

Chapitre VIII : les opérateurs morphologiques

1. Définition
2. Elément structurant
3. Erosion et Dilatation binaire
4. Ouverture et fermeture binaire
5. Dilatation conditionnelle

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. « Introduction au traitement numérique des images s», Humbert Florent, 2007
2. « Introduction Au Traitement d'Images», Diane Lingrand, code : L/004.1057.
3. Computer Vision, A Modern Approach (Forsyth & Ponce) : chap. 6: Color & chap. 4: Radiometry - Measuring Light, 2003.
4. Digital Image Processing (Gonzalez & Woods) : chap. 6: Color Image Processing, second edition. Prentice-Hall, 2002.
5. « Image Processing, Analysis, and Machine Vision», Milan Sonka, Vaclav Hlavac, et Roger Boyle, Amazon, 2007.
6. « Handbook of Machine Vision", Lavoisier, HORNBERG Alexander, LETOKHOV Vladilen S., KLIMOV .V 2006.
7. « Principles of Digital Image Processing », Wilhelm Burger, Mark J. Burge

Semestre 8
Unité d'enseignement : UEF2
Matière : Sécurité des données
Coefficient : 3

Objectifs de l'enseignement

- Étudier les différents types d'attaques auxquels un système informatique peut faire face.
- Connaître les types d'attaques contradictoires en ML
- Protéger les systèmes d'apprentissage automatique contre lesattaques adverses
- Étudier les principes et les techniques de la cryptographie.

Connaissances préalables recommandées

Les aspects fondamentaux de la sécurité informatique.

Contenu de la matière :

Chapitre I : rappels sur les concepts de base de la sécurité

1. Objectifs de la sécurité.
2. La triade de sécurité d'information (CIA)
3. Les acteurs de la menace
4. Les types de malware
5. Reconnaître les attaques courantes
   5.1. Social Engineering
   5.2. Shoulder Surfing
   5.3. Tricking Users with Hoaxes
   5.4. Tailgating and Mantraps
   5.5. Dumpster Diving
   5.6. Watering Hole Attacks
   5.7. Attacks via Email and Phone (spam, phishing, spear phishing, and whaling)
6. Bloquer les Malwares et d'autres attaques
   6.1. Protecting Systems from Malware
   6.2. Antivirus and Anti-Malware Software
   6.3. Signature-Based Detection
   6.4. Heuristic-Based Detection
   6.5. Checking File Integrity
   6.6. Data ExecutionPrevention
   6.7. Advanced Malware Tools
   6.8. Spam Filters
7. Former les utilisateurs
   1. New Viruses, Phishing Attacks, Zero-Day vulnerabilities et Exploits

Chapitre II : Crypto-système et science de la cryptographie

1. Cryptographie symétrique
   1.1. Chiffrement par blocs (DES, AES) et chiffrement par flots
   1.2. Modes opératoires (ECB, CBC, CFB, OFB, CTR)
   1.3. Gestion des clés symétriques
2. Cryptographie asymétrique
   2.1. Principes de la cryptographie à clé publique
   2.2. Algorithmes de chiffrement asymétriques (RSA, ElGamal)
   2.3. Algorithmes de signature numérique (DSA, ECDSA)

2.4. Échange de clés et protocoles associés (Diffie-Hellman, signatures de clés)
3. Fonctions de hachage cryptographiques
   3.1. Propriétés des fonctions de hachage sécurisées (MD5, SHA-2, SHA-3)
   3.2. Applications des fonctions de hachage (HMAC, dérivation de clés, etc.)
4. Protocoles cryptographiques
   4.1. Protocoles d'authentification (Kerberos, SSL/TLS)
   4.2. Protocoles d'échange de clés (IKE/IPsec)
   4.3. Protocoles de paiement sécurisé (SET, 3D Secure)
5. Infrastructures à clé publique (PKI)
   5.1. Concept d'autorité de certification
   5.2. Gestion de cycle de vie des certificats
   5.3. Normes et protocoles PKI (X.509, PKCS, OCSP, etc.)
6. Attaques contre les crypto-systèmes
   6.1. Types d'attaques (cryptanalyse, attaques par canaux auxiliaires, etc.)
   6.2. Contre-mesures et bonnes pratiques de mise en œuvre

Chapitre III: système de contrôle d'accès

1. Introduction au contrôle d'accès
   1.1. Définition et objectifs du contrôle d'accès
   1.2. Principes de base (moindre privilège, séparation des tâches, etc.)
   1.3. Matrice de contrôle d'accès
2. Modèles de contrôle d'accès
   2.1. Contrôle d'accès discrétionnaire (DAC)
   2.2. Contrôle d'accès obligatoire (MAC)
   2.3. Contrôle d'accès basé sur les rôles (RBAC)
   2.4. Contrôle d'accès basé sur les attributs (ABAC)
3. Mécanismes de contrôle d'accès
   3.1. Listes de contrôle d'accès (ACL)
   3.2. Capacités et jetons
   3.3. Règles de contrôle d'accès
4. Authentification et autorisation
   4.1. Processus d'authentification (facteurs d'authentification)
   4.2. Protocoles d'authentification (Kerberos, RADIUS, SAML, OAuth)
   4.3. Gestion des autorisations (permissions, privilèges, etc.)
5. Contrôle d'accès physique
   5.1. Contrôles d'accès aux installations et équipements
   5.2. Systèmes de contrôle d'accès physique (badges, biométrie, etc.)
6. Contrôle d'accès logique
   6.1. Contrôle d'accès aux systèmes d'exploitation
   6.2. Contrôle d'accès aux applications et bases de données
   6.3. Journalisation et audit des accès
7. Gestion des identités et des accès
   7.1. Provisionnement et déprovisionnement des accès
   7.2. Gestion des cycles de vie des identités
   7.3. Solutions d'Identity and Access Management (IAM)
8. Administration et surveillance du contrôle d'accès
   8.1. Revue et analyse des accès
   8.2. Outils de surveillance et rapports
   8.3. Gestion des violations et incidents

Chapitre IV : Gestion de risque

1. Identification des risques

1.1. Définition et catégories des risques (techniques, organisationnels, physiques, etc.)
1.2. Sources de risques (menaces, vulnérabilités, impacts potentiels)
1.3. Inventaire des actifs à protéger (données, systèmes, infrastructures, etc.)
1.4. Méthodologies d'identification des risques (EBIOS, MEHARI, etc.)

