ASSESSMENT FOR IDENTIFYING SKILLS GAPS IN HIGH PERFORMANCE COMPUTING RELATED HIGHER EDUCATION PROGRAMS BY USING NLP

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜLŞAH KARGIN ASLIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

APRIL 2023

Approval of the thesis:

ASSESSMENT FOR IDENTIFYING SKILLS GAPS IN HIGH PERFORMANCE COMPUTING RELATED HIGHER EDUCATION PROGRAMS BY USING NLP

Submitted by Gülşah KARGIN ASLIM in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, Graduate School of Informatics _____

Prof. Dr. Altan Koçyiğit
Head of Department, Information Systems _____

Prof. Dr. Banu Günel Kılıç
Supervisor, Information Systems Dept., METU _____

Examining Committee Members:

Assoc. Prof. Dr. Erhan Eren
Information Systems Dept., METU _____

Prof. Dr. Banu Günel Kılıç
Information Systems Dept., METU _____

Assoc. Prof. Dr. Çiğdem Turhan
Software Engineering Dept., Atılım University _____

Date: 11.04.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Gülşah KARGIN ASLIM

Signature :

ABSTRACT

ASSESSMENT FOR IDENTIFYING SKILLS GAPS IN HIGH PERFORMANCE COMPUTING RELATED HIGHER EDUCATION PROGRAMS BY USING NLP

KARGIN ASLIM, Gülşah
MSc., Department of Information Systems
Supervisor: Prof. Dr. Banu GÜNEL KILIÇ

April 2023, 150 pages

People require a greater and broader set of abilities to work, communicate, access information, products, and services, as well as to participate in social and civic activities in the world of today and future. In order to assist people in acquiring and updating skills throughout their lives as they transition between various types and degrees of education, between education and employment, and across national boundaries, a good understanding and valuation of the skills available are vital. The supply of skills and the needs of the labor market can better match each other in this way.

In particular, the search for the most appropriate and necessary hard/soft skills and competencies to keep up with the new developing technologies and competition is very important. For this reason, studies on the definition and classification of skills requirements are encountered today. In this context, one of the most important examples is the European Skills, Competences, Qualifications and Occupations framework, which has a multilingual structure. The objectives defined in ESCO are to describe the learning outcomes of the qualifications by using multilingual terminology, the adaptation of the programs according to feedback from the labor market, and work with academia closely. Although the importance of skills requirements is recognized in the industry, skills misalignment still exists, which poses an obstacle for people who want to specialize in a field after graduating from university to achieve operational excellence.

Therefore, this thesis aims to reveal skill mismatches in a field of specialization and to quantitatively show how well the selected graduate programs meet the requirements

of these occupations by developing a methodology that uses Natural Language Processing (NLP) and shows up what skills are required for some of the occupations that are most relevant in the chosen field of specialization. The motivation of this thesis work is based on developing a methodology adaptable to all education levels, academic programs, and curriculums and can be applied to any domain for comparing the existing curriculum of the defined academic program with qualifications frameworks to produce measurable results for academia and business. In order to demonstrate the developed methodology, high performance computing has been selected as an example, skills have been identified and two graduate level programs have been evaluated.

Keywords: Skills Gap, High-Performance Computing (HPC), Higher Education, ESCO Database, Curriculum, Natural Language Processing (NLP)

ÖZ

YÜKSEK BAŞARIMLI HESAPLAMA İLE İLGİLİ YÜKSEKÖĞRETİM PROGRAMLARINDAKİ BECERİ BOŞLUKLARINI NLP KULLANARAK BELİRLEMEK İÇİN DEĞERLENDİRME

Kargın Aslım, Gülşah
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Banu Günel Kılıç

Nisan 2023, 150 sayfa

İnsanlar, bugünün ve geleceğin dünyasında çalışmak, iletişim kurmak, bilgilere, ürünlere ve hizmetlere erişmek ve ayrıca sosyal faaliyetlerine katılmak için daha büyük ve daha geniş yeteneklere ihtiyaç duyar. İnsanlara yaşamları boyunca çeşitli eğitim türleri ve dereceleri arasında, eğitim ve istihdam arasında ve ulusal sınırların ötesinde geçiş yaparken becerileri edinmelerinde ve güncellemelerinde yardımcı olmak için, mevcut becerilerin iyi bir şekilde anlaşılması ve değerlendirilmesi hayati önem taşımaktadır. Beceri/yetenek arzı ve işgücü piyasasının ihtiyaçları bu şekilde birbiriyle daha iyi eşleşebilir.

Özellikle yeni gelişen teknolojilere ve rekabete ayak uydurmak için en uygun ve gerekli teknik/yumuşak beceri ve yetkinliklerin araştırılması çok önemlidir. Bu nedenle günümüzde beceri gereksinimlerinin tanımlanması ve sınıflandırılmasına yönelik çalışmalara rastlanmaktadır. Bu bağlamda en önemli örneklerden biri çok dilli bir yapıya sahip olan Avrupa Beceriler, Yetkinlikler, Yeterlilikler ve Meslekler (ESCO) çerçevesidir. ESCO'da tanımlanan hedefler, çok dilli terminoloji kullanarak yeterliliklerin öğrenme çıktılarını tanımlamak, programların işgücü piyasasından gelen geri bildirimlere göre uyarlanması ve akademi ile yakın çalışmak olarak belirlenmiştir. Sektörde beceri gereksinimlerinin önemi kabul edilse de beceri uyumsuzluğu hala mevcuttur, bu da üniversiteden mezun olduktan sonra bir alanda uzmanlaşmak isteyen kişilerin operasyonel mükemmelliğe ulaşmasının önünde engel teşkil etmektedir.

Bu nedenle, bu tez, bir uzmanlık alanındaki beceri uyumsuzluklarını ortaya çıkarmayı ve en alakalı birkaç meslek için hangi becerilerin gerekli olduğunu gösteren Doğal Dil

İşleme (NLP) kullanılarak hazırlanan bir metodoloji geliştirerek seçilen lisansüstü programların bu mesleklerin gereksinimlerini ne kadar iyi karşıladığını nicel olarak göstermeyi amaçlamaktadır. Bu tez çalışmasının motivasyonu, tüm eğitim seviyelerine, akademik programlara ve müfredatlara uyarlanabilir bir metodoloji geliştirmeye dayanmaktadır. Bu çalışma akademi ve iş dünyası için ölçülebilir sonuçlar üretmek amacı ile akademik programların mevcut müfredatlarını yeterlilik çerçeveleri ile karşılaştırmak için herhangi bir alanda uygulanabilir. Geliştirilen metodolojiyi göstermek için yüksek performanslı bilgi işlem örnek olarak seçilmiş, beceriler belirlenmiş ve iki lisansüstü program değerlendirilmiştir.

Anahtar Kelimeler: Beceri Boşluğu, Yüksek Başarımlı Hesaplama, Yüksek Öğretim, ESCO Veri Tabanı, Müfredat, Doğal Dil İşleme (NLP)

To My Little One

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Prof. Dr. Banu Günel Kılıç, who supported me throughout this process, encouraged me and motivated me in a different way every time I gave up.

I am also grateful my only son, Kaan Berk, who asked tirelessly “Mother, when this homework will be finished?”, but patiently waited for it to be completed, and gave me time and sat quietly next to me during my thesis work.

This thesis is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 101051997, project EUMaster4HPC (European Master for High Performance Computing).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
CS	Computer Science
DevOps	Development and Operations
e-CF	European e-Competence Framework
ECVET/ECTS	European Credit System for Vocational Education & Training)
Eng	Engineer
EntreComp	European Entrepreneurship Competence Framework
EQAVET	European Quality Assurance Reference Framework for Vocational Education and Training
EQF	European Qualification Framework
ESCO	European Skills, Competences, Qualifications and Occupations
EU	European Union
EUMaster4HPC	European Master for High-Performance Computing
HPC	High-Performance Computing
ICT	Information Communication Technology
ICT BOK	Information Communication Technology Body of Knowledge
ISCO	International Standard Classification of Occupations
LOD	Linked Open Data
LSI	Latent Semantic Index
METU	Middle East Technical University
MSc	Master of Science
NLP	Natural Language Processing
NER	Named Entity Recognition
OECD	Organization for Economic Co-operation and Development
O*NET	Occupational Information Network
QS	Quacquarelli Symonds
RO	Research Objective
STS	Semantic Textual Similarity
SYS	System
SW	Software
THE	Times Higher Education

CHAPTER 1

1 INTRODUCTION

People's abilities, competencies, attitudes, and personality traits make up the key building blocks on which businesses build their strategy for gaining a competitive edge and achieving superior business results. To stay ahead of the fierce competition, it's important to look for the best hard/soft skills, competencies, and personal criteria, particularly in the disciplines of Computer Science and Information Communication Technology. For this reason, numerous industries have recently increased their efforts to describe, specify, and classify the skills needs.

The European Skills, Competences, Qualifications and Occupations database (ESCO), a multilingual classification of occupations across the EU labor market that goes far beyond the Computer Science (CS) and Information, Communication Technology (ICT) areas, is the most noteworthy example in this regard. The ESCO database intends to improve the matching between individuals and industry professions by offering a framework to link the essential players in the job market. There are also other European tools and principles which are European Qualifications Framework (EQF), ECVET/ECTS (European Credit System for Vocational Education & Training), EuroPass, EQAVET (European Quality Assurance Reference Framework for Vocational Education and Training) that has all same shared purpose to improve qualifications' transparency and comparability, enable mobility, assist persons in validating and capturing their knowledge, skills, and competences.

Supercomputing, often known as High-Performance Computing (HPC), is a category of computing where speed, or alternatively, the amount of time it takes to get a solution is the key characteristic. The development of high-end computing has spanned several centuries of enumeration and recording, nearly 500 years of mechanical and electronic computation, and the current era of digital electronic computing. In the fields of engineering, health, science, economics, and defense, HPC has made advancements

possible (Thomas Sterling, 2018). Due to the acceleration of the developments in the field of high-performance computing (HPC) and the increasing need for this field, it is very important to train human resources that can work in this field. In recent years, the volume of data has increased quickly, and numerous new applications are utilizing HPC's capabilities. When compared to traditional computing, this power enables the achievement of outcomes in less time and at a cheaper cost by performing compute-intensive processes across shared resources. HPC hardware and software have also improved in availability and usage. Scientists, engineers, and researchers employ HPC for a wide range of applications, including medicine, weather forecasting, physics, quantum mechanics, and other academic and industrial sectors.

So, in order to fill the skills gaps created by these new business areas, it is necessary to train appropriate human resources for this area. It's crucial to be able to design university curricula in a way that technology dictates the direction to go to close the skills gap in order to accomplish this goal, keep up with the competition, and keep up with new developing technologies.

For the HPC area, the definition of skills, competencies, and qualifications have been explored recently. European e-Competence Framework (e-CF), ESCO and European ICT BOK applications, which use a common terminology to reveal new standards and competencies specific to the rapidly developing field of HPC, are taken as an important reference to provide framework for competences in Europe. In particular, the search for the most appropriate and necessary hard/soft skills and competencies in the field of HPC to keep up with the new developing technologies and competition is very important. For this reason, studies on the definition and classification of skills requirements are encountered today. ESCO framework, one of the most important examples in this context, which has a multilingual structure was analyzed in this thesis. The objectives defined in ESCO are to describe the learning outcomes of the qualifications by using multilingual terminology, the adaptation of the programs according to feedback from the labor market, and work with the academia closely (European Commission ESCO, 2022).

In the light of the ESCO database criteria and industry requirements, the purpose of this research is to assess the skills mismatch of curricula of MSc programs on HPC for some of the most pertinent jobs in the HPC domains. This research hence focuses on four different profiles related with “Data Science, Computer Architecture, Parallel Programming, and DevOps Engineer” skills which are believed to be essential in HPC. The approach of this contribution specifically looks at the major tasks for the aforementioned jobs and runs a gap analysis based on Natural Language Processing (NLP) methods in the MSc degree’s curricula for the abilities needed for each. The objective of using NLP in this thesis is defined as, to reveal the extent to which level the skills defined for an occupation in ESCO and the HPC graduate courses syllabuses are semantically similar and at what level they cover each other.

Although the importance of skills requirements is recognized in the industry, skills misalignment still exists, which poses an obstacle for people who want to specialize in a field after graduating from university to achieve operational excellence. This study also provides guidance for developing the ESCO database and aligning it with industry requirements, as well as providing useful information to the academy about required courses in MSc programs on HPC.

1.1 Background of the Study

Even though many universities’ computer related courses cover fundamental computer science concepts and programming languages, current educational initiatives sometimes fall short of what is needed to prepare students for the fast-evolving HPC technology ecosystem. So, the EUMaster4HPC project was started in 2022 by the major players in HPC in Europe to create a pilot Master's program curriculum that will support Europe’s Digital Transformations strategy and ensure the sustainability of high-priority business sectors, in a modular format to enable complete or partial integration of the modules into other Master's programs, whether they are new or existing. In this thesis, High-Performance Computing has been chosen as the field due to developing technology and increasing supercomputing needs to develop an

assessment methodology, and due to considering HPC MSc programs are still new and there are areas to be researched and developed in the curriculums.

The objectives of this public project are described as: *"The **HPC European Consortium Leading Education Activities** (EUMaster4HPC) aims to develop a new and innovative European Master Programme focusing on high-performance solutions to address these issues. The master Programme aims at catalyzing various aspects of the HPC ecosystem and its applications into different scientific and industrial domains."* (EUMaster4HPC, 2022). This program began education in the fall of 2022. The main objective of the consortium is to link HPC domains/activities in industry and academia, by preparing HPC based MSc program to strengthen activity between academia, industry, and research centers. After the completion of the pilot Programme, which will create a basis for future HPC educational programs, the lessons learned from this program will speed up the development of HPC in academia and industry.

Awarding universities will offer courses about HPC in 2 cohorts. For the 1st cohort, the courses will be chosen from the host university's Masters' Program. Awarding universities in this consortium are reported as Figure 1 and contributing partners who support the EUMaster4HPC program, including **METU (Middle East Technical University)**, are shown in Figure 2.

Figure 1: Awarding universities in EUMaster4HPC

Figure 2: Contributing partners of EUMaster4HPC

The project's goals include fostering a close relationship between academia and industry and educating the next generation of HPC professionals in the use of supercomputing technology for the European digital revolution. To achieve this goal, the master will create curricula with a modular structure in order to enable the full or partial integration of modules into current MSc's programs in fundamental computer science and programming languages. This is made possible by cutting-edge teaching paradigms.

There is another ongoing process across Europe called Bologna Process. The Bologna Process aims to increase coherence among European higher education systems throughout Europe. Building the trust necessary for effective cross-border academic collaboration, learning mobility, and the mutual recognition of study abroad experiences and credentials are the priorities of the Bologna reform. One of the main goals of the Bologna Process is to improve the standard and relevance of education and in addition to its long-standing commitments, which demand ongoing effort, it addresses emerging themes including core values, learning, and teaching. The 49 participating countries, however, have varied degrees of reform implementation. In other words, although appearing to be a structural reform, the Bologna process' reform is focused on curricular modifications, beginning with a reevaluation of the finalities

and continuing with adjustments at the practice level. The academics throughout the majority of Europe started to support the concept of completely redesigning the programs rather than only giving the new arrangements. Naturally, this necessitated determining what knowledge students needed to possess in terms of both content and inquiry-based and other teaching strategies. This is exactly the result that is tried to be revealed with the methodology developed in this thesis, and the question of whether the curricula of the universities are really designed to give the necessary skills for related occupations is asked in another way.

This thesis aims to reveal skill mismatches in the field of HPC and to quantitatively show how well the chosen universities' HPC Master's programs meet the requirements of these occupations by developing a methodology that shows up what skills are required for "Data Science", "Computer Architecture", "Parallel Programming", and "DevOps Engineer" related profiles that are most relevant in the chosen field of specialization. It is worth observing to compare the compatibility of the courses of the new/existing MSc program with the ESCO Database by using NLP techniques.

1.2 Significance of the Study

In this section, how the developed methodology will contribute to a better understanding of the broader research area and the specific research objectives are explained.

The ESCO categorization defines and groups skills, competences, qualifications, and occupations that are pertinent to the EU labor market as well as to education and training. The connections between the different concepts are demonstrated methodically in ESCO. It is stated that the ESCO can be utilized in a variety of business cases to offer various services;

- Enhancing communication between education and the work
- Facilitating mobility
- Assisting training and education in the transition to learning outcomes

While there are millions of unemployed persons in Europe, some areas and economic sectors now face skills gaps. Employers are having a harder time finding qualified applicants for open positions. Especially, for HPC, which is a specialized field, it is very important to find professionals with more specific skills. Today, HPC was started to be employed to tackle challenging issues in business, engineering, and science. A few academic institutions also employed HPC technology. For complicated applications, several government institutions, especially the military, rely on HPC. Businesses of all sizes will probably be interested in HPC as the demand for processing power and speed for practical applications increases, especially for transaction processing and data warehouses. Numerous other sectors of the economy also employ HPC systems, including but not restricted to automotive, health, aerospace, military, production, and finance.

The labor market's demands are causing educational systems to struggle. For instance, in order to keep up with the changes in the labor market, universities are required to quickly update their curriculum to include new competencies. Traditional educational institutions, however, require time to successfully develop and execute new or updated curriculum. Due to this, there is a discrepancy in the supply and demand of new competences. With the increasing need for HPC and the widespread establishment and use of HPC centers, the need for professionals to work in this field is increasing day by day. In order to close this skills gap that will arise in the sector, there is a study conducted in Europe to update the computer-science related course curricula of universities or to open HPC-specific graduate programs. With this study, EUMaster4HPC tries to determine the curricula of the newly opened HPC graduate programs in order to train suitable candidates to meet the needs of the sector. With the pilot project, curricula were determined, HPC programs were opened, and the training is in progress. In order to measure the results of this project, it is expected that the education will start and the graduates will be provided for the sector.

Meanwhile, the methodology described in this thesis is proposed to have beneficial results for both the academy, ESCO, and industry to compare HPC related MSc syllabuses with ESCO, whose purpose is to define the learning outcomes of the

qualifications, adapt the programs according to the feedback from the labor market, and work closely with the academy. The results of this comparison, when evaluated together with the work done in the EUMaster4HPC project, will shed light on the development and updating of the ESCO database, making recommendations to the academy and telling how ready the industry is for the HPC field.

The methodology created for this thesis is not constrained to a single academic field, one particular HPC position, or even simply one stage. In contrast to most other research, it was created to identify the most demanded skills in the HPC sector and hope to be able to offer suggestions to academics, business, and ESCO in that regard. According to this viewpoint, this technique not only investigates HPC activities and fields, but it also investigates how educational background relates to these practices and fields. As a result, it is anticipated that the results of this thesis may be used to inform decision-making in defining syllabuses and keeping ESCO database up-to-date in line with sectoral needs. The need for increasing soft and digital skills in HPC courses is also highlighted in this thesis.

1.3 Research Aims and Objectives

This thesis relies on Natural Language Processing techniques and suggest a new assessment methodology that can analyze textual data from the labor market and compare it with textual data from the universities, utilizing a reliable source for competences, the European Skills, Competences, Qualifications, and Occupations (ESCO) database created by the European Commission, developed with keywords from the EUMaster4HPC public deliverables in the field of HPC. This strategy enables to measure the academia's ability to keep up with market change quantitatively and objectively. The method has been used in the sphere of HPC, which is where this kind of instrument is most needed. So, the main goal of this thesis is to evaluate the competencies of existing/new graduate programs in the field of HPC, by the following specific objectives:

- RO1: Recommend a methodology to determine skills gaps.

- RO2: Evaluate whether ESCO is up-to-date on the basis of occupation and skill according to market requirements for HPC field.
- RO3: Identify the skills gaps and coverage of the industry requirements in the MSc education.
- RO4: By using NLP methods, compare the ESCO skills and university curriculums.
- RO5: Visualize the evaluation results.
- RO6: Compare different MSc programs' curriculums.
- RO7: Provide individualized recommendations for improvement of syllabuses.

The detailed information about research problem and ROs are given in Section 3.1.

1.4 Outline of the Thesis

Chapter 2 "Literature Review" presents the previous work done on skills gap analysis and ESCO as well as the NLP method used to reveal the semantical similarity of analysis data with each other.
Chapter 3 introduces the overview of the methodology developed.
Chapter 4 describes the methodology to compare the ESCO requirements with the HPC MSc curricula by applying NLP methodology and shows its applications in details.
Chapter 5 presents the results and discussions.
Chapter 6 concludes the thesis.

CHAPTER 2

2 LITERATURE REVIEW

As the business world and ways of doing business are changing rapidly day by day, to contribute value to businesses so must the curriculum of ICT and other computing disciplines. This kind of adaptation in the curriculum is necessary so that new graduates can add value to their work. Since computer related sciences are shaped through technological developments, related computing education should change on the basis of lifelong learning Industry-Academia collaboration.

In this thesis, the subjects to be reviewed were grouped as follows;

1. The skills gaps issue
2. European Skills, Competences, Qualifications and Occupations Classification (ESCO)
    a. The structure of ESCO database
    b. The other frameworks for mapping and consolidating models for the skills, competencies, and knowledge
    c. Comparison of ESCO with other qualifications frameworks
    d. Maintaining and improving ESCO
    e. Understanding the required skills/competencies and the gaps between academia and industry expectations
    f. The skills gaps analysis through ESCO
3. Natural Language Processing (NLP) Method

2.1 The Skills Gaps Issue

The effects of the misalignment between industry and academia are examined from many perspectives. When education cannot meet the needs of the industry, there arises

severe implications such as the increase in unemployment rates, the inability of the countries to develop, and the decline of the economy.

Brunello reviewed the economic literature to define skill mismatch, skill shortages, and skill gaps in Europe (Brunello, 2019) . The definitions of these three terms are as follows;

- At the macro level, the term "skills mismatch" refers to the discrepancy between supply and demand, but at the micro level, it refers to the mismatch between the degree of skills demanded of workers and those that are already available;
- Skills gap refers to the lack of certain skills for the workforce of the actual organization;
- Skills shortage refers to the lack of workers with the requisite skill level, at the usual “ongoing rate of pay”.

The authors’ aim was to show the consequences of skill shortages & mismatch in Europe in terms of economy. Thus, they started to review skill mismatch and skill shortages in a conceptual overview and tried to define how to measure the consequences. The results showed that in the long term, as new technologies are adopted, there is a demand for new skills that are not immediately present in the labor market. This leads to skill shortages until the general education system is ready to fulfill the new skill requirements. If wages and working conditions do not adequately signal relative scarcity, these shortages' significance and the adjustment process's length may increase. Skill shortages and mismatches are expensive for individuals, businesses, and society as a whole because they have a negative impact on earnings, productivity, innovation, and productivity growth. The responsibility for developing the abilities that employers seek, which includes funding skill development, can have relatively long-lasting implications on pay.

The "survey-based" is the methodology that is most frequently used in the current literature, an example is (Pellizzari, 2017). The authors create a theoretical framework that they then use to the OECD Survey of Adult Skills to determine the extent of skills

mismatch through an assessment of the degree of job matching (PIAAC). (Liu, 2016) put up a novel technique to estimating in which they assess the impact of a mismatch between industry demands and the skills of college graduates, particularly when there is an economic downturn, using data from the Norwegian market and a mathematical regression approach. They demonstrate that those market conditions have a long-lasting, albeit diminishing, detrimental impact on the likelihood of mismatch and, as a result, on the wages of the employees.

Because of the socioeconomic significance of skill mismatch, both academics and governmental organizations have looked closely at its impacts over the years. One of the old outstanding examples is (Manacorda, 1999) which focused on a point, that is also valid today. This research has concentrated on mismatch along the educational dimension, which serves as a good substitute for skills. According to the findings, skill mismatch should be a good indicator of recent structural changes in the labor market.

2.1.1 The Gaps Model for Education

(Parasuraman, Zeithaml, & Berry, 1988), three American authors, established the model of service quality, also referred to as the gaps model, during a systematic study program conducted between 1983 and 1988. The model describes the main aspects of service quality, recommends a scale for evaluating service quality (SERVQUAL), and offers potential solutions for issues with service quality. Dimensions of SERVQUAL are reliability, assurance, tangibles, empathy, and responsiveness. Based on this study, several researchers give their updated instruments creative names as ARTSQUAL (art museum), LibQUAL (libraries), EDUQUAL (education), and HEALTHQUAL (hospital) (Wikipedia, 2022).

To measure service quality in higher education, there are remarkable studies in the literature. One of the outstanding examples is (Yousapronpaiboon, 2014). To gather empirical data, a self-administered questionnaire was given to undergraduate students at five universities in Bangkok, with a random sample selected from both the public and private sectors. The study concluded that Thailand higher education fell short of

what undergraduate students had hoped for. The other examples are; (Legčević, 2009) measured the service quality of faculty of law in Croatia, (Al-Alak, 2012) faculty of Business in Jordan, (Md. Mamun-ur-Rashid, 2017) for Bangladesh, (Radenko Milojević, 2022) for Serbia.

The magnitude and direction of the 'internal gaps' affect the SERVQUAL model, which is used to evaluate service quality. The three gaps are as follows:

- Gap 1 (positioning gap) between customer expectations and management's perceptions of those expectations;
- Gap 2 (specification gap) between management's views of customer expectations and the firm's service quality specifications;
- Gap 3 (delivery gap) between service quality specifications and actual service delivery by employees. The fourth gap, the "communication gap," concerns both the actual provision of the service and external communications about it (Khodayari & Khodayari, 2011).

The fourth gap is communication gap, which concerns both the actual provision of the service and external communications about it (Khodayari, 2011). The last gap is the perception gap that means discrepancy between the customer's internal perception and expectations of related services.

To measure the gap between academia and businesses, in the literature, it seems that the most preferred method is surveys and questionnaires. After finding the gaps, the most important thing is to be able to make suggestions on how to take measures in the face of this problem, what improvements can be made in curriculums.

The management of digital transformation should involve higher education institutions. On the one hand, they are expected to offer educational and training opportunities to satisfy the demands of both students and the businesses at the time; on the other hand, they must be adaptable so that their programs can be effectively modified to fit future trends (Carayannis, 2022). Today's curriculum development

typically considers the contemporary curricular frameworks of other countries in addition to building on the previous curricular papers of the specific school system. (E.Rata, 2021) put forth one modern paradigm in response to the "return to knowledge" movement. In OECD's (Organization for Economic Co-operation and Development) 2030 Learning Framework and Learning Compass, which include a range of competences that include knowledge, skills, attitudes, and values that enable a person to act in a coherent and responsible way that changes the future, the OECD recently produced their own version of "21st century skills" (OECD, 2019).

The critical skill set and the discrepancies between industry expectations and academic standards have been the subject of numerous studies in the literature. Most of these articles are based on questionnaires or surveys. In one of the outstanding examples, the author started his work with a large-scale literature review to focus on the gaps between the software industry and the academia (Akdur, 2021). The online survey was used to reach more SW practitioners, the chosen method was "accidental non-probabilistic sampling" via LinkedIn. The designed survey was delivered to "659 participants working as SW practitioners from 14 countries", but only the participants who graduated from Türkiye and working as "Embedded Software Practitioners" (393 participants) were included in the study. The primary goal of this study was to find out the answers to the following research questions;

- the most important "Software Engineering Key Areas" for embedded software-related occupations,
- What are the industry expectations following a university education in terms of knowledge gaps and coverage?
- What soft skills are most crucial to the embedded software sector?
- Does the academic program include any courses that help students develop these soft skills?

At the beginning of the survey, a questionnaire was prepared to gather demographic data of the participants. The survey consisted of two parts, the first part aimed at finding out the most important hard skills of embedded software-related jobs, and the

second part for soft skills. According to the survey results, “hard skills analysis” and “soft skills analysis” was done according to demographics. The results showed that curriculums were designed to catch technological developments, but there were still hard skills gaps. The results revealed the need for a truly major updated curriculum for soft skills. The author completed the study by making suggestions to the academy in Türkiye by providing the expected skillset of Embedded Software related jobs.

A similar study to (Akdur, 2021) is (Marwedel, 2020). The 2020 survey in (Marwedel, 2020), was proposed to provide a framework for the many methods for producing graduates who are prepared for cyber-physical systems (CPS). The authors emphasized the gaps in the curriculum and discussed the difficulties in developing both technical and soft skills. Due to the focus on soft skills and the capacity to really build systems, they recommended using a variety of projects or instructional methods to hone diverse skills.