2. Gestion des risques
2.1. Principes de la gestion des risques (ISO 31000)
2.2. Processus de gestion des risques (établir le contexte, évaluation, traitement, suivi)
2.3. Intégration de la gestion des risques dans la gouvernance de la sécurité

3. Méthodes d'analyse et d'évaluation des risques
3.1. Analyses qualitatives (matrices, diagrammes, etc.)
3.2. Analyses quantitatives (modèles, calculs de risques, etc.)
3.3. Méthodologies d'évaluation des risques (EBIOS RM, MEHARI, OCTAVE, etc.)

4. Évaluations de la vulnérabilité
4.1. Définition et types de vulnérabilités (techniques, humaines, procédures, etc.)
4.2. Processus d'évaluation des vulnérabilités
4.3. Outils d'analyse de vulnérabilités (scanners, évaluations manuelles, etc.)

5. Analyses de vulnérabilité
5.1. Analyses statiques (code source, configurations, etc.)
5.2. Analyses dynamiques (tests d'intrusion, fuzzing, etc.)
5.3. Analyses de surfaces d'attaque (vecteurs, chemins d'accès, etc.)

6. Tests de pénétration
6.1. Objectifs et types de tests de pénétration
6.2. Méthodologies de tests (boîte noire, boîte blanche, etc.)
6.3. Phases des tests (reconnaissances, exploitation, post-exploitation, etc.)
6.4. Outils et techniques utilisés par les pentesteurs

7. Outils de sécurité
7.1. Outils de surveillance et de détection (SIEM, IDS/IPS, etc.)
7.2. Outils de gestion des vulnérabilités et des patchs
7.3. Outils de chiffrement et de gestion des clés
7.4. Outils de contrôle d'accès et d'authentification
7.5. Solutions de sauvegarde et de reprise après sinistre

Chapitre V: Apprentissage automatique : attaques et défense adverses

1. Introduction à l'apprentissage automatique contradictoire (Adversarial Machine Learning)
1.1. Définition et concepts clés
1.2. Vulnérabilités des modèles d'apprentissage automatique aux attaques adverses
1.3. Importance de la sécurité des systèmes d'IA dans divers domaines

2. Différences entre les attaques contradictoires dans les domaines de la vision par ordinateur et de la cybersécurité
2.1. Attaques dans la vision par ordinateur (perturbations d'images, adversarial examples)
2.2. Attaques en cybersécurité (contournement de détecteurs de malwares, d'IDS, etc.)
2.3. Différences d'objectifs, de contraintes et d'impacts

3. Types d'attaques contradictoires en ML
3.1. Attaques par perturbation (ajout de bruit imperceptible)
3.2. Attaques par manipulation des données d'entraînement (data poisoning)
3.3. Attaques par extraction de modèles (model stealing)
3.4. Attaques par inversion de modèles (model inversion)

4. Attaques durant la phase d'entraînement
4.1. Empoisonnement des données d'entraînement (backdoor, data pollution)
4.2. Attaques par évasion de modèle (évasion durant l'entraînement)
4.3. Attaques par extraction de modèles (model stealing)

5. Attaques durant la phase d'inférence
   5.1. Attaques par perturbation (adversarial examples, adversarial patches)
   5.2. Attaques par inversion de modèles (model inversion)
   5.3. Attaques par contournement (adversarial reprogramming)
6. Défenses contre les attaques adverses
   6.1. Méthodes proactives (entraînement robuste, data sanitization, etc.)
   6.2. Méthodes réactives (détection, nettoyage, correction, etc.)
   6.3. Vérification formelle et certification de robustesse

Chapitre VI: Sécurité du cloud

1. Risques et défis de sécurité dans le cloud
   1.1. Risques liés aux données (confidentialité, intégrité, disponibilité)
   1.2. Risques liés à l'accès et aux identités (usurpation d'identité, accès non autorisés)
   1.3. Risques liés à la perte de contrôle et de visibilité
   1.4. Risques liés à la multi-locataire et à l'isolation insuffisante
   1.5. Risques liés à la supply chain et aux fournisseurs tiers
   1.6. Autres risques (conformité, gouvernance, lock-in, etc.)
2. Modèles de services cloud
   2.1. IaaS (Infrastructure as a Service) : machines virtuelles, stockage, réseaux
   2.2. PaaS (Platform as a Service) : environnements d'exécution, bases de données, etc.
   2.3. SaaS (Software as a Service) : applications cloud, boîtes de réception partagées
   2.4. Responsabilités de sécurité selon le modèle de service
3. Sécurité des charges de travail et du réseau dans le cloud
   3.1. Sécurité des machines virtuelles et conteneurs (configuration, mise à jour, etc.)
   3.2. Sécurité du stockage objet et des bases de données cloud
   3.3. Sécurité des réseaux virtuels et des contrôles d'accès réseau
   3.4. Gestion des identités et des accès dans le cloud (IAM)
   3.5. Surveillance de la sécurité et détection des menaces
4. Conformité cloud
   4.1. Réglementations applicables (RGPD, CCPA, HIPAA, PCI-DSS, etc.)
   4.2. Normes de sécurité cloud (ISO 27017, 27018, CSA STAR, etc.)
   4.3. Programmes de conformité et certification des fournisseurs cloud
   4.4. Responsabilités partagées et gestion des risques résiduels
   4.5. Audits de sécurité et attestations de conformité
5. Sécurité du cloud hybride et multi-cloud
   5.1. Déploiements hybrides (cloud public, privé, on-premise)
   5.2. Stratégies multi-cloud et gestion des fournisseurs multiples
   5.3. Interconnexion et passerelles de sécurité cloud
   5.4. Gestion unifiée de la sécurité multi-cloud
6. Pratiques et architectures pour la sécurité cloud
   6.1. Security by Design et DevSecOps pour le cloud
   6.2. Principes de moindre privilège et de défense en profondeur
   6.3. Architectures de référence sécurisées (bien configurées, Cloud Native)
   6.4. Automatisation et gestion du cycle de vie de la sécurité