(Di Luozzo, D'Orazio, & Schiraldi, 2021) used job descriptions on LinkedIn in the fields of operations management (OM) and supply chain management (SCM) as the source of their analysis to find the skills gaps between academia and industry. In a similar study presented by (Guoyan Li, 2021), the authors analyzed the DSA (data science and analytics) skills gaps, in order to identify the most important and critical skillset for intelligent-manufacturing and data science related occupations. The gap analysis was carried out in U.S. on Emsi job advertising and profile data in this paper offered insights into the trends in manufacturing jobs that utilize data science, automation, cyber, and sensor technologies. (Pınar Özdemir, 2023) presented the purpose of the study as to address the need for a new postgraduate program that will bridge the curriculum gap between the present postgraduate programs and the current marine industry requirements. The ultimate objective should be to promote sustainable management of the marine sector, thus new programs with new courses or materials should be established to satisfy the educational demands of the sector and to build skills to raise awareness on growing maritime concerns. In this paper, a survey, responded by 224 stakeholders from 5 different countries, was prepared as a part of

MINE-EMI (Maritime innovative network of education for emerging maritime issues) project.

A different approach was presented by (Moldovan, 2019), which was based on a large-scale survey implemented among two different target groups which are manufacturing small and medium enterprises and vocational, educational, and training providers from 6 European countries for work place or work-based learning processes.

In general, in the literature for the skills gap analysis, it has been seen that the most preferred method is the classical method of conducting a survey. While designing curriculums, identifying professional demands with the help of domain professionals with questionnaires or workshops, and then fitting results to academic qualifications with the help of domain educators, comes across as a classical method.

2.2 European Skills, Competences, Qualifications and Occupations Classification (ESCO)

2.2.1 The Purpose / Mission of ESCO

More than ever, skills are important today. The pace of change in the economy and communities has accelerated due to the crisis. Instead of focusing on official credentials, employers are becoming more concerned with what employees know, understand, and are able to do in the real world. A rising appreciation of the significance of transversal skills and competences, such as communication abilities, learning capabilities, and a feeling of initiative, is another development. In response to this change, public and private employment services gradually shift their attention from an occupation-focused to a skills and competence-oriented strategy. Thus, several employment services have started to add skill lists to their existing occupational classifications.

A committee of impartial experts advocated creating a common vocabulary between education/training and the workplace as part of the New Skills for New Jobs project

(2009). This idea was supported by "A European strategy" and the Education Council which advocated for a common language and an operational tool and were endorsed on May 13, 2010. In addition, access to high-quality education, training, and lifelong learning is mentioned in the first principle of the European Pillar of Social Rights in order to achieve full involvement in Europe. The European Skills Agenda, which is the last step, offers a vision and specific actions to support lifelong learning and provide people the freedom to develop the skills they need for both work and life. These requirements are met by European Skills, Competences, Qualifications and Occupations (ESCO) (joinUp, 2022).

"ESCO (European Skills, Competences, Qualifications and Occupations) is the European multilingual classification of Skills, Competences, and Occupations. ESCO works as a dictionary, describing, identifying, and classifying professional occupations and skills relevant to the EU labor market and education and training" (European Commission ESCO, 2022). (Brunello, 2019) believed that ESCO is the most remarkable initiative to reduce skills gaps in the industry and in academia.

2.2.2 ESCO Database

ESCO Database has two pillar structures: Occupations and Skills & Competencies. ESCO V1.1 contains 3.008 occupations, 13.980 skills & competencies in 27 languages. The ecosystem of V1.1 is shown in Figure 3. The ESCO ecosystem explains the structure of ESCO. This structure provides mobility across Europe by offering up-to-date occupations and skills of employment and education continuously. The new minor version of ESCO, ESCO V1.1.1, including an updated skills/competence hierarchy structure that means revised mapping for skills concepts to skills groups is released. In this version, also more than 1.200 skills and knowledge concepts indicate digital competencies (European Commission ESCO, 2022).

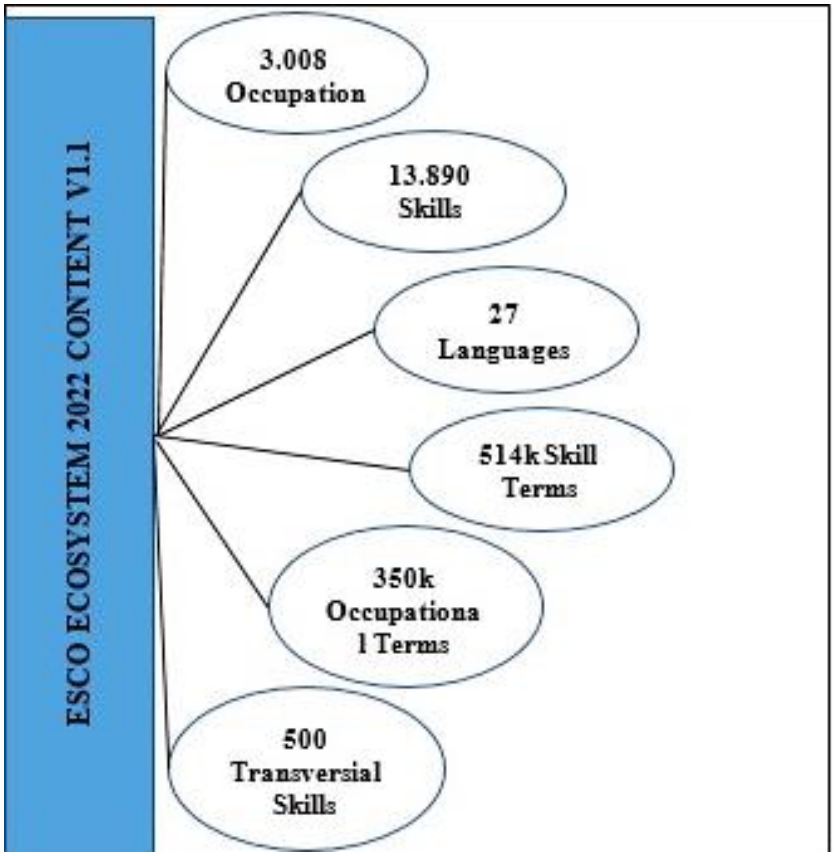

Figure 3: ESCO v.1.1.1 ecosystem

The occupation pillar has a hierarchical structure and it uses International Standard Classification of Occupations (ISCO) to organize the structure of the occupations. It is a set of job profiles with similar tasks. There are 10 occupations groups;

0. Armed Forces Occupations
1. Managers
2. Professionals
3. Technicians and associate professionals
4. Clerical support workers
5. Service and sales workers
6. Skilled agricultural, forestry and fishery workers
7. Craft and related trades workers
8. Plant and machine operators and assemblers
9. Elementary occupations

The row number is called “Code”, and when one occupation (major group) is chosen from the list, a detailed “Description”, “Sub-major Group”, and “Narrower ISCO Groups” explanations are listed. The hierarchical structure of the occupations pillar provides a more specific and detailed job profile when going down to the bottom. At the bottom, essential skills and competences, essential knowledge, optional skills and competences, and optional knowledge is reached. Also, alternative labels for the chosen occupation are shown to support searching the same profile with different keywords.

The skill & competence pillar provides a detailed list of skills according to criteria as “essential” or “optional”. Skills pillar has a hierarchical structure and contains four main sub-classifications;

- Knowledge
- Language skills and knowledge
- Skills
- Attitudes and values

Additionally, there are subsets of skills;

- A transversal skill hierarchy
- A collection of languages
- A collection of digital skills

ESCO provides two-way search, it can be done from skills to occupations and from occupations to skills.

2.2.3 Frameworks for Mapping and Consolidating Models for the Skills, Competences, and Knowledge

There are some other frameworks for mapping and consolidating models for the skills & competences, and knowledge for ICT occupations which are the European e-Competence Framework (e-CF), Body of Knowledge (BOK), and Occupational

Information Network in USA (O*NET). (Fernandez, 2017) The authors stated that by developing a framework called e-Skills match, they enable the development of recommender modules that will guide job seekers in their training for target occupations and the verification of knowledge compatible with certain skills. With this effort, they developed the eSKM framework, which is argued to be a coherent framework integrating ICT-related reference schemes and standards, namely ESCO, e-CF, and BOK. This work was a part of the eSKM project funded by the European Commission. The final framework highlighted the linkable parts of existing frameworks while producing a method to create a clear and functional interaction between competences, job profiles, skills, and knowledge. The authors believed that the lack of standardization and not using common language in the definitions of ICT professions would affect mobility, and therefore they focused on an integration study for ICT professions. The authors also claimed that merging the competence frameworks is possible with additional support by keeping those frameworks up-to-date.

2.2.4 Comparison of ESCO with Other Qualifications Frameworks

A study comparing ESCO with other qualifications frameworks was also done by (Neutel, 2021). The authors examined a methodology for automatic alignment of occupation ontologies which are ESCO and O*NET using NLP (Natural Language Processing) methods. O*NET application was developed by the National Center for U.S Department of Labor. Like ESCO, O*NET includes skills, knowledge, abilities, and work activities related to occupations (O*NET Online, 2022). (Neutel, 2021) analyzed occupations classifications of ESCO and O*NET by using Fasttext Labels, Fasttext Descriptions, BERT CLS Descriptions, BERT Mean Token Descriptions, and SBERT Descriptions. As a result of this study, they believed that the data in both ESCO and O*NET were not compatible to make a comparison, because datasets were not hierarchical and the classification of occupations in both frameworks was very different from each other. This has led to the conclusion that structural information was useless for this use case. They confirmed that even though SBERT has not yet produced a ready-to-use alignment, it performs significantly better than the more

traditional methods and offers a promising foundation for creating alignment systems that are more efficient.

O*NET (Occupational Information Network), called as tool for career exploration and job analysis, was developed by the U.S. Department of Labor/Employment and Training Administration to characterize occupations in terms of the knowledge, skills, and abilities needed as well as the tasks, work activities, and other descriptors that help to describe how the work is done like ESCO (O*NET, 2023). A frequently updated collection of information on occupational features and worker requirements from across the US economy is used to operate the O*NET system. The O*NET database is compiled and maintained through ongoing surveys of employees in each occupation, often with additional input from specialists in those fields. The users of O*NET portal are students, counselors, business, researchers, and developers. Also, ESCO can be reached through O*NET portal with ESCO crosswalk search interface. The content and functionality of O*NET show their value for research on labor markets (Handel, 2016), occupations, and university-level training. O*NET is used in studies for a variety of objectives, including exploring how work characteristics change over time and assessing the susceptibility of jobs to automation by looking into occupational skills and activities (Josten, 2020).

O*NET portal has online features;

- Occupation keyword search,
- Find occupations: 8 different ways to search
    1. Bright look feature has 4 different sub features;
       occupations that are;
        - expected to grow rapidly,
        - have numerous job openings,
        - new and emerging,
        - all
    2. Career cluster: includes occupations that have similar skills sets.
    3. Hot technology: technology and software related skills commonly included in the job postings,

4. Industry,
5. Job family,
6. Job zone;

   Into five categories;

   - Little or no preparation needed,
   - Some preparation needed,
   - Medium preparation needed,
   - Considerable preparation needed,
   - Extensive preparation needed,

7. STEM (Science, Technology, Engineering, and Mathematics)
8. All

- Advanced searches through job duties, professional associations, related activities, soft skills, technology skills
- Browse by O*NET Data: abilities, interests, knowledge, skills (basic, cross-functional), work activities, work context, work styles, and work values.
- Crosswalks: military, education, occupation handbook, SOC (2018 Standard Occupational Classification code or title), DOT (Dictionary of occupational titles code or title), RAPIDS (Registered Apprenticeship Partners Information Data System code or title)
- ESCO
- O*NET Interest Profiler
- My Next Move

ESCO is becoming more widely known and utilized on a global scale. Stakeholders frequently view ESCO and O*NET as the industry-standard frameworks for jobs and skills. They both have similar occupations. There are no defined high-performance computing related occupations in O*NET yet, but unlike ESCO, O*NET has more high-performance computing related skills.

When comparing O*NET with ESCO, it is seen that the data structure in O*NET is more broken down and has more search possibilities. With the method developed in this thesis, O*NET skill-occupation data can also be used to analyze the semantic

similarity with the curricula of universities, just like the comparison made with ESCO, only using the data source as O*NET will be sufficient for the method to work.

2.2.5 Maintaining and Improving ESCO

Today, the European Commission continues to maintain and improve ESCO to keep it up to date. “Data Science Technologies” including statistics, data analysis, machine learning, and artificial intelligence are used to provide the best tool and to prevent ESCO from information gaps. There are past researches that recommend ESCO be improved to bridge the gaps in knowledge, skills, and occupations. (Kahlawi, 2020) concentrated on determining how similar the descriptions of different ESCO objects are, they use the “Latent Semantic Index” (LSI) technique. Exploratory paths are intended to be made possible in order to optimize queries and find relationships and redundant ESCO descriptive items. The primary goal of the authors was to organize knowledge about the European labor market as well as about the education and training industry, in order to better match educational requirements with labor market demands and to better connect job seekers with employers. The study was based on improving ESCO skills descriptions. The early findings supported the idea of integrating the suggested signs into ESCO interfaces, empowering query protocols, and testing and debugging repository contents automatically.

Another study (Chiarello, 2021), which developed a text mining algorithm by extracting information both from ESCO and Industry 4.0-related scientific literature and then comparing the results for identifying the gaps in ESCO, showed that ESCO partially complies with Industry 4.0-related technological advancements. Here, examples of how text mining techniques can be used to analyze data on the new skills requirements brought on by Industry 4.0 are provided, ensuring that the information provided by ESCO is up to date. The authors emphasized the importance of ESCO timeliness as “Rapid technological change means that ESCO needs to be updated in a timely manner”.

ESCO is designed to be used by developers as a building block for different types of applications such as auto-complete/suggestion systems, and job search/matching algorithms and is published as Linked Open Data (LOD). LOD provides ease of integration of data into other systems, ease of link to other data, and ensures up-to-date data without administrative costs and quality-assured data (European Commission ESCO, 2022). Tim Berners used the following sentence in line with this LOD subject: “it will be used by other people to do wonderful things, in way that they never could imagine”.

An explanatory study (Smedt, 2020) provided an explanation of how the Simple Knowledge Organization System was used to develop the ESCO data model and agreed that ESCO serves as the foundation for a system of semantic resources in the labor market. They go into additional detail about how datasets are updated and how applications might use them. They emphasized that it is essential to maintain the ESCO classification throughout time so that it keeps up with new changes in the labor market or in the education and training sector if ESCO is to become a “de facto standard”.

2.2.6 The Skills Gaps Analysis Through ESCO

If the literature is searched from another point of view, it is seen that there are studies to determine the gaps between the academic curriculum and the expectations of the industry. There are several studies that concentrate on understanding the required skills/competencies and the gaps between academia and industry expectations. (Di Luozzo, D'Orazio, & Schiraldi, 2021) examined the key job tasks of the Operations Manager and Supply Chain Manager in the field of Operations Management and Supply Chain Management. Using ESCO, they conducted a gap analysis for the required skillsets of these key job tasks. The authors made a frequency analysis of the most common verbs used in ESCO skills definitions for both occupations to build up a methodology including deductive content analysis of Job Description Sets (311 JDs extracted from LinkedIn) and Occupations. Their methodology depended on past research of (D'Orazio, 2020) for SCM Competencies Framework for job posting

analysis. The results revealed that there was a need for deepening Soft Skills in ESCO classification.

(Vasilicia, 2019) stated that "keeping the curriculum up to date is today's necessity, because tomorrow it may no longer correspond to the requirement". The article described how to use ISCO and ESCO to design academic programs that will make it easier for credentials to be recognized. The model had the benefit of being founded on widely accepted standards. The authors stated that it can be used in higher education and professional training, but learning objectives must be set with employers based on ESCO competencies. The suggested model was simple to use and comprehend, and it also establishes an ISCO-compliant hierarchy of abilities. It was clear that the authors used the same pyramid as USA Skills Model, the only difference is that they included ESCO abilities and competencies that are ordered by certification levels and ISCO groupings. The result stated that if ESCO completes its first tasks and receives additional backing at the European and local levels, the labor market and education market can be related.

(Trevelyan, 2019) claims that with only minor curriculum revisions, student engineers can easily understand that productivity enhancement is the ultimate goal of engineering. Even though the curriculums may have been created with that goal in mind, the author also argued in this work that universities may not currently be the best setting for students to learn about engineering practice and how to deliver results that are in accordance with expectations. To ensure that engineers learn engineering practice in the workplace for the benefit of businesses, societies, and humanity, the author proposed that higher education institutions, businesses, and governments work together.

There are also many other technical studies to provide insights for improving ESCO, comparing ESCO with other qualifications frameworks, and explaining the structure of the ESCO Database.

Although various theoretical methodologies have been developed to determine the extent to which proficiency mismatches are related to different occupations, there are hardly any contributions that provide quantitative measures to compare this mismatch with curriculums of MSc programs. In addition, while relevant efforts have been made to standardize the job description and enrich the ESCO database, especially considering the EU job market, insufficient attention has been paid to the comparison of EU qualifications with postgraduate courses in the literature. Apart from those who want to pursue an academic career after graduating from the university, the MSc helps to gain the necessary qualifications to specialize in a field after graduating from the university. Therefore, especially the curriculums of MSc programs should match the requirements of the industry. For these reasons, this thesis study follows the path of scientific research and aims to provide useful insights and perceptions in order to evaluate the competencies of existing/new graduate programs.

2.3 Natural Language Processing (NLP) method for semantic comparison of textual data

The author stated that "*The NLP is the subject of computational linguistics—the study of computer systems for understanding and generating natural language*" (Chowdhary, 2020). Computational linguistics and theoretical linguistics are the two subfields of linguistics. The goal of computational linguistics has been to create algorithms that can effectively handle a variety of valuable natural language input, while theoretical linguistics has mostly concentrated on one component of language proficiency, grammatical competence - how people perceive some sentences as correctly adhering to grammatical norms and other sentences as being ungrammatical. They are interested in linguistic universals, or grammatical rules that hold true for all natural languages. By simulating the cognitive processes that are known to exist in the human brain and are used for language processing by humans, many new computational models are making attempts to close the cognitive gap. These methods rely on semantic elements that the text cannot explicitly express. The computational models are valuable for both theoretical and practical and commercial applications, such as facilitating efficient human-machine connections. Examples of theoretical

applications include research investigations that explore the nature and features of linguistic communication.

Similarity measures are becoming more crucial in NLP, and as a result of this there are various methods tried for calculating sentence similarity. In text related research and applications like information retrieval, text clustering, text mining, similarity measures have been utilized. These usages demonstrate how computing sentence similarity has evolved into a standard component for the knowledge representation and discovery for research community. There were a lot of researches on comparing documents' similarity in general, but there were very few articles on comparing the similarity of short texts and phrases in the past years. But nowadays, calculating similarity for short texts and phrases has been used frequently.

The semantic comparison of text data from two different data sources, which is intended to be done with this thesis, using NLP is very common in the literature. Here, the data sources for the texts for which semantic comparisons are to be made are the curricula of universities and the skills of ESCO. Some authors have looked at the relationship between universities and the development of the job market using big data and text mining tools. Regarding the approaches utilized, the majority of studies have trained a classification algorithm using machine learning to extract information from documents on knowledges, skills, and job profiles. In addition, (Chiarello, 2021) showed that, in contrast to job postings and educational institutions, scientific literature is more prepared to map new trends. As a result, it can be regarded as a reliable source. It makes sense to assume that, rather than the other way around, scientific literature influences both the job market and academic programs. Similar to author, (Kipper, 2021) also proved that keywords from scientific articles can be used to map and provided insights on future needs with respect to skills by using text mining techniques.

Use of big data and text mining techniques, various authors have examined the correlation between educational and advancement and changes in the labor market. To extract latent knowledge concealed in documents, text mining was used in a variety of techniques for the automated extraction of information from written resources and their efficient synthesis (Gupta V., 2009).

(Chiarello, 2021) examined comparing scientific papers with ESCO skills by using Named Entity Recognition (NER) method. A lexicon-based approach was utilized for this purpose's NER method, which uses the lexicon created by another author to automatically identify Industry 4.0 technologies. They examined the scientific papers and grouped the papers in the year of 2018-2020 and 2012-2020 to compare. The analysis presented here showed how well ESCO v.1.0 and v.1.1 covers the technologies and accompanying skill requirements related with Industry 4.0.

While there are numerous studies in English language, the label lemmatization with the UdPipe package was used in (Spada, 2022) for the texts in ESCO skills and marketing related documents for the marketing field in Italian language. They used single word extraction, sequential keyword matching, and semantic similarity methods. After applying the methods, to calculate similarity between textual data sources, the authors used Bag-of-Words (BoW) (Harris, 1954) and Bidirectional Encoder Representations from Transformers (BERT) model (Devlin, 2019). As a summary, they examined more than 1200 marketing-related documents from the labor market and higher education institutions in Italy to propose a data-driven approach to examining the congruence between educational and labor market components in the marketing area. To automatically identify the skills mentioned in job openings and degree program descriptions, they employed NLP techniques. The frequency of the competences obtained in both sections was then measured, allowing for a comparison of the two textual sources. The findings demonstrated that the concentration and distribution of skills taught by Italian universities are in line with those required by the labor market.

CHAPTER 3

3 METHODOLOGY

3.1 Research Problem and Research Objectives

Universities' abilities to train highly-qualified professionals is under attack of rapid technological developments. The HPC is also recognized as one of these rapidly developing fields. For this reason, universities in Europe have started working to open new master's programs in order to produce graduates who are experts in the field of HPC. While some of these MSc programs will be opened from scratch individually, others have started to place the HPC-related courses of other departments under the appropriate programs. In this context, with a new project (EUMaster4HPC), it is aimed to create the curricula of the newly opened HPC MSc programs.

In this thesis, an assessment methodology has been developed to achieve the defined objectives. These objectives were written in Section 1.3 where the main aim is to compare textual data from the ESCO with information from university MSc degree programs, and the methodology depends on NLP techniques and presents a new assessment method in the area of high-performance computing. This method will contribute to measure the universities' capacity to adapt to market change quantitatively and objectively. A methodology like this is the most needed in the field of HPC because traditional educational institutions require time to successfully develop and execute new or updated curriculum, and the results of curriculum changes can only be measured within a long time.

Quasi is defined as "similarity", since revealing the similarity of data in ESCO and curricula, and measuring the extent to which they overlap are the main goals in this thesis, the quasi-experimental research method was used. The data collected in this thesis, could be used as a basis to create new research ideas for future studies.

3.2 Overall Research Design

A broad technique has been devised to carry out the skills alignment assessment between the MSc curricula for the chosen universities and the ESCO classification. The process is explained as follows:

Data Gathering: In this thesis, quantitative data collection method was used. This step was completed in two levels. The first level included gathering data from;

1. EUMaster4HPC Deliverable 2.2: To determine occupations, the most wanted profiles in the HPC field were selected from 577 job postings in Deliverable 2.2. Although job postings were viewed only from a non-industrial point of view in this study, non-industrial ones, among the most sought-after profiles listed in the job names and subarea fields, were eliminated and only industrial ones were selected. This deliverable was published as public. Furthermore, universities, whose curricula will be analyzed, were selected based on the awarding universities at the beginning of data gathering step.
2. e-CF: The roles in the e-CF were selected (pre-defined roles, coming from EUMaster4HPC Deliverable 2.2 profiles) and detailed information was obtained by listing the details of the roles.
3. HPC Skill Tree: The main tasks in the roles from e-CF and the top-level skills in the HPC Skill Tree were compared and the occupations to be used in the analysis were reached within the scope of HPC top level skills.
4. ESCO: For the purposes of the research, ESCO classification was used. ESCO scanning was performed for occupations obtained as a result of the collected data. ESCO dataset v.1.1.1 of the API was downloaded in the “classification” content in English as csv file to reach ESCO occupations-skills matrix.

The second level included gathering data from MSc Programs Curriculum. By listing the course contents, information on which hard/soft skills were available in which course was found out to perform an assessment for identifying skills gaps in MSc programs according to industry perceptions for HPC area. The criteria for choosing HPC MSc programs will be discussed in the following sections.

Data Selection: This step allowed to determine the data to be used in the assessment. The chosen data are EUMaster4HPC Deliverable 2.2, e-CF, HPC Skill Tree, and ESCO to create comparison tables for defining the profiles in HPC and required skillsets of these profiles. EUMaster4HPC Deliverable 2.2 and HPC Skill Tree provided information for industry requirements. ESCO provided the occupations-skills matrix. These three data resources were used as independent variables in assessment methodology.

Course syllabuses were used as the under-observation groups.

There were three steps followed in data selection process;

1. Defining the most related job profiles in HPC field,
2. Listing the skills required for chosen job profiles,
3. Finding the MSc HPC programs that describe syllabuses and course contents in sufficient detail to be able to do research.

Data Analysis: Gathered data (defined in "Data Gathering" step) were disorganized and non-correlated. Before starting the quantitative assessment methodology, the data needed to be correlated and structured. So, the tables shown in Section 4 were created for the data interpretation step.

Data Interpretation: In order to transform the collected data into information, the tables shown in Section 4 were created.

Data Cleaning: Data cleaning was applied first in the "Data Analysis" step to ensure the correlation within the created tables. These tables form the basis of the assessment

methodology. Basic data wrangling and text cleaning activities must be performed on the textual data (Grolemund, 2018). The other data cleaning procedures applied in NLP were;

- Sentence detection
- Removing stop words
- Making all letters lower-case

Data cleaning was applied manually for course syllabuses' text in the following way;

- Remove bullets in the textual data,
- Delete the data or sentence in the format that will not be included as any skill in ESCO

For example: *"The course consists of a set of lectures and laboratory session"*
"The course covers the following topics"
"The content is divided into several parts"

This type of data cleaning was made manually because NLP and the training set used in NLP was not trained to filter or eliminate such kind of sentences that were irrelevant with ESCO skills.

These cleaning procedures were defined in detail in Section 4.

Skills Assessment Methodology Using NLP: After applying the above steps, data was made ready for comparison by using NLP. In this step, the methodology was developed to compare the ESCO skills with the HPC MSc curricula by applying NLP methodology.

Data Visualization: After the information, obtained from the applied evaluation methodology by using NLP, started to be sufficiently meaningful and stable, this

information was recorded and the results were interpreted. Visualization techniques were used to present the results.

For the above-mentioned processes an illustration is shown in Figure 4;

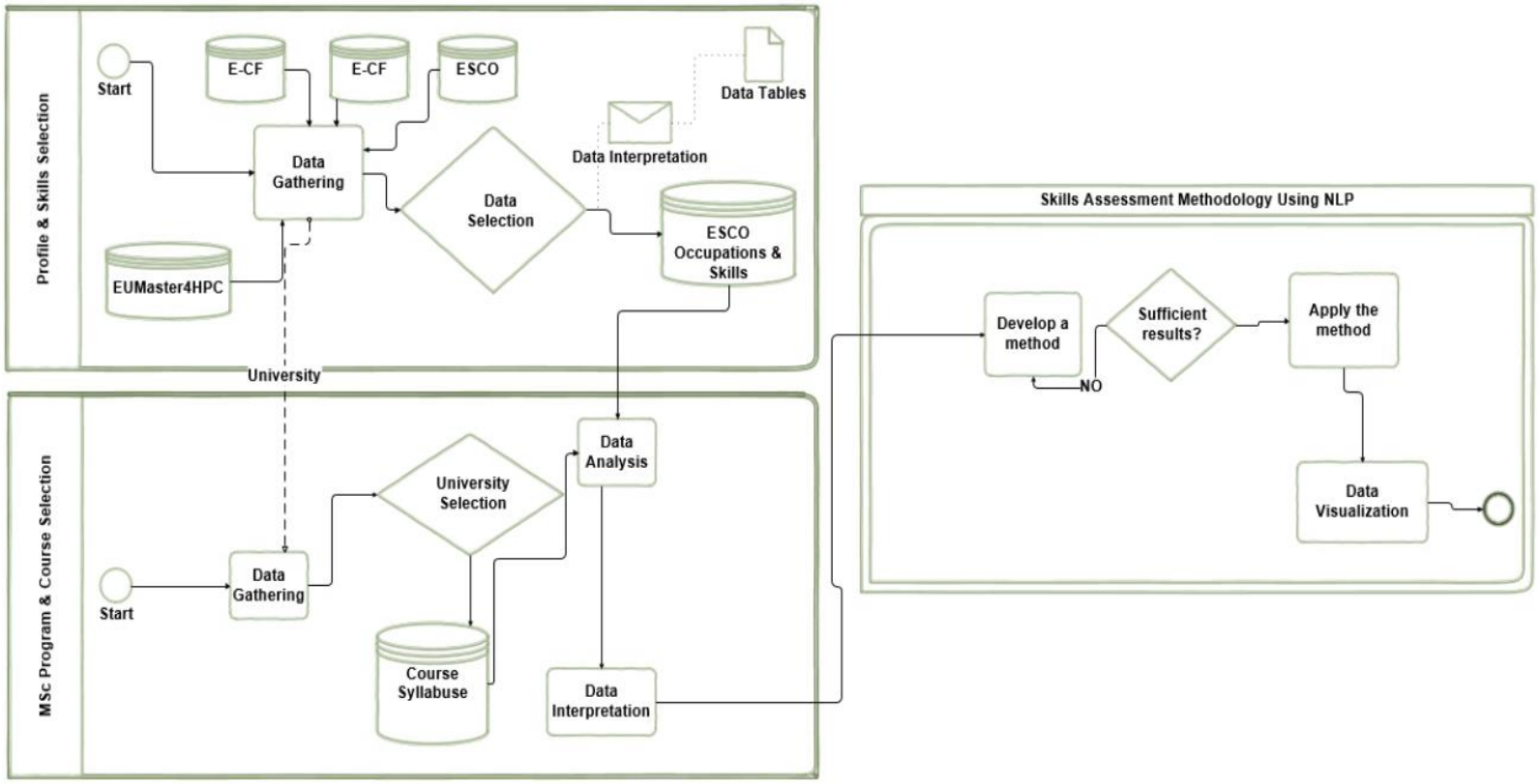

Figure 4: Skills gaps assessment methodology overview block-diagram

CHAPTER 4

4 IMPLEMENTATION

4.1 Data Gathering

4.1.1 Occupations/Profiles Selection

High-Performance Computing is defined as a technic of the use of parallel data processing for enhancing computing efficiency and carrying out complicated calculations. Performance, or time to solution, is the key characteristic of High-Performance Computing, often known as supercomputing. The developments in HPC encompass several centuries of counting and recording, nearly 500 years of automated computation, and the current era of digital computing, which ranges from vacuum tubes to multicore very-large-scale integration processors. Advances in economics, science, engineering, and society have been made possible by HPC (Thomas Sterling, High Performance Computing, 2018).