Chapitre VII : Défis et enjeux de la sécurité dans les nouveaux paradigmes émergents (objets connectés, données massives, etc.)

1. Sécurité de l'Internet des Objets (IoT)
   1.1. Risques de sécurité IoT (faible sécurité par conception)
   1.2. Attaques visant les objets connectés (Mirai, etc.)
   1.3. Sécuriser les déploiements IoT à grande échelle, etc.
2. Sécurité des données massives

2.1. Défis de sécurité posés par les environnements big data
2.2. Risques et menaces pour les environnements big data
2.3. Sécurisation des infrastructures, pipelines et traitements big data
2.4. Protection de la confidentialité des données massives

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Practical Malware Analysis - Andrew Honig and Michael Sikorski. Published: February 29, 2012 by No Starch Press
2. "Cloud Computing Security: Foundations and Challenges" par John R. Vacca (2017)
3. "Security in Cloud Computing" par Xingming Sun, Dong Yuan (2018)
4. "Cloud Security: A Comprehensive Guide to Secure Cloud Computing" par Ronald L. Krutz, Russell Dean Vines (2019)
5. "Internet of Things Security" par Sudarshan Thottan (2020)
6. "The IoT Hacker's Handbook: A Practical Guide to Hacking the Internet of Things" par Aditya Gurudatta (2022)
7. "The Internet of Things: Building Secure Consumer IoT Products" par Justin Rajewski (2022)
8. "Big Data Security" par Obulysamy Manikandan et Mani S. Palani (2021)
9. "Big Data Security and Privacy" par Yuri Demchenko et al. (2016)
10. "Security and Privacy Issues in Big Data" par Yasin N. Silva et al. (2019)

11. Ahmed Abusnaina, Aminollah Khormali, Hisham Alasmary, Jeman Park, Afsah Anwar, and Aziz Mohaisen. 2019. Adversarial learning attacks on graph-based IoT malware detection systems. In Proc. of CDCS, Vol. 10.
12. Naveed Akhtar and Ajmal S. Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access 6 (2018), 14410--14430
13. https://arxiv.org/pdf/2007.02407.pdf
14. https://medium.com/cltc-bulletin/adversarial-machine-learning-43b6de6aafdb
15. https://www.thewolfofallstreets.io/bitcoin-and-the-byzantine-generals-problem/
16. https://hackernoon.com/adversarial-machine-learning-a-beginners-guide-to-adversarial-attacks-and-defenses.

Semestre 8
Unité d'enseignement : UEM1
Matière : Traitement de données massives
Coefficient : 2

Objectifs de l'enseignement

- D'héberger des données massives sur un cluster de machines.
- Écrire un programme permettant un traitement distribué/parallèle sur ces données.
- Traitement en temps réel des données.
- Se familiariser avec les outils généralement utilisés en production.

Connaissances préalables recommandées

Algorithmique, bases de données. Traitement distribué et parallèle. Programmation java

Contenu de la matière :

Chapitre I : Paradigme de traitement des données massives

1. Identification
   1.1. Données massives : Définitions et principes
   1.2. Caractéristiques (5 V) : Volume, Vélocité, Variété, Véracité, Valeur.
2. Modèles de stockages des données massives : SGF Distribués
   2.1. Qu'est-ce qu'un Système de Gestion de fichiers
   2.2. Etude de cas : HDFS Hadoop
      a. Introduction à l'écosystème Hadoop
      b. ii. Hadoop (Définition et Historique)
      c. iii. HDFS (rôle et principe)
      d. iv. HDFS (Commandes usuelles)
      e. v. HDFS (Architecture)
      f. vi. HDFS (Configuration du Cluster)
      g. vii. HDFS (Configuration Hardware du Cluster)
3. Modèle de traitement des données massives : MapReduce
   3.1. Introduction aux modèles de traitement distribué
      a. Traitement distribué avec MapReduce
         - Qu'est-ce que Hadoop MapReduce et pourquoi l'utiliser ?
         - Phases clés de MapReduce
         - Adaptation d'un algorithme pour le modèle MapReduce
         - MapReduce (Use cases)
      b. Traitement en mémoire avec Apache Spark
         - Qu'est-ce qu'Apache Spark et pourquoi l'utiliser ?
         - Utilisation de RDD dans Spark
         - Utilisation de Spark SQL
         - Adaptation d'un algorithme pour le modèle Apache Spark
      c. Installation et configuration de Spark : Hadoop HDFS et Yarn
4. Langages d'abstraction des données
   4.1. Langage d'Abstraction avec Apache Hive
      a. Introduction à Apache Hive (Qu'est-ce que Apache Hive et pourquoi l'utiliser ?)
      b. Le modèle de données Hive (tables, colonnes et types de données, les partitions)
      c. HiveQL
      d. Chargement des données avec Hive
      e. Analyse de données avec Hive (filtrage, agrégation et jointure)
      f. Installation et Configuration de Hive.

4.2. Langage d'Abstraction avec Apache PIG
   a. Introduction à PIG (Qu'est-ce que Apache Pig et pourquoi l'utiliser ?)
   b. Modèle de données PIG (types de données, relations et opérations)
   c. Fonctions PIG (fonctions de base, et fonctions utilisateur UDF)
   d. Langage PIG Latin (syntaxe, commandes et expressions)
   e. Analyse des données avec PIG (chargement, filtrage, transformation, agrégation)
   f. Installation et Configuration de PIG

5. Les bases de données décentralisées
   5.1. Introduction aux bases de données NoSQL.
   5.2. Mécanismes de mise en oeuvre des bases de données NoSQL.
   5.3. Introduction à Apache HBase
   5.4. Modélisation, chargement et interrogation des données avec HBase
   5.5. Installation et configuration de HBase

Chapitre II : Ecosystèmes pour le traitement des données massives

1. Qu'est-ce qu'un écosystème ?
2. Architectures matérielles et virtualisation
3. Etude de cas
   3.1. Hadoop, Spack, Yarn, …. etc

Chapitre III : Intégration des données : Création des pipelines

1. Introduction aux pipelines
2. Création de pipeline avec à Apache Flume et Sqoop
3. Ingestion de données avec Flume (fonctionnement Source, Channel, Sink)
4. Importation et l'exportation de données avec Sqoop
5. Configuration de Flume et Sqoop pour des cas d'utilisation spécifiques
6. Installation et configuration d'Apache Flume et Sqoop

Chapitre IV : Mécanismes de traitement des flux de données continus

1. Introduction au traitement de flux de données en continu
2. Traitement de flux de données en continu avec Spark Streaming
   2.1. Qu'est-ce que Spark Streaming et pourquoi l'utiliser ?
   2.2. Utilisation de DStreams dans Spark Streaming
   2.3. Utilisation de sources de données en continu
   2.4. Utilisation de fenêtres de temps dans Spark Streaming
3. Traitement de flux de données en continu avec Kafka
   3.1. Architecture de Kafka
   3.2. Utilisation de Kafka Producer/Consumer/Connect

Chapitre V : Visualisation des données massives : Dashbord

1. Qu'est-ce qu'un tableau de bord
2. Mise en place d'un tableau de bord pour traitement de données en temps réel (Kafka, Spark Stream, Hive, HDFS, etc.).

Ateliers et TP

- Installation de l'environnement (Machine virtuelle Linux Ubuntu + Docker)
- Configuration d'un Cluster Hadoop avec des conteneurs Docker
- Hadoop HDFS
- Hadoop MapReduce
- Pig
- Hive
- Flume et Sqoop
- Hbase et Hive

- Spark et (Spark+HBase)
- Spark Streaming
- Kafka et (Kafka+Spark)

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Chokogoue, J. (2017). Hadoop: Devenez opérationnel dans le monde du Big Data.
2. White, T. (2012). Hadoop: The Definitive Guide (3rd ed.). Sebastopol, CA: O'Reilly Media.
3. Ting, K., & Cecho, J. J. (2013). Apache Sqoop Cookbook. O'Reilly Media.
4. Narkhede, N., Shapira, G., & Palino, T. (2017). Kafka - The Definitive Guide. Sebastopol, CA: O'Reilly Media.
5. Thomas Erl, Wajid Khattak, and Paul Buhler (2016): Big Data Fundamentals Concepts, Drivers & Techniques.
6. Rajkumar Buyya Ph.D., Rodrigo N. Calheiros, Amir Vahid Dastjerdi (2016) : Big Data: Principles and Paradigms.

Semestre 8
Unité d'enseignement : UEM1
Matière : Technologies de Calcul Distribué
Coefficient : 2

Objectifs de l'enseignement

Mettre l'accent sur l'apport de l'IA dans les nouvelles technologies à commencer par un bref historique sur les grilles de calculs pour mettre en avant leur intérêt dans la parallélisation et la distribution des calculs en IA. Pour ensuite, aborder les solutions IA pour répondre aux problématiques du Cloud, Fog et Edge computing, et finir par la concrétisation de l'IoT.

Connaissances préalables recommandées

Système / Réseau, Machine et Deep Learning, Traitement des données Massives

Contenu de la matière :

Chapitre I : Grid Computing

1. Principe
2. Infrastructure matérielle
3. Architecture logicielle
4. Grid de calcul pour l'intelligence artificielle
5. Applications :
   5.1. Cluster programming
   5.2. GPU/GPGPU programming

Chapitre II : Cloud Computing

1. Principe
2. Infrastructure matérielle
3. Architecture logicielle
4. Problématique du Cloud
5. Solutions IA pour le Cloud

Chapitre III : Fog Computing

1. Principe : L'informatique Geodistribuée
2. Infrastructure matérielle
3. Architecture logicielle
4. Problématique du Fog Computing
5. Solutions IA pour le Fog

Chapitre IV : Edge Computing

1. Principe : Les Réseaux programmables
2. Infrastructure matérielle
3. Architecture logiciel
4. Problématique du Edge Computing
5. Solutions IA pour le Edge

Chapitre V : IoT

1. Principe
2. Infrastructure matériel
3. Architecture logiciel
4. Solutions IA pour la concrétisation de l'IoT.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. You, S., Zhao, Y., Mandich, M., Cui, Y., Li, H., Xiao, H., ... & Zhang, Y. (2020, November). A review on artificial intelligence for grid stability assessment. In 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) (pp. 1-6). IEEE.
2. Belgaum, M. R., Alansari, Z., Musa, S., Alam, M. M., & Mazliham, M. S. (2021). Role of artificial intelligence in cloud computing, IoT and SDN: Reliability and scalability issues. International Journal of Electrical and Computer Engineering, 11(5), 4458.
3. Saeik, F., Avgeris, M., Spatharakis, D., Santi, N., Dechouniotis, D., Violos, J., ... & Papavassiliou, S. (2021). Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. *Computer Networks*, *195*, 108177.
4. Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), 854-864.
5. Buyya, R., & Srirama, S. N. (Eds.). (2019). Fog and edge computing: principles and paradigms. John Wiley & Sons.