The best hard/soft skills, abilities, and personal characteristics should be sought after, especially in the fields of computer science and information communication technology, in order to keep ahead of the severe competition. Because of this, many industries have recently stepped up their efforts to identify, categorize, and characterize the skills required. Considering the current and future importance of HPC, it is therefore extremely important to be able to determine what skills are required to train qualified personnel who can work in HPC field, from now on.

In order to set the standard for European supercomputing, the European High-Performance Computing Joint Undertaking (EuroHPC JU) was established in 2018 and is headquartered in Luxembourg (EuroHPC JU, 2022). One of the EuroHPC JU's project is EUMaster4HPC. The goal of EUMaster4HPC is to establish a common curriculum in high performance computing (HPC) across Europe, define the body of

knowledge required to master the area, and build a collaborative network by utilizing and enhancing the existing European HPC ecosystem.

A higher education curriculum will be created by EUMaster4HPC in order to teach students about topics including the design, implementation, use, and/or operation of present-day and upcoming HPC systems and HPC-related technologies in Europe; to educate professionals capable of promoting HPC adoption and information transfer in various strategic sectors, thereby tying together HPC operations in industry and academia.

*"The **HPC European Consortium Leading Education Activities** (EUMaster4HPC)"* published the last deliverable version 2.2 in public. This deliverable's objective is defined as "*Analyse and document the skill profiles required on the future European HPC to serve the needs of scientific communities*." (EUMaster4HPC, 2022). With this work, academic sector related HPC occupations were analyzed to design a new HPC MSc program in Europe. The occupations were filtered from job portals as HPCWire, HiPEAc Jobs, and the BSC-CNS. The number of analyzed jobs is 577. They categorized the profiles as Applications, Parallel Programming & Tool Support, DevOps Engineer, Manager / Consultant, and System Architect. Although they state that they are only interested in non-industrial profiles in that deliverable, the interest in this thesis was both in industrial and non-industrial work profiles based on their study. The professional profiles created by the EUMaster4HPC were used when considering professional profiles and occupations in the HPC domain.

In order to select the occupations that are considered as case studies to develop the methodology, the data in EUMaster4HPC study were used. According to the results of the research, profiles in the job postings, as shown in Figure 5, the top four most sought jobs in the HPC field are Applications/Domain Expert, System Architect, Parallel Programming & Tool Support/Solution Designer, and DevOps Engineer. Moreover, in order to analyze in details *"Manager/Consultant"* was excluded because this profile was defined in the way of soft-skills.

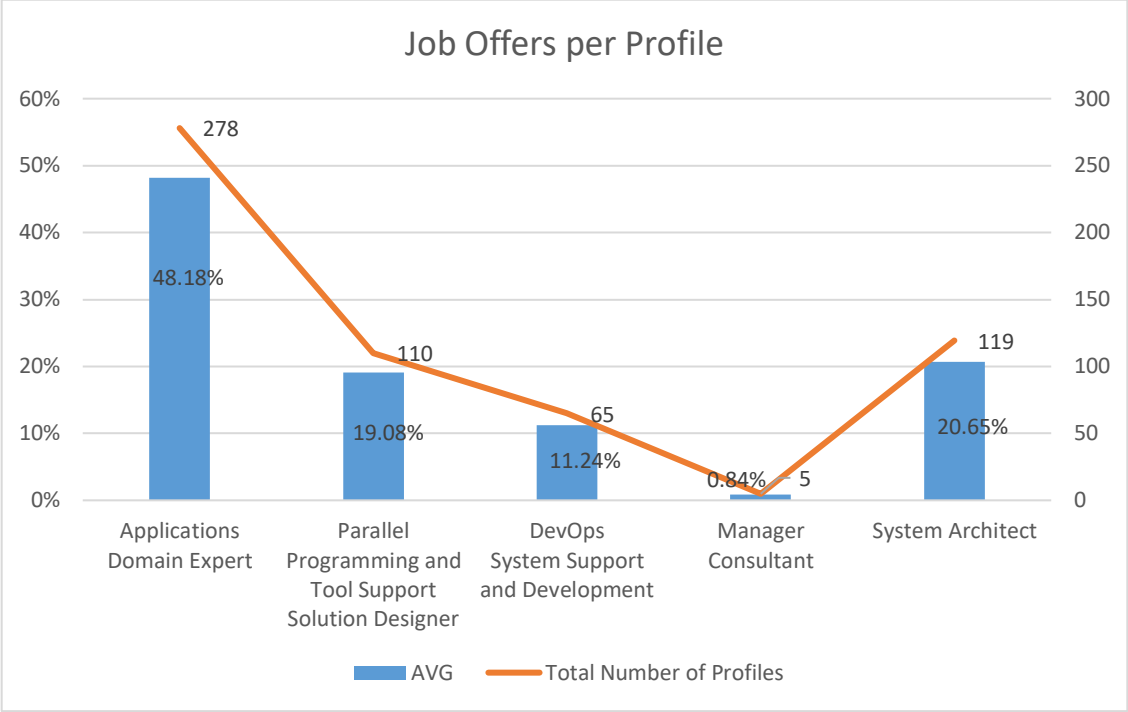

Figure 5: Job offers per profile in EUMaster4HPC public deliverable v.2.2. Future Needs HPC Scientific Areas

The job profiles were also defined in public Deliverable v.2.2. Future Needs HPC Scientific Areas;

- a specialist in a certain discipline who has had little exposure to a particular set of software tools is defined as “Applications” profile.
- a person with knowledge of general-purpose CPUs, graphics processing units (GPUs), memory subsystems, or specific accelerators is defined as “System Architect” profile.
- Someone with knowledge of specific tools and HPC infrastructure to make interactions with application users easier is defined as “Parallel Programming & Tool Support” profile.
- For system deployment, maintenance, and support of applications, servers, and HPC platforms that support the development needs of scientific projects across a wide range of languages, operating systems, and testing platforms, an IT expert in one or more areas is required. This IT expert is defined as “DevOps Engineer” profile.

According to the definitions of job profiles in EUMaster4HPC Deliverable v.2.2, a research was conducted and it was tried to find out which roles these profiles correspond to in e-CF. The reason for this research in e-CF is to find out what the corresponding job profiles are in the sector and to be able to reveal the necessary skill sets for these profiles by researching in ESCO with the results to be obtained from here.

Since the profile definitions mentioned above are not sufficient to carry out the necessary research in e-CF, the definitions of subarea and job names in deliverable 2.2 are used. Job profiles were introduced in detail by making an inference from these data.

According to job profiles and their descriptions/subareas defined in EUMaster4HPC deliverable 2.2, ESCO review was done for "High performance computing, supercomputing, grid computing, performance engineer, AI engineer, high performance computing architect, high performance computing engineer, neural data engineer, performance architect, machine learning engineer". However, it was observed that in ESCO there is no occupation with those names. According to this result, the need for a new study emerged and it was tried to determine what the professions defined in ESCO that could be related to HPC were.

As the foregoing has indicated, the HPC CF was guided to choose the most relevant HPC related occupations in ESCO. Defining the High-Performance Computing skillset is effort intensive, and there are some studies conducted in sector for this purpose. Internationally accepted one is HPC Competence Framework (HPC CF). This forum provides developing the skill tree to define HPC skillsets. Their goal is that as required by the HPC community, the skill tree corresponding to the competency standard will be created and modified. This implies that talents may be changed, added, or eliminated in order to best serve the community (HPC CF, 2022). In this thesis work, to cope with the challenge of finding the right skillsets for HPC, the skillsets provided by HPC CF was used as shown in Figure 6. A fishbone diagram for skill tree was created to show the hierarchy.

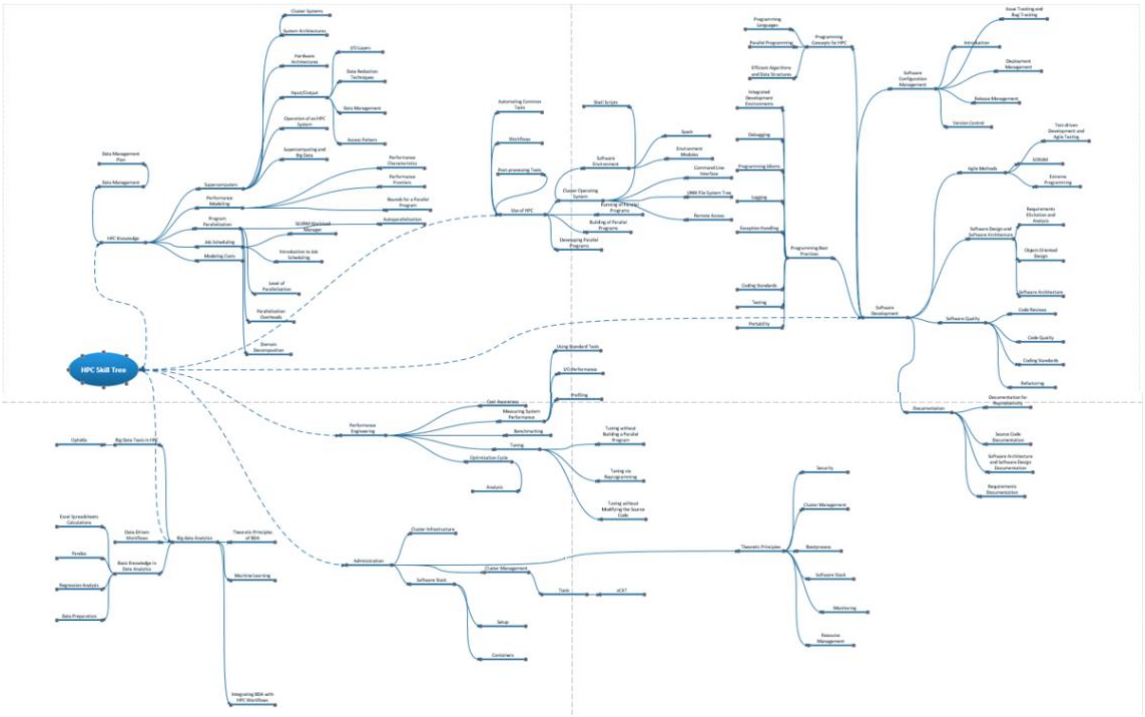

Figure 6: HPC skill tree fishbone diagram

The occupations are determined as *"Data Scientist, System Architect, Software Developer, and DevOps Engineer"* by analyzing the defined skills in ESCO. It also shows that ESCO does not yet contain sufficient up-to-date data for HPC-related occupations.

Before applying the methodology in more detail, the e-CF explorer was used to research 4 professions in terms of ICT. e-Competence Framework (e-CF) identifies 41 workplace competencies in Information and Communication Technology (ICT) using a common language of competence, skills, knowledge and abilities. This interactive tool enables exploring the competencies and 30 ICT professional role profiles defined by the European Commission for Standardization (CEN). In e-CF;

- "Dimension 1" represents the macro-IT process,
- "Dimension 2" represents all identified e-competencies,
- "Dimension 3" represents the skill level applicable to each competency.

Common areas for the 4 selected profiles were selected and their details from e-CF were listed and examined. The common areas are *"Data Scientist, Data Specialist, Developer, DevOps Engineer Expert, Solution Designer, Systems Analyst, and Systems Architect"* in e-CF, but only *"Data Scientist, System Architect, DevOps Engineer Expert, Developer, and Solution Designer"* were analyzed.

The e-CF roles listed above have been examined in terms of dimension 1, 2, and 3. A description of the considered roles is reported as in Table 1, exemplarily for "Data Scientist".

Table 1: e-CF roles and dimensions - Data scientist role detail

Dimension 1	Dimension 2	Dimension 3				
		e-1	e-2	e-3	e-4	e-5
Plan	A.7.Tecnology Trend Monitoring			X	X	X
	A.9.Innovating				X	X
Enable	D.10.Information and Knowledge Management			X	X	X
	D.11.Needs Identification			X	X	X
Manage	E.1.Forecast Development			X	X	

This table is an output of the e-CF. In addition to these, the main tasks of the role are listed.

Main tasks for "Data Scientist";

- Represent business challenges through mathematical models
- Collect, understand, clean, analyze, integrate and investigate internal and external data to achieve the mission
- Create and test hypothesis
- Uncover data correlations/relationships in support of measurement and predication
- Identify the right visualization models depending on the business challenges and the data sets
- Address data security through active preventative strategies

- Select and optimize algorithms using data science tools
- Comply with ethical guidelines and legal requirements

Proficiency levels were also marked for each e-competence. This table has been examined for all common areas of roles and main tasks have been registered to use for research in ESCO and the outputs of e-CF are shown in Figure 7, Figure 8, Figure 9, and Figure 10 as adapted from (European e-Competence Framework, 2022).

Dimension 1	Dimension 2	Dimension 3				
		e-1	e-2	e-3	e-4	e-5
Build	B.2 : Component Integration					
	B.3 : Testing					
Run	C.2 : Change Support					
	C.4 : Problem Management					
Manage	E.8 : Information Security Management					

Figure 7: e-CF System administrator role detail

Dimension 1	Dimension 2	Dimension 3				
		e-1	e-2	e-3	e-4	e-5
Build	B.1 : Application Development					
	B.2 : Component Integration					
	B.3 : Testing					
	B.5 : Documentation Production					
Run	C.4 : Problem Management					

Figure 8: e-CF Developer role detail

Dimension 1	Dimension 2	Dimension 3				
		e-1	e-2	e-3	e-4	e-5
Build	B.1 : Application Development					
	B.2 : Component Integration					
	B.3 : Testing					
	B.4 : Solution Deployment					
Run	C.2 : Change Support					

Figure 9: e-CF DevOps Engineer role detail

Dimension 1	Dimension 2	Dimension 3				
		e-1	e-2	e-3	e-4	e-5
Plan	A.6 : Application Design					
	A.9 : Innovating					
Enable	D.10 : Information and Knowledge Management					
	D.11 : Needs Identification					

Figure 10: e-CF Solution designer role detail

The assessment methodology's objective is to compare quantitatively the overall set of competencies required for the development of the "Data Scientist, System Architect, Parallel Programming, and DevOps Engineer" professions in HPC by mapping those competencies to the definitions given in ESCO with syllabuses of universities' HPC related MSc programs.

In order to create a framework for comparison of data gathered from EUMaster4HPC public Deliverable 2.2, e-CF, ESCO, HPC Skill Tree, the comparison chart shown below in Table 2 has been prepared. This table has been created in order to ensure that the study is carried out on correct and meaningful data. To be able to create this table, data from ESCO, e-CF, EUMaster4HPC, and HPC Skill Tree were analyzed, classified and grouped according to job names & subarea in EUMaster4HPC Deliverable 2.2 and ESCO Occupation names and alternative labels. This table provides a mapping to the e-CF, ESCO, EUMaster4HPC, and HPC Skill Tree.

Table 2: Comparison table of chosen occupations / profiles for EUMaster4HPC, e-CF, ESCO, and HPC skill tree

EUMaster4HPC	EUMaster4HPC Chosen Subarea	e-CF	ESCO	Coverage in HPC Skill Tree
Application/Domain Expert	Data Science	Data Scientist, Data Specialist	Data Scientist, Data Engineer, Research Data Scientisr, Data Expert, Data Research Scientist	Big Data Analytics, Software Development
System Architect	Computer Architecture	Systems Architect	ICT system architect, ICT systems architect, solution architect, IT systems architect, IT system architect, systems architect, ICT system architect, ICT systems architects, information system architect	HPC Knowledge, Use of the HPC Environment, Administration
Parallel Programming and Tool Support/Solution Designer	Parallel Programming	Developer	Software Developer, Application Developer, Solutions Developer	Software Development, HPC Knowledge, Use of HPC Environment
DevOps/System Support and Development	DevOps	DevOps Expert	DevOps Specialist, DevOps Engineer	All

With this study, it was ensured that 3 sources from which the data were obtained, were linked with each other, and also coverage of profiles/occupations in the HPC skill tree was shown.

The goal in this case is to provide more proof of how Natural Language Processing can assist in managing and comparing a complicated skills dataset. Datasets are defined as 1) ESCO occupations and skills 2) course syllabuses in HPC MSc programs.

4.1.1.1 Occupations and Skillsets in the European Skills, Competences, and Occupations (ESCO) Database

Four profiles have been taken into consideration for the research because they are representative of the many occupations listed under the ESCO classification: *"Data Scientist, System Architect, Software Developer, and DevOps Engineer".* Local API of ESCO v.1.1.1 is used to access the data of these occupations. The data extracted using API was;

- Occupation Name: a group of jobs with activities, **3.008** defined occupations,
- Alternative Labels: alternative names of occupations,
- Essential Skills: list of compulsory basic skills to do the specific job/task,
- Essential Knowledge: list of compulsory basic expertise /proficiency to the specific job/task,
- Optional Skills and Competences: list of non-compulsory basic skills to do the specific job/task,
- Optional Knowledge: list of non-compulsory basic expertise /proficiency to the specific job/task,

There are **13890** skills and **514000** skill terms defined in the ESCO database.

The visual representation of ESCO in practice is given as an example to show "data processing" skill of "data scientist" occupation in Figure 11: Example of ESCO in practice for data scientist's skill "data processing" adapted from (Skill Man, 2022).

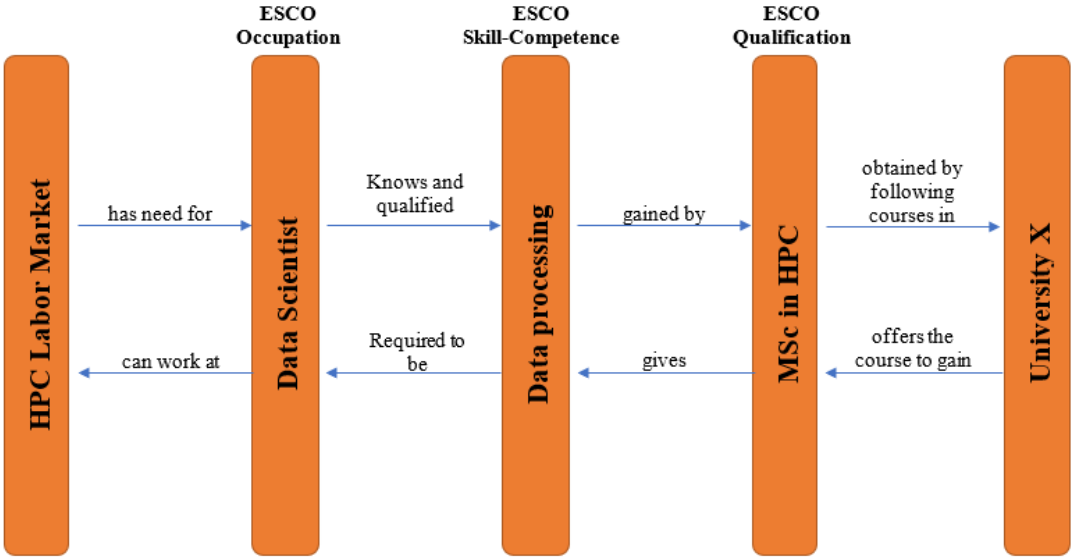

Figure 11: Example of ESCO in practice for data scientist's skill "data processing" adapted from (Skill Man, 2022)

The path in ESCO to reach selected occupation *"Data Scientist"* is defined in Figure 12.

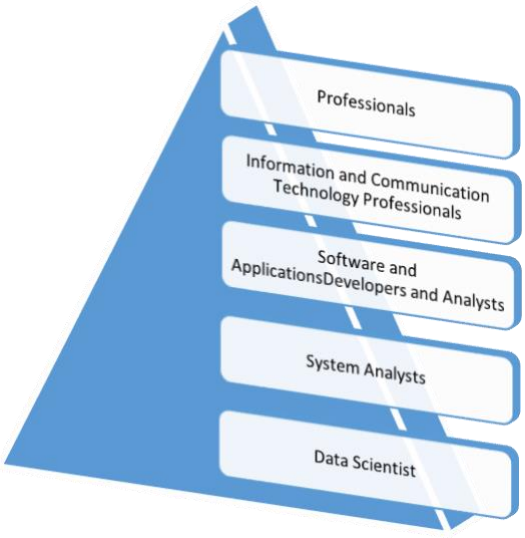

Figure 12: Data scientist occupation path in ESCO

The other occupations and their paths in ESCO are as follows;

- **System Architect:** Professionals→Information and communications technology professionals→ Software and applications developers and analysts→ Systems analysts→ ICT System architect
- **DevOps Engineer:** Not listed in ESCO's occupations. It was listed as under "Alternative Labels" of "ICT change and configuration manager" occupation. The path is: Professionals→Information and communications technology professionals→ Software and applications developers and analysts→ Software and applications developers and analysts not elsewhere classified→ ICT change and configuration manager
- **Software Developer:** Professionals→Information and communications technology professionals→ Software and applications developers and analysts→ Software developers

A description of the extracted information by using ESCO Local API for the selected occupations is shown in

Table 3 with details of “Occupation/Alternative Labels, Skills/Knowledge Type, and Skills/Knowledge/Competences.

Table 3: ESCO occupations / skills details table

Occupation/Alternative Labels	Skills/Knowledge Type	Skills/Knowledge/Competencies
data engineer data scientist research data scientist data expert data research scientist	Essential Skill	develop data processing applications,evaluate research activities,draft scientific or academic papers and technical documentation, develop professional network with researchers and scientists,design database scheme,execute analytical mathematical calculations, integrate gender dimension in research,perform project management,collect ICT data,manage data collection systems, implement data quality processes,manage personal professional development,synthesise information,use databases, conduct research across disciplines,apply for research finding,perform scientific reserch,disseminate results to the scientific community, establish data processes,interpret current data,manage open publications,perform data cleansing,speak different languages, write scientific publications,communicate with non-scientific audience,demonstrate disciplinary expertise, build recommender systems,manage intellectual property rights,mentor individuals,handle data samples, deliver visual presentation of data,increase the impact of science on policy and society,publish academic research, promote the transfer of knowledge,use data processing techniques,promote the participation of citizens in scientific and research activities, normalise data,manage findable accessible interoperable and usable data,think abstractly,manage research data, interact professionally in research and professional environments,report analysis results,apply research ethics and scientific integrity principles in research activities,promote open innovation in research,operate open source software,
	Essential Knowledge	visual presentation techniques,statistics,data mining,query languages,information categorisation, resource description framework query language,information extraction,data models,online analytical processing,
	Optional Skills and Competences	integrate ICT data,design database in the cloud,perform data mining,manage ICT data classification,apply blended learning, manage ICT data architecture,manage data,teach in academic or vocational contexts,use spreadsheets software,create data models,define data quality criteria,
	Optional Knowledge	Hadoop,data quality assessment,business intelligence,LINQ,MDX,unstructured data,SPARQL,Xquery,LDAP,N1QL,
ICT system architect ICT systems architect solution architect IT systems architect IT system architect systems architect ICT system architects ICT systems architects information systems architect	Essential Skill	create data models,manage system testing,apply ICT system theory,define technical requirements,use markup languages,analyse business requirements, acquire system compenent,design information system,use an application-specific interface,design enterprise architecture,assess ICT knowledge, implement ICT safety policies,integrate system components,manage database,align software with system architectures,
	Essential Knowledge	systems development life-cycle ,database development tools,business process modelling,hardware platforms,systems theory,web programming,
	Optional Skills and Competences	apply technical communication skills,plan migration to cloud,build business relationships,solve ICT system problems,design database in the cloud, design for organisational complexity,perform resource planning,use object-oriented programming,manage standards for data exchange, design database scheme,develop with cloud services,manage staff,design process,design cloud architecture,manage cloud data and storage, provide technical documentation,provide cost benefit analysis reports
	Optional Knowledge	COBOL,ICT System integration,ICT system programming,Smalltalk(computer programming),Scala,ICT project management methodologies, Java (computer programming),Scratch (computer programming),Assembly (computer programming),Prolog (computer programming), ICT security legislation,Ruby (computer programming),Pascal (computer programming),ABAP,Common Lisp,task algorithmisation,SAP R3, information structure,Perl,ICT process quality models,ASP.NET,Swift (computer programming),APL,Haskell,CoffeeScript, OpenEdge Advanced Business Language,SAS Language,Lean project management,Groovy,Microsoft Visual C++, defence standard procedures,computer programming,Visual Studio .NET,TypeScript,Oracle WebLogic,R,VBScript, ML (computer programming),Objective-C,MATLAB,model based system engineering,Agile project management,C++, PHP,AJAX,Process-based management,Erlang,Lisp,Python (computer programming),JavaScript,C#
configuration manager application lifecycle manager devops specialist software change and configuration manager devops engineer	Essential Skill	manage ICT virtualisation environments,integrate system components,build business relationships,use scripting programming, provide technical documentation,deploy ICT systems,administer ICT system,train employees,manage software releases, manage changes in ICT system,perform project management,develop automated migration methods,
	Essential Knowledge	tools for software configuration management,DevOps,ICT project management methodologies, ICT process quality models,project configuration management,
	Optional Skills and Competences	perform software unit testing,utilise computer-aided software engineering tools, perform software recovery testing,apply operations for an ITIL-based environment,
	Optional Knowledge	design cloud architecture,migrate existing data,apply change management Puppet (tool for software configuration management),Vagrant,Octopus Deploy,Jenkins (tool for software configuration management), Codenvy,Salt (tool for software configuration management),integrated development environment software,STAF, computer programming,Apache Maven,control objectives for information and related technology, Chef (tool for software configuration management),embedded systems
application developer application programmer solutions developer programmer software specialist application software developer software developers software engineer applications engineer software developer soft developer developer of software	Essential Skill	develop automated migration methods,interpret technical requirements,analyse software specifications,identify customer requirements,perform scientific research,use technical drawing software,utilise computer-aided software engineering tools,use an application-specific interface,manage engineering project,develop software prototype,use software design patterns,create flowchart diagram,use software libraries,provide technical documentation,define technical requirements,debug software
	Essential Knowledge	technical drawings,tools for software configuration management,engineering principles,project management,ICT debugging tools,engineering processes,integrated development environment software,computer programming
	Optional Skills and Competences	collect customer feedback on applications,adapt to changes in technological development plans,design user interface,use functional programming,use query languages,use concurrent programming,use logic programming,migrate existing data,utilise machine learning,integrate system components,use object-oriented programming,do cloud refactoring,develop creative ideas,use automatic programming
	Optional Knowledge	COBOL,blockchain openness,Oracle Application Development Framework,Scala,Java (computer programming),ICT security legislation,ABAP,SAP R3,Perl,Groovy,Visual Studio .NET,TypeScript,R,Objective-C,MATLAB,C++,Lisp,JavaScript,C#,Kdevelop,Eclipse (integrated development environment software),Internet of Things,Xcode,software anomalies,Salt (tools for software configuration management),blockchain platforms,Scratch (computer programming),Drupal,JavaScript framework,SQL,COBOL,Assembly (computer programming),cyber attack counter-measures,defence standard procedures,Prolog (computer programming),Ruby (computer programming),IBM WebSphere,WordPress,Haskell,PHP,NoSQL,SAS Language,Pascal (computer programming),STAF,smart contract,ML (computer programming),AJAX,Jenkins (tools for software configuration management),Smalltalk (computer programming),OpenEdge Advanced Business Language,Swift (computer programming),Apache Tomcat,Microsoft Visual C++,Apache Maven,ASP.NET,Puppet (tools for software configuration management),AJAX framework,Erlang,CoffeeScript,Ansible,Python (computer programming),Common Lisp,Object-oriented modelling,VBScript,software frameworks,APL,World Wide Web Consortium standards

4.1.2 Course Syllabuses of MSc Programs on HPC

The websites of courses (universities with MSc programs on HPC) are used to acquire course descriptions. It is not possible to extract course information through an API, because universities do not provide web services. In this case, it was decided to choose two universities with HPC MSc programs. These two universities and their programs were selected as examples in order to demonstrate the developed methodology. This study does not intend to provide guidance to prospective students willing to study in HPC-related programs. Therefore, to ensure confidentiality and prevent any misunderstandings, the names of these two universities will not be explicitly given and from now on they will be mentioned as the European university and the American university.