Semestre 8
Unité d'enseignement : UET1
Matière : Projet pluridisciplinaire
Coefficient : 3

Objectifs de l'enseignement

Le but de cette matière est l'immersion des étudiants dans le milieu socio-économique en les plaçant dans des stages dans les entreprises. Le projet se déroule durant le second semestre de la quatrième année. Il consiste en la conception et la réalisation d'un petit projet informatique qui se déroule dans une entreprise.

Stage : un stage de 15 jours dans une entreprise pendant les vacances du printemps.

Déroulement du projet :

Le projet est décrit à travers un cahier des charges précis et peut porter sur des thèmes très variés. Il est proposé et encadré par un enseignant du département et il doit couvrir au moins deux disciplines.

Le groupe de projet est composé de 4 étudiants au maximum. Outre le contenu technique, qui consistera en l'application des connaissances acquises pour la mise en œuvre du cycle de développement d'un petit logiciel, l'accent sera mis sur l'acquisition et l'application des aspects organisationnels et relationnels entre les membres du groupe, de l'encadreur et de l'entreprise d'accueil en respectant les points suivants :

- ✓ Analyse et découpage du travail,
- ✓ Répartition des charges de travail entre les membres du groupe par l'encadreur.
- ✓ Circulation de l'information entre les membres du groupe,
- ✓ Mise en place d'un planning de travail,
- ✓ Exposés périodiques de l'avancement du projet,
- ✓ Délivrance des livrables fixés dans la fiche de projet,
- ✓ Rédaction d'un rapport de stage (entre 20 et 30 pages).
- ✓ Exposé du travail réalisé devant une commission d'examen.

Modalités d'évaluation du projet

L'évaluation du projet aura la forme d'une note sur vingt et repose sur les critères suivants :
Le groupe remet un rapport de stage et le logiciel accompagnés d'une lettre de présence dans l'entreprise d'accueil.
Une commission d'examen composée de l'encadreur, d'un enseignant du département et éventuellement d'un représentant de l'entreprise d'accueil examinera le dossier en présence du groupe d'étudiants.
La note finale est délivrée à chaque étudiant du groupe (note globale attribuée à l'équipe ou individuelle au cas où il est fait constat que le volume de travail fourni par les membres est inégal) selon le barème suivant :

- Le rapport de stage est noté sur 6 points
- Le logiciel est noté sur 6 points
- La présentation et les réponses aux questions sont notées sur 6 points.

(La note attribuée sur 18 est égale à la moyenne des notes octroyées par les membres de la commission d'examen).

- Une note de travail continu (sur 2 points) est donnée par l'encadreur. Cette note validera en quelque sorte l'assiduité des étudiants aux réunions périodiques et le respect des objectifs fixés.

Mode d'évaluation : Soutenance (100%)

Semestre 9
Unité d'enseignement : UEF1
Matière : Data Visualization
Coefficient : 4

Objectifs de l'enseignement

This course aims to equip students with the skills to effectively represent and communicate data insights through various graphical and interactive techniques. Objectives include understanding the principles of visual design, mastering tools and libraries for creating diverse visualizations, and applying these techniques to real-world data to enhance decision-making and storytelling.

Connaissances préalables recommandées

Students should have a solid understanding of statistics and data analysis fundamentals. They should also be proficient in programming, preferably with tools like Python (and libraries such as Matplotlib or Seaborn) or R, and have a grasp of visual design principles to create clear and informative graphics.

Contenu de la matière :

Chapter I : Introduction and Fundamentals

1. Introduction to Data Visualization
   - 1.1. Importance and goals of data visualization
   - 1.2. History and evolution of data visualization
   - 1.3. Key principles of effective data visualization
   - 1.4. Types of data and visualizations: categorical, numerical, temporal, spatial
   - 1.5. **PW**: Set up the development environment (Python, R, Tableau) and Explore different types of visualizations using an existing dataset
2. Data Visualization Tools and Libraries
   - 2.1. Overview of popular tools and libraries: Matplotlib, Seaborn, Plotly, D3.js, ggplot2, Tableau
   - 2.2. Comparison of strengths and weaknesses
   - 2.3. Installation and basic usage
   - 2.4. **PW**: Create basic plots using Matplotlib and Seaborn

Chapter II : Fundamental Visualization Techniques

1. Basic Plots and Charts
   - 1.1. Bar charts, line charts, scatter plots
   - 1.2. Histograms, box plots, and violin plots
   - 1.3. Pie charts, donut charts, and their limitations
   - 1.4. **PW**: Create and customize various basic plots using a dataset
2. Advanced Plotting Techniques
   - 2.1. Heatmaps and density plots
   - 2.2. Pair plots and correlation matrices
   - 2.3. Small multiples and faceting
   - 2.4. **PW**: Create advanced plots to explore relationships in data

Chapter III : Interactive Visualizations and Dashboards

1. Introduction to Interactive Visualizations
   - 1.1. Benefits of interactivity in visualizations
   - 1.2. Introduction to Plotly and Bokeh for Python
   - 1.3. Introduction to Shiny for R
   - 1.4. **PW**: Create interactive plots using Plotly or Bokeh