One of the universities is a European university and is a participant of EUMaster4HPC project, and the other university is from United States of America. Thanks to this selection, it will be possible to compare two universities in America and Europe that use different qualification frameworks.

According to the QS World University Ranking 2021, the European university is rated between 100-150. It receives citations per faculty of 76.1, employer reputation of 62.9, academic reputation of 28.9, and an overall score of 55.9. According to US News & World Report's Global Universities, this university's rankings is between 300-350. Also, in Times Higher Education (THE) rankings, it performs in the rank 251-300 for world university rankings, 88 for engineering & technology based on teaching, research, citations, industry outcome, and international outlook (Times Higher Education, 2023).

High performance computer systems master's program at this European university is the subject of this thesis. This two-year full-time MSc program is under computer science and engineering department. With current rising applications like artificial intelligence and deep learning in mind, this curriculum focuses on the hardware-software co-design components to construct domain-specific architectures. The program's main theme and core concept is how systematic approaches based on the most recent findings in the field of computer systems engineering can be used to address the needs of future industries in terms of high computational performance and energy efficiency. This master's program is intended for all students who want to gain in-depth knowledge, skills, and approaches in the field of computer engineering, specifically in the area of high-performance computer systems. The main subjects of the program are;

- Computer architecture,
- Parallel programming,
- Sustainability/energy-efficiency

The program was designed in the way that program starts with courses which create basis for high performance computer systems. These compulsory courses are computer architecture, high performance parallel programming, and sustainable computing. Elective courses enable students to choose the disciplines like machine learning, computational sciences, operating systems, and networking. So, the university prefers students with basic computer organization, machine-oriented programming, principles of concurrent programming, and mathematical modelling and problem-solving courses experience to start high performance computer systems master's program. The compulsory elective courses students can specialize also in entrepreneurship, computer graphics, computer systems, and real-time systems. Several of the compulsory elective courses offer a deeper focus in high performance computing subjects like advanced computer graphics, real-time systems, reliable real-time systems, and parallel computer architecture.

The second chosen university is from U.S.A in this thesis. In 2023 Times Higher Education rankings, it is between 10-50 rank for world university rankings, for US college rankings, and for world reputation rankings (Times Higher Education, 2023).

High performance computing master's program at this university is the subject of this thesis. This MSc program is under computer science department in this university. There are three other computer science specialization programs under computer science department which are;

- Application development,
- Data analytics,
- Software engineering

One of the compulsory courses in this program is high performance computing. Other compulsory courses are bioinformatics for computer scientists, introduction to scientific computing, numerical methods, parallel programming, time series analysis and stochastic processes, and machine learning or applied data analysis or applied machine learning. Students have to take two of these courses. The recommended core

classes are divided into two categories as core programming and core systems. Core programming courses are C and advanced programming, where core systems courses are compilers, introduction to computer systems, advanced computer systems, computer architecture, parallel programming, operating systems, and distributed systems. The remaining courses are recommended elective courses are advanced algorithms, cloud computing, C++ for advanced programmers, and advanced C++. To stay up with the quickly evolving world of technology, electives include cutting-edge courses in software engineering, machine learning, high performance computing, web development, cloud computing, big data analytics, application development, and information security. The computer science department provides math and programming preparatory courses for students without a background in those subjects to introduce them to computing and the fundamental and introductory abilities required to successfully begin masters-level education. Advanced backgrounded students can start in higher-level classes.

Scholars concur that students must now possess soft skills in areas like problem-solving, creative thinking, communication, teamwork, and marketing in order to succeed in ICT-focused courses (Passow, 2017). Yet, there have been allegations that opportunities for acquiring these abilities in their current forms are insufficiently provided by science and engineering education (Male, 2010). In this thesis, the courses are not analyzed in the way of whether the course is about soft skills or hard skills. In addition, the course type (compulsory, elective etc.) was not used in this analysis. All courses offered by the universities were included in the analysis. The gathered course data is saved as xls file as in Table 4. This was given as an example to show the text format of a course detail. For the American university, there is no information for what skills students should acquire after the completion of the course. Thanks to this developed methodology, ESCO skills and desired universities and courses can be compared.

Table 4: An example: The European university computer architecture course detail

Course Type	Course Name	Course Detail	After completion of the course the students should be able to
Compulsory	Computer architecture	architectural techniques essential for achieving high performance for application software. simulation-based analysis methods for quantitative assessment of the impact a certain architectural technique has on performance and power consumption. trends that affect the evolution of computer technology including Moore s law, metrics of performance (execution time versus throughput) and power consumption, benchmarking as well as fundamentals of computer performance such as Amdahl s law and locality of reference. how simulation based techniques can be used to quantitatively evaluate the impact of design principles on computer performance. various techniques for exploitation of instruction level parallelism (ILP) by defining key concepts for what ILP is and what limits it. dynamic and static techniques. Tomasulo s algorithm, branch prediction, and speculation. loop unrolling, software pipelining, trace scheduling, and predicated execution. memory hierarchies. attack the different sources of performance bottlenecks in the memory hierarchy such as techniques to reduce the miss rate, the miss penalty, and the hit time. victim caches,	master concepts and structures in modern computer architectures. understand the principles behind a modern microprocessor; advanced pipelining techniques that can execute multiple instructions in parallel in order to be able to establish performance of computer systems; understand the principles behind modern memory hierarchies in order to be able to assess performance of computer systems; and proficiency in quantitatively establishing the impact of architectural techniques on the performance of application software using state-of-the-art simulation tools.

In the rest of the “Implementation” section, the methodological steps used to extract the competences from ESCO database and MSc course syllabuses, and Natural Language Processing methodology are explained in detail to address the research objectives.

4.2 Data preparation

The NLP method often entails running a software pipeline with the intention of extracting data from text. The task of comparing the similarity of two texts is known as sentence similarity. Sentence similarity models take input texts and turn them into vectors (embeddings) that capture semantic data and determine how similar (near) they are to one another. Clustering and grouping as well as information retrieval benefit greatly from this assignment. Since, the skills datasets (ESCO and course syllabuses) have to be compared semantically, sentence similarity method is applied.

Before applying NLP, in order for this method to give a proper result, the data (sentences) that are not related to proficiency in the course sentences were cleaned manually (*see* Overall Research Design). Bullets, unrelated sentences, non-alphanumeric symbols were removed. This type of data cleaning was done manually because NLP and the training set utilized in NLP were not trained to filter or delete/eliminate such types of irrelevant sentences with ESCO skills. Finally, those manually cleaned texts were saved as two different csv files (one for ESCO and one for course detail).

To use NLP, JupyterLab server was used as a web-based interactive development environment to write Python. One of the Python libraries “spacy” was used for sentence detection to divide the text in

Table 3 and Table 4 (all course data for both universities) into linguistically meaningful parts. For each sentence found by spacy, full stop(.) was used as the sentence delimiter.

4.3 Data pre-processing for skills

The following operations for data pre-processing after sentence detection were performed: Convert all sentences to lower-case, remove all stop-words, convert final version of textual data into a csv file. Then, data from two different data sources has been exported to csv file for use when writing the code. Note that, in order to choose the most appropriate method, all versions of datasets were saved. Finally, two different datasets, which have a semantically comparable structure was obtained.

Since, tokenization, lemmatization, and named entity recognition (NER) techniques do not ensure semantic similarity for sentences, and the same subject was explained with different words in the course details and ESCO, these techniques were not preferred.

4.4 Sentence similarity detection

By using the csv file (output of Data pre-processing for skills), two different models were applied to compare datasets to catch the semantic similarity for sentences. In order to use whichever model gives more meaningful comparison results, the outputs of the two models were recorded and compared; en_core_web_lg spacy **BERT** model and spacy-universal-sentence-encoder **USE** model.

4.4.1 Using en_core_web_lg spacy BERT model

Spacy is a cutting-edge Python natural language library for text processing, and features POS tagging, NER (named entity recognition), word vectors and dependency parsing. It is particularly useful for obtaining relevant information from the text since

it makes it simple to determine the context of the text, and en_core_web_lg is the pipeline trained for English language (Data Science Learner, 2022).

Since, the ESCO has skill sentences for specialized in computer science such as;

- Oracle Application Development Framework
- Java (computer programming)
- Eclipse (integrated development environment software)
- COBOL

computer science related phrases and specialized names were added to dictionary in NLP pipeline. The components used in pipeline were shown in Table 5.

Table 5: Components added to en_core_web_lg pipeline for specialized terms

AttributeRuler	Set token attributes using matcher rules.
+ DependencyParser	Predict syntactic dependencies.
EditTreeLemmatizer	Predict base forms of words.
EntityLinker	Disambiguate named entities to nodes in a knowledge base.
+EntityRecognizer	Predict named entities, e.g. persons or products.
+EntityRuler	Add entity spans to the Doc using token-based rules or exact phrase matches.
Lemmatizer	Determine the base forms of words using rules and lookups.
Morphologizer	Predict morphological features and coarse-grained part-of-speech tags.
+SentenceRecognizer	Predict sentence boundaries.
+Sentencizer	Implement rule-based sentence boundary detection that doesn't require the dependency parse.
Tagger	Predict part-of-speech tags.
TextCategorizer	Predict categories or labels over the whole document.
Tok2Vec	Apply a "token-to-vector" model and set its outputs.
Tokenizer	Segment raw text and create Doc objects from the words.
TrainablePipe	Class that all trainable pipeline components inherit from.
Transformer	Use a transformer model and set its outputs.
+Other functions	Automatically apply something to the Doc, e.g. to merge spans of tokens.

In Python, the csv files created from ESCO skills and course details were used as an input for "data-frames". Firstly, the paragraph by paragraph similarity was analyzed, but this comparison showed that the similarity was very high (similarity scale is between 0 and 1 in this model). It was checked whether they were similar semantically by reading the paragraphs that seemed highly similar in results. It was observed that for paragraphs, including "computer science, programming, etc.", the similarity was very high because of the context of the paragraphs. Since this result did not contribute to the assessment, it was decided to make a sentence-based comparison by changing the method. Then, sentence by sentence similarity was analyzed, and the results

showed that this method resulted in more precise and meaningful comparison. The paragraph and sentence similarity results were presented below in Figure 13, Figure 14, Figure 15 as an example;

ESCO Data Scientist essential skills vs. technical writing course comparison results for "paragraph by paragraph" and "sentence by sentence"

------PARAGRAPH 1 Texts----------"

1 -- to develop the student's awareness of the underlying structure of scientific and engineering research papers .

2 -- improve proficiency in reviewing and writing scientific research papers .

3 -- presenting such papers in public.

4 -- ethical issues in scientific writing, plagiarism and authorship.

------PARAGRAPH 1 ENTITIES ----------"

Figure 13: Data scientist essential skills vs. the European university's HPC MSc programme's technical writing course paragraph comparison

------PARAGRAPH 2 Texts ----------"

1 --> develop data processing applications.
2 --> evaluate research activities.
3 --> draft scientific or academic papers and technical documentation.
4 --> professional network with researchers and scientists.
5 --> design database scheme.
6 --> execute analytical mathematical calculations.
7 --> integrate gender dimension in research.
8 --> perform project management.
9 --> collect ict data.
10 --> manage data collection systems.
11 --> implement data quality processes.
12 --> manage personal professional development.
13 --> synthesize information.
14 --> use databases.
15 --> conduct research across disciplines.
16 --> apply for research funding.
17 --> perform scientific research.
18 --> disseminate results to the scientific community.
19 --> establish data processes.
20 --> interpret current data.
21 --> manage open publications.
22 --> perform data cleaning.
24 --> write scientific publications.
25 --> communicate with non-scientific audience.
26 --> demonstrate disciplinary expertise.
27 --> build recommender systems.
28 --> manage intellectual property rights.
29 --> mentor individuals.
30 --> handle data samples.
31 --> deliver visual presentation of data.
32 --> increase the impact of science on policy and society.
33 --> publish academic research.
34 --> promote the transfer of knowledge.
35 --> use data processing techniques.
36 --> promote the participation of citizens in scientific and research activities.
37 --> normalize data.
38 --> manage findable accessible interoperable and usable data.
39 --> think abstractly.
40 --> manage research data.
41 --> interact professionally in research and professional environments.
42 --> report analysis results.
43 --> apply research ethics and scientific integrity principles in research activities.
44 --> promote open innovation in research.
45 --> operate open source software

------PARAGRAPH 2 ENTITIES ----------"
ict DEFINED_ENTITY

------PARAGRAPH by PARAGRAPH TEXT COMPARISON----------"
0.31754445639085255

Figure 14: ESCO data scientist occupation- essential skills text converted to a paragraph

1 . SENTENCE in course: to develop the student's awareness of the underlying structure of scientific and engineering research papers .