2. Building Dashboards
    2.1. Dashboard design principles
    2.2. Tools for dashboard creation: Tableau, Power BI, Dash (Plotly)
    2.3. Integrating multiple visualizations into a cohesive dashboard
    2.4. **PW**: Build a simple interactive dashboard using Tableau or Dash

Chapter IV : Storytelling with Data

1. Principles of Data Storytelling
    1.1. Elements of a good data story
    1.2. Structuring narratives with data
    1.3. Using visualizations to support storytelling
    1.4. **PW**: Develop a narrative around a dataset and create visualizations to support the story
2. Advanced Storytelling Techniques
    2.1. Use of annotations and highlights
    2.2. Sequencing and transitions in storytelling
    2.3. Case studies of effective data storytelling
    2.4. **PW**: Create a data story presentation using a dataset

Chapter V : Specialized Visualization Techniques

1. Geospatial Data Visualization
    1.1. Basics of geospatial data
    1.2. Tools and libraries: Folium, GeoPandas, Mapbox, Leaflet
    1.3. Creating maps: Choropleth maps, bubble maps, heatmaps
    1.4. **PW**: Visualize geospatial data using Folium or GeoPandas
2. Network Visualization
    2.1. Basics of network data
    2.2. Tools and libraries: NetworkX, Gephi, Cytoscape
    2.3. Visualizing networks: node-link diagrams, adjacency matrices
    2.4. **PW**: Visualize network data using NetworkX or Gephi

Chapter VI : Visual Analytics and Big Data

1. Visual Analytics
    1.1. Overview of visual analytics
    1.2. Integrating visualizations with analytical processes
    1.3. Case studies in visual analytics
    1.4. **PW**: Apply visual analytics techniques to a complex dataset
2. Big Data Visualization
    2.1. Challenges of visualizing big data
    2.2. Tools and techniques for big data visualization: Apache Zeppelin, Kibana, D3.js
    2.3. Case studies and applications
    2.4. **PW**: Visualize a large dataset using appropriate big data tools

Chapter VII : Advanced Topics and Trends

1. Real-time Data Visualization
    1.1. Introduction to real-time data visualization
    1.2. Tools and frameworks: Kafka, D3.js, Plotly
    1.3. Applications and examples
    1.4. **PW**: Create a real-time dashboard using a streaming data source
2. Augmented and Virtual Reality for Data Visualization
    2.1. Overview of AR and VR in data visualization
    2.2. Tools and technologies: Unity, VR.js, Three.js
    2.3. Applications and case studies
    2.4. **PW**: Develop a simple AR/VR data visualization prototype

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. "Data Visualization: A Practical Introduction", Kieran Healy, Princeton University Press, 2018
2. "Interactive Data Visualization for the Web: An Introduction to Designing with D3", Scott Murray, O'Reilly Media, 2021 (3rd Edition)
3. "Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures", Claudiu Fatu, O'Reilly Media, 2021
4. "Storytelling with Data: A Data Visualization Guide for Business Professionals", Cole Nussbaumer Knaflic, Wiley, 2019 (2nd Edition)
5. "Data Visualization with Python and JavaScript: Scrape, Clean, Analyze, and Visualize Data", Karthik Ramasamy, Packt Publishing, 2021

Semestre 9
Unité d'enseignement : UEF1
Matière : Vision par ordinateur
Coefficient : 3

Objectifs de l'enseignement

L'objectif du cours est de donner à l'étudiant l'ensemble des techniques d'analyse d'image pour reconstruire la scène 3D et de comprendre le contenu de la scène statique ou dynamique.

Connaissances préalables recommandées

Traitement d'image, Algorithmique.

Contenu de la matière :

Chapitre I : Modèle de caméra

1. Le modèle de caméra Pin-hole (trou d'épingle)
2. Calibration de caméra
3. Calibration d'une tête stéréoscopique
4. Géométrie épipolaire
5. Estimation de la pose de caméra à partir de plans et marqueurs
6. Application : placement d'un objet virtuel sur l'image.

Chapitre II : La reconstruction 3D

1. Introduction
2. Techniques de reconstruction 3D
3. Reconstruction 3D par triangulation
   3.1. La stéréo correspondance
   3.2. La rectification
4. Reconstruction Photométrique
   4.1. Modélisation mathématique de la reconstruction photométrique.
   4.2. Génération du champ de normales avec la SVD (Matrice pseudo inverse)

Chapitre III : Reconnaissance : techniques et applications

1. Détection d'objet à reconnaître (faces, piétons)
2. Recognition de faces
   2.1. Modèles actifs d'apparence
   2.2. Modèles 3D de formes
3. Reconnaissance de catégories d'objets
   3.1. Reconnaissance basée sur les modèles basés sur les parties
   3.2. 3.3.2. Reconnaissance basée sur la segmentation
   3.3. 3.3.3. Reconnaissance par apprentissage automatique

Chapitre IV : Quelques applications

1. La réalité virtuelle
   1.1. Introduction
   1.2. Les concepts de base
   1.3. Modélisation 3D L'interaction
   1.4. Détection et réponse à la collision
   1.5. Les périphériques de la RV
   1.6. Vision stéréoscopique
2. La réalité augmentée
   2.1. Définitions, classifications et périphériques
   2.2. Estimation de la pose

2.3. Détection des points d'intérêt et extraction de primitives
2.4. Algorithmes de recalage
3. Réalité mixte

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Richard Szeliski, Computer Vision: Algorithms and Applications Springer, 2010
2. Jan Erik Solem, Programming Computer Vision with Python, 2012, Creative Commons
3. Simon J.D. Prince, Computer vision: models, learning and inference Cambridge University Press 2012.
4. Forsyth Ponce, Computer vision : A modern approach,
5. J.R. Parker, Algorithms for Image Processing and Computer Vision, Wiley Publishing, Inc. 2011.
6. Mubarek Shah Fundamentals of Computer Vision, 1997.
7. Linda Shapiro, George C. Stockman, Computer Vision, Ed. Prentice Hall. 2001

Semestre 9
Unité d'enseignement : UED1
Matière : Architectures et technologies blockchain
Coefficient : 2

Objectifs de l'enseignement

Ce module vise principalement à fournir aux étudiants une compréhension des concepts, des technologies et des applications de la blockchain dans le but d'explorer ses implications à long terme pour les entreprises et sa relation avec d'autres technologies émergentes telle que l'IA. L'objectif est donc de permettre aux étudiants de proposer des solutions basées sur la blockchain pour des problèmes réels.

Connaissances préalables recommandées

Programmation orienté ob**jet,** Systèmes distribués**,** Cryptographie

Contenu de la matière :

Chapitre I : Introduction à la blockchain

1. Définition et concepts fondamentaux de la blockchain (Définitions, Caractéristiques, Composants clés, les types de réseaux Blockchain (Blockchains publiques, privées, permissionnées, hybrides, etc.).
2. Utilisations et cas d'usage réels (finance, supply chain, gouvernance, etc.)
3. Smart contracts: fonctionnement, langages de programmation (Solidity, Vyper, etc.)
4. Dapps (applications décentralisées) : architecture, développement, défis
5. DAOs (organisations décentralisées autonomes) : gouvernance, prise de décision collective

Chapitre II : Structure de données pour la blockchain

1. Blocs : en-tête, transactions, merkle-tree
2. Chaîne de blocs : structure de données, propriétés de sécurité
3. Transactions : structure, signatures numériques, UTXO vs comptes
4. Adresses : génération, gestion des clés publiques/privées

Chapitre III : fonctionnement de la blockchain

1. Emission d'une transaction
2. Validation de la transaction
3. Constitution d'un bloc candidat
4. Obtention d'un consensus décentralisé
5. Validation du nouveau bloc

Chapitre IV : Protocoles de consensus

1. Preuve de travail (PoW) : fonctionnement, minage
2. Preuve d'enjeu (PoS) : différentes variantes (Casper, Ouroboros, etc.)
3. Autres protocoles : preuve d'autorité, preuve d'espace, etc.
4. Tolérance aux pannes byzantines et propriétés recherchées
5. Attaques contre le consensus : étude de cas et parades

Chapitre V : Distributed Ledger Technology (DLT)

1. Définition
2. Différence entre blockchain et DLT.
3. Avantages de la technologie DLT.
4. Domaines d'application : finance, identités, IoT, etc.
5. Une comparaison des avantages/inconvénients par rapport aux blockchains

Chapitre VI : Plateformes de blockchain

1. Blockchains publiques (Bitcoin : La première blockchain, focalisée sur la crypto-monnaie, Ethereum : Plateforme de contrats intelligents et dapps, protocole proof-of-work, Solana, Avalanche : Nouvelles blockchains hautes performances, Cardano, Polkadot : Prometteuses blockchains de 3e génération, Autres : Litecoin, Bitcoin Cash, etc.)
2. Blockchains d'entreprises permissionnées : (Hyperledger Fabric (Linux Foundation) : Modulaire, adaptée aux consortiums, R3 Corda : Optimisée pour les applications financières, Quorum (JP Morgan) : Fourche privée d'Ethereum avec confidentialité
3. Distributed Ledgers non-blockchain (Hashgraph (Hedera) : Basée sur les graphes hashés gossiping, IOTA Tangle : DLT pour l'Internet des objets sans blocs, Holochain: Framework d'applications distribués pair-à-pair
4. Plateformes hybrides/interopérabilité (Cosmos : Écosystème de blockchains interconnectées, Polkadot : Protocole de transfert de données cross-chain, Ondulation: Passerelle de liquidité entre blockchains)

Chapitre VII : processus de conception des application blochckain,

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. "Building Decentralized Blockchain Applications: Learn How to Use Blockchain as a Backbone for Distributed Applications" par Shahid Shaikh
2. "Blockchain Basics: A Non-Technical Introduction in 25 Steps" par Daniel Drescher
3. "Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World" par Don Tapscott et Alex Tapscott

Semestre 9
Unité d'enseignement : UEF2
Matière : Méthodes bio-inspirées
Coefficient : 3

Objectifs de l'enseignement

L'objectif de l'informatique bio-inspirée est de présenter un panorama des modèles informatiques inspirés de la structure des systèmes vivants (biologiques), d'expliquer les principes généraux, les modes de fonctionnement et les domaines d'application en intelligence artificielle.

Connaissances préalables recommandées

Algorithmique et complexité, machine learning.

Contenu de la matière :

Chapitre I : Introduction à l'informatique bio-inspirée
Définitions, principes généraux de l'informatique bio-inspirée, domaines d'application, historique.
Chapitre II : Panorama des différents modèles
Diversité des sources d'inspiration, diversités des modèles, exemples de mécanismes émergents
Chapitre III : Les approches évolutionnistes
Algorithmes génétiques, programmation génétique
Chapitre IV : Les approches cellulaires
Réseaux de neurones et réseaux immunitaires
Chapitre V Les approches populationnelles
Systèmes à base de fourmis, l'optimisation par essaims particulaires

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. J, M Alliot.t, Schiex (1993), Intelligence Artificielle et Informatique Théorique, Editions Cépaduès.
2. Jean-Michel Renders (1995), Algorithmes génétiques et réseaux de neurones, Editeur(s) : Hermès - Lavoisier.
3. Back .T (1996) .Evolutionary algorithms in theory and practice. Oxford university press.
4. Dessales J-L (1996) .L'ordinateur génétique. Paris. Hermès.
5. Goldberg D.E (1994). Algorithmes génétiques, exploration, optimisation et apprentissage automatique. Paris, Addison-Wesley France.
6. G. Dreyfus , M. Samuelides , J.-M. Martinez , M. B. Gordon , F. Badran , S. Thiria , L. Hérault (2004), Réseaux de neurones - Méthodologies et applications, 2° Edition, Editions Eyrolles Paris.
7. D. Dasgupta (1999), Artificial Immune Systems and Their Applications, Springer-Verlag, Inc. Berlin.

Semestre 9
Unité d'enseignement : UEF2
Matière : Generative AI
Coefficient : 4

Objectifs de l'enseignement

By the end of this course, learners will gain a solid foundation in generative AI, including its core principles and technological frameworks, develop practical skills through hands-on projects and experiential learning to confidently create and manipulate AI-generated content, and achieve a functional understanding of how generative AI systems work.

Connaissances préalables recommandées

Machine Learning and Deep learning

Contenu de la matière :

Chapter I: Generative AI and Model Pre-Training

1. Introduction
2. An overview of generative AI, its importance, and its applications.
3. Generative AI & Large Language Models (LLMs)
4. Transformers Architecture
5. Generating Text with Transformers
6. Prompt Engineering
7. Pre-Training LLMs

Chapter II : Fine-Tuning and Evaluation of Large Language Models

1. Fine-Tuning on a Single Task
2. Multi-Task Instruction Fine-Tuning
3. Model Evaluation
4. Benchmarks

Chapter III : Reinforcement Learning and LLM-Powered Applications

1. Aligning Models with Human Values
2. Reinforcement Learning from Human Feedback
3. Model Optimizations for Deployment
4. Generative AI Project Lifecycle Cheat Sheet deployment.
5. Using LLMs in Applications

Chapter IV : Generative AI in Practice

1. Coding and Programming
2. Healthcare Transformation
3. Personalized Learning
4. Video Game Design and Testing
5. Image and video generation:
6. Creating realistic images from text descriptions, generating variations on existing images, or even producing videos.

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

- Jay Alammar, (2023), Hands-On Large Language Models, O'Reilly Media, Inc.
- David Foster, (2023), Generative Deep Learning, 2nd Edition, O'Reilly Media, Inc.
- Zhihan Lyu, (2024), Applications of Generative AI, Springer
- Bernard Marr, (2024), Generative AI in Practice, Wiley.

Semestre 9
Unité d'enseignement : UED1
Matière : Recherche d'information
Coefficient : 2

Objectifs de l'enseignement

Le cours couvre de nombreux sujets, notamment l'indexation, la pondération et la représentation de l'information, les modèles de la RI, l'évaluation des performances des Systèmes de Recherche d'Information (SRIs) et l'intégration des techniques de traitement automatique du langage naturel (TALN) et les techniques d'apprentissage automatique afin d'améliorer l'efficacité et l'efficience des SRIs .

Connaissances préalables recommandées

Les étudiants doivent avoir des connaissances de base en mathématiques, notamment en statistiques et en algèbre, ainsi qu'une compréhension des principes fondamentaux des bases de données. Ils doivent également être à l'aise avec les concepts de recherche et de traitement de données, et posséder des compétences en programmation pour utiliser des outils de recherche et d'analyse.

Contenu de la matière :

Chapitre I : Introduction à la Recherche d'Information (RI)

1. Définition du domaine de la RI
2. Le besoin en information et requête
3. Notion de pertinence
4. Notion de similarité
5. Approche générale de la RI
6. Processus de la RI

Chapitre II : Indexation, pondération et représentation de l'information

1. Introduction
2. Types d'indexations
3. Notion du vocabulaire
4. Le processus d'indexations
5. Normalisation de l'information
6. Pondération statistique des termes
7. Fichier inverse

Chapitre III : Les modèles de base de la RI

1. Le modèle booléen
2. Le modèle vectoriel
3. Le modèle booléen basé sur les ensembles flous
4. Le modèle P-norme
5. Le modèle LSI

Chapitre IV : Evaluation des performances des systèmes de RI

1. Objectif
2. Démarche d'évaluation
3. Environnement d'évaluation
4. Les mesures d'évaluation
5. Le Rappel et la Précision
6. Le Rappel et la Précision à chaque document pertinent
7. Courbe rappel/précision
8. Interpolation de la courbe Rappel/Précision

9. Précision moyenne pour une requête
10. Moyenne de Précision sur plusieurs requêtes
11. Mesures focalisées sur le "top" de la liste

Chapitre V : La recherche d'information avancée et intelligente

1. La recherche d'information personnalisée
2. L'apprentissage automatique du langage Naturel (NLP) pour la RI
3. Le machine Learning pour la RI

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :

1. Schütze, H., Manning, C. D., & Raghavan, P. (2008). Introduction to information retrieval. Cambridge: Cambridge University Press.
2. Croft, W. B., Metzler, D., & Strohman, T. (2010). Search engines: Information retrieval in practice (Vol. 520, pp. 131-141). Reading: Addison-Wesley.
3. Buttcher, S., Clarke, C. L., & Cormack, G. V. (2016). Information retrieval: Implementing and evaluating search engines. MIT Press.
4. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern information retrieval (Vol. 463). New York: ACM press.
5. Amini, M. R., & Gaussier, E. (2013). Recherche d'information: Applications, modèles et algorithmes-Fouille de données, décisionnel et big data. Editions Eyrolles.

Semestre 9
Unité d'enseignement : UET1
Matière : Séminaire & workshops
Coefficient : 2

Objectifs de l'enseignement

Proposer des thèmes d'actualité dans le domaine des applications de l'IA

Connaissances préalables recommandées

Contenu de la matière :

Mode d'évaluation : Contrôle continu (40%), Examen (60%)

Références bibliographiques *(Livres et polycopiés, sites internet, etc)* :