---

1 <-> 1 -----> 0.061892032623291016 develop data processing applications.
1 <-> 2 -----> 0.14873246848583221 evaluate research activities.
1 <-> 3 -----> 0.38997045159339905 draft scientific or academic papers and technical documentation.
1 <-> 4 -----> 0.26266714930534363 professional network with researchers and scientists.
1 <-> 5 -----> 0.0891144871711731 design database scheme.
1 <-> 6 -----> 0.12771859765052795 execute analytical mathematical calculations.
1 <-> 7 -----> 0.161333829164505 integrate gender dimension in research.
1 <-> 8 -----> 0.066912941634655 perform project management.
1 <-> 9 -----> 0.06128904968500137 collect ict data.
1 <-> 10 -----> -0.01885901391506195 manage data collection systems.
1 <-> 11 -----> 0.008682533167302608 implement data quality processes.
1 <-> 12 -----> -0.012963822111487389 manage personal professional development.
1 <-> 13 -----> 0.05239705368876457 synthesize information.
1 <-> 14 -----> 0.05003416910767555 use databases.
1 <-> 15 -----> 0.24776259064674377 conduct research across disciplines.
1 <-> 16 -----> 0.2647782862186432 apply for research funding.
1 <-> 17 -----> 0.24023477733135223 perform scientific research.
1 <-> 18 -----> 0.271527498960495 disseminate results to the scientific community.
1 <-> 19 -----> 0.05266067758202553 establish data processes.
1 <-> 20 -----> -0.013596970587968826 interpret current data.
1 <-> 21 -----> 0.04246843606233597 manage open publications.
1 <-> 22 -----> -0.08239012211561203 perform data cleaning.
1 <-> 23 -----> -0.06946136057376862 speak different languages.
1 <-> 24 -----> 0.33158573508262634 write scientific publications.
1 <-> 25 -----> 0.12129399180412292 communicate with non-scientific audience.
1 <-> 26 -----> -0.04266699030995369 demonstrate disciplinary expertise.
1 <-> 27 -----> 0.16183260083198547 build recommender systems.
1 <-> 28 -----> -0.027911566197872162 manage intellectual property rights.
1 <-> 29 -----> -0.02692476473748684 mentor individuals.
1 <-> 30 -----> -0.007281729951500893 handle data samples.
1 <-> 31 -----> 0.05934295058250427 deliver visual presentation of data.
1 <-> 32 -----> 0.30259621143341064 increase the impact of science on policy and society.
1 <-> 33 -----> 0.35734719038009644 publish academic research.
1 <-> 34 -----> 0.18880994617938995 promote the transfer of knowledge.
1 <-> 35 -----> 0.006050878204405308 use data processing techniques.
1 <-> 36 -----> 0.3972088098526001 promote the participation of citizens in scientific and research activities.
1 <-> 37 -----> -0.09620340913534164 normalize data.
1 <-> 38 -----> -0.007627774029970169 manage findable accessible interoperable and usable data.
1 <-> 39 -----> 0.037750255316495895 think abstractly.
1 <-> 40 -----> 0.1535622626543045 manage research data.
1 <-> 41 -----> 0.17553360760211945 interact professionally in research and professional environments.
1 <-> 42 -----> 0.07336544990539551 report analysis results.

Figure 15: First sentence of technical writing course of the European university compared with all essential skills in data scientist occupation of ESCO

4.4.2 Using spacy-universal-sentence-encoder USE model

In order to determine sentence similarities, the Universal Sentence Encoder was trained on a variety of tasks. One of the key objectives that the USE was trained on was to find "semantic textual similarity" (STS) between sentence pairs. In the literature BERT or NER models were used for sentence similarity in general. In this case, while changing the model in NLP, (Floydhub, 2022) was used as a reference for choosing the right model. There is a comparison chart (shown in Figure 16) for BERT and USE. They presented the cosine similarity results of the sentences compared in those models.

	The Sentence	The Compared Sentence	BERT Score	USE Score
1	Blah Blah	Does this integrate with gmail?	0.765	0.519
2	I really do not like this product	I really like this product	0.865	0.747
3	How do I change my password?	I can't find the settings page	0.868	0.671
4	How much will this cost?	Is this expensive?	0.892	0.803

Figure 16: BERT and USE, cosine similarity results comparison- adapted from (Floydhub, 2022)

NLP was continued with the USE model, as these results showed that using the USE model, not the BERT model, gave more accurate results according to semantic textual similarity.

The USE model was applied with the csv. file (the cleaned and last saved, described at the beginning of Section 4.3). the results were saved and compared with en_core_web_lg spacy model. It was decided to change data cleaning model, as not to use stop-words filtering. Since, catching the semantic textual similarity is the main purpose, it was decided to continue without this filtering, since it was thought that the state without stop words caused a change in meaning.

n→n [ESCO skills sentences→University→ course sentences] similarity comparison for each sentence pairs was done by using the USE model (similarity scale is between -1 and 1 in this model). After applying the model, the results including cosine similarity results was saved in a csv file format. Finally, to define the cut off value

descriptive analysis was used, and the hit number (how many similarities occurred in model with a defined cut off value) was counted for the courses in the MSc program with compared to ESCO skills.

The USE model uses [-1, 1] cosine similarity where;

- -1 indicates that two vectors are strongly opposite,
- 0 for independent vectors,
- 1 indicates that two vectors are strongly similar.
- Cosine similarity is defined as "*a metric which measures the cosine of the angle between two vectors projected in a multi-dimensional space*", that means the smaller the angle, the more similar they are, shown in Figure 17.

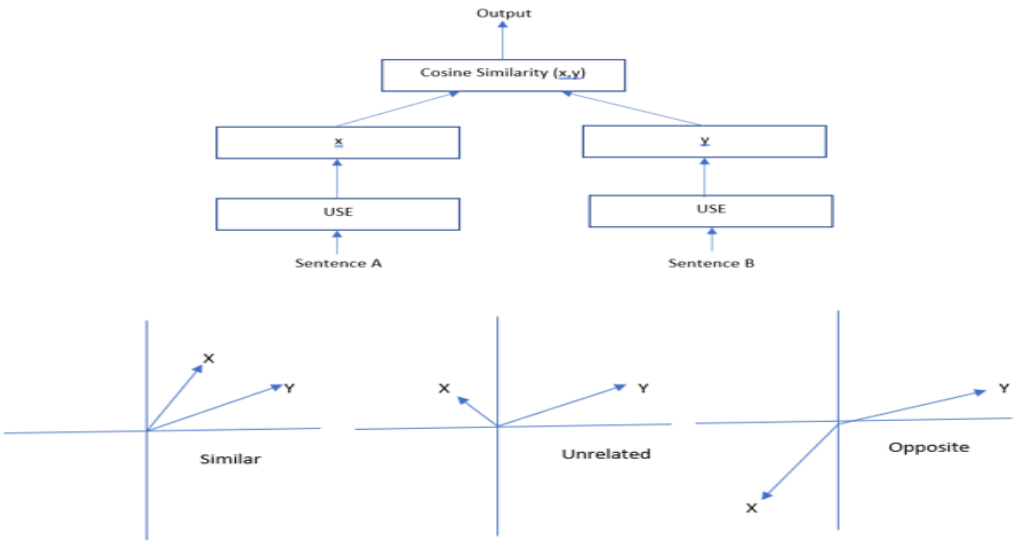

Figure 17: USE model cosine similarity scores measurement

In this analysis, the number of combinations for an n-to-n sentence comparison is 146124 for the *European university's High Performance Computing MSc program* because there are 297 skill sentences in 4 occupations (including "*Essential skills, Essential Knowledge, Optional Skills and Competences, and Optional Knowledge*") and 492 course sentences in 29 courses to evaluate, and for the American university,

a total of 82269 comparisons were made via 297 skill sentences in ESCO and 277 course sentences in MSc program.

After obtaining the similarity results, the question of "what would be the cut off value?" came to the fore. To set a cut off value within similarity values descriptive statistic was used for each occupation and university pair;

- **The results were ordered as ascending**
- **The list was cut into four equal parts**
- **The quartiles were defined as "cut off" values**

where;

- 25% of observations ≤ lower quartile, Q1,
- Middle quartile is the median, Q 2,
- 25% of observations ≥ upper quartile, Q3

shown in

Figure 18 and Figure 19 as an example for the European and American universities, respectively.

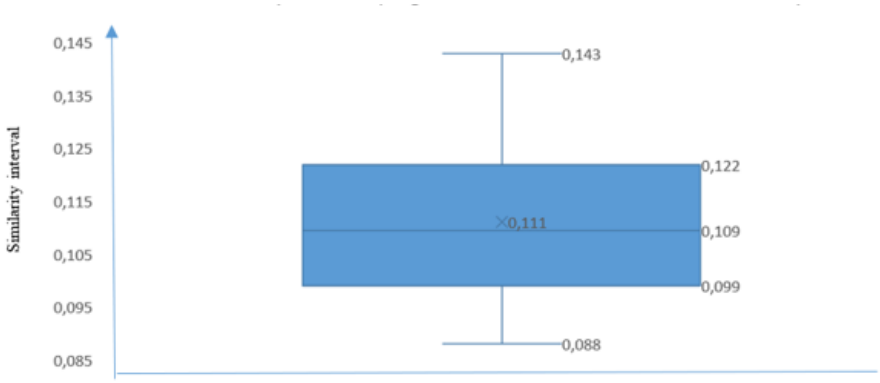

Figure 18: The European university's HPC MSc program vs. ESCO Data Scientist occupation- example of calculated quartile values

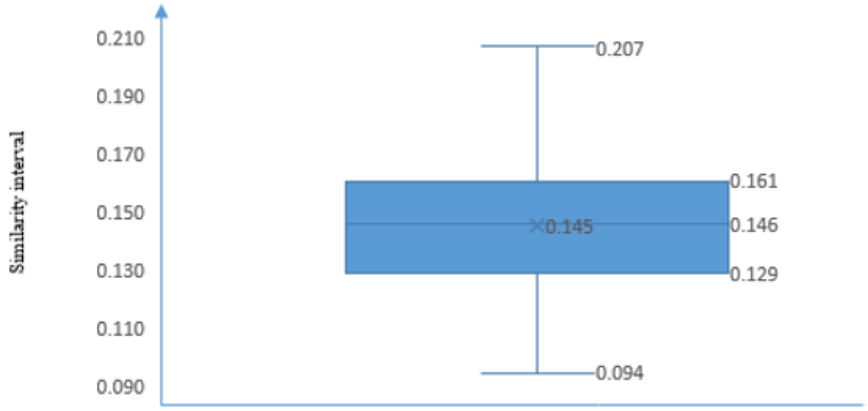

Figure 19: The American university's HPC MSc program vs. ESCO System Architect occupation- example of calculated quartile values

According to the results,

- ✓ **Lower quartile value (Q1)** was chosen, **and the hit values > Q1 were accepted as "similar".**

4.5 Chapter Summary

The details on ESCO skills/competencies that were included in the course descriptions were highlighted. This made it possible to compare both in a more precise way, which made it easier to quantify and analyze the differences in each skill set. NLP model was created exclusively for English language. With some slight changes, the approach can also be used in other languages.

The datasets were manually created that included details of MSc programs on HPC that were offered at universities in the academic year 2021–2022. Only MSc programs with the term "high performance computing" in the title were selected, and the necessary data was gathered from the universities' websites.

For semantic similarity analysis USE model was preferred after comparing it with the BERT model. The results of USE model were shown with the cut off value defined by descriptive analysis. The results are discussed in the following sections.

CHAPTER 5

5 RESULTS AND DISCUSSIONS

5.1 A methodology to determine skills gaps

In order to evaluate and visually present the results of the study; the averages of the positive vector values (smaller than zero and equal to zero values were eliminated) showing the relationships between the courses at the European and American universities and the occupations determined were taken on the basis of courses, and heat maps were created by drawing box plots and histogram graphs over these averages. Then, comments on the current situation were made on these graphs and heat maps. In this context, visual information and comments about the relations between the courses taught in universities and the occupations determined are in the following sections.

Although the course contents were defined in the sentence structure, these sentences were analyzed in order to obtain more meaningful results, and the sentences were arranged and structured as described in Section 3.2 and in more detail in Section 4.2. In order for the written code to produce more accurate results, tables were prepared from the course sentences of weekly course programs. These tables can be found in Appendix A. Examples of these tables are presented for the European and American universities in Table 6 and
Table 7, respectively.

Table 6: The European university HPC MSc courses - rewritten sentences for NLP

Course	DETAIL
Computer architecture	architectural techniques essential for achieving high performance for application software. simulation-based analysis methods for quantitative assessment of the impact a certain architectural technique has on performance and power consumption. trends that affect the evolution of computer technology including Moore s law, metrics of performance (execution time versus throughput) and power consumption, benchmarking as well as fundamentals of computer performance such as Amdahl s law and locality of reference. how simulation based techniques can be used to quantitatively evaluate the impact of design principles on computer performance. various techniques for exploitation of instruction level parallelism (ILP) by defining key concepts for what ILP is and what limits it. dynamic and static techniques. Tomasulo s algorithm, branch prediction, and speculation. loop unrolling, software pipelining, trace scheduling, and predicated execution. memory hierarchies. attack the different sources of performance bottlenecks in the memory hierarchy such as techniques to reduce the miss rate, the miss penalty, and the hit time. victim caches, lockup-free caches, prefetching, virtually addressed caches. main memory technology is covered in this part. multicore, multithreaded architectures. programming model and how processor cores on a chip can communicate with each other through a shared address space. micro architecture level it deals with

Table 7: The American university HPC MSc courses - rewritten sentences for NLP

Course	DETAIL
High Performance Computing	Overview of CPU and GPU Architectures. Instruction sets. Functional units. Memory hierarchies. Performance Metrics. Latency and bandwidth. Roofline modeling. Single-core optimization. Compiler-assisted vectorization. data-level parallelism. Design patterns for cache-based optimization. Multi-threaded CPU programming. Worksharing, synchronization, and atomic operations. Memory access patterns, including non-uniform memory access. The OpenMP API. GPU programming. Thread-mapping for optimal vectorization and memory access. Task-scheduling for latency reduction. The CUDA and OpenMP offload APIs. Distributed parallelism. Synchronous and asynchronous communication patterns. Data decomposition. Hybrid models for distributed multi-threaded and GPU programming. The MPI API.
Bioinformatics for Computer Scientists	Genomics, Bioinformatics and Molecular Biology. A high-level view of increasingly important role of computing in the biological sciences will be presented. Genomes, Sequences and Databases. A survey of the current state of the art in storing, organizing and analyzing large data sets will be discussed. The advantages and disadvantages of these methods will be explored in the context of academic and commercial research initiatives. Sequence Alignment. Fast, reliable alignment of text strings started the bioinformatics revolution. Protein Structure and Function. Spatial assembly and interactions of proteins support life and cause of disease. Protein Motifs and Modeling. Understanding protein function holds the promise developing therapeutics and curing diseases, but the computational complexity of analyzing three-dimensional models presents obstacles that have been difficult to overcome. shape analysis and comparison that can be scaled to large data sets. High-Performance Computing for Bioinformatics. Strategies for conducting large-scale analysis of genes and proteins will be presented. Microarray Data Analysis. The technologies used to power these services will be introduced as well as the different approaches used to provide web services to analyze the data. SNPs and Disease. trace the genetic origin of disease. different approaches to cataloging and analyzing these changes. In Silico Drug Discovery. Approaches to using computer models to develop new drugs will be presented.

5.1.1 The European and American universities' HPC MSc programme overview

In order to see the strength of the relationship vector between the courses taught in universities and the occupations in a holistic manner, the intensity of the relationship was determined from the darkest shade of green to the darkest shade of red. In this context, the most intense relationship is indicated in the darkest shade of green, and the least relationship is indicated in the darkest shade of red. Thanks to these heat maps, the courses in which the relationship intensifies or decreases can be seen in general.

When the heat maps are examined in general, it is seen from the Figure 20 that the Algorithms course is the most related course among the courses taught at the European university, and the Master's Thesis course is the least related course. 5-point Likert Scale was used to show the similarity level.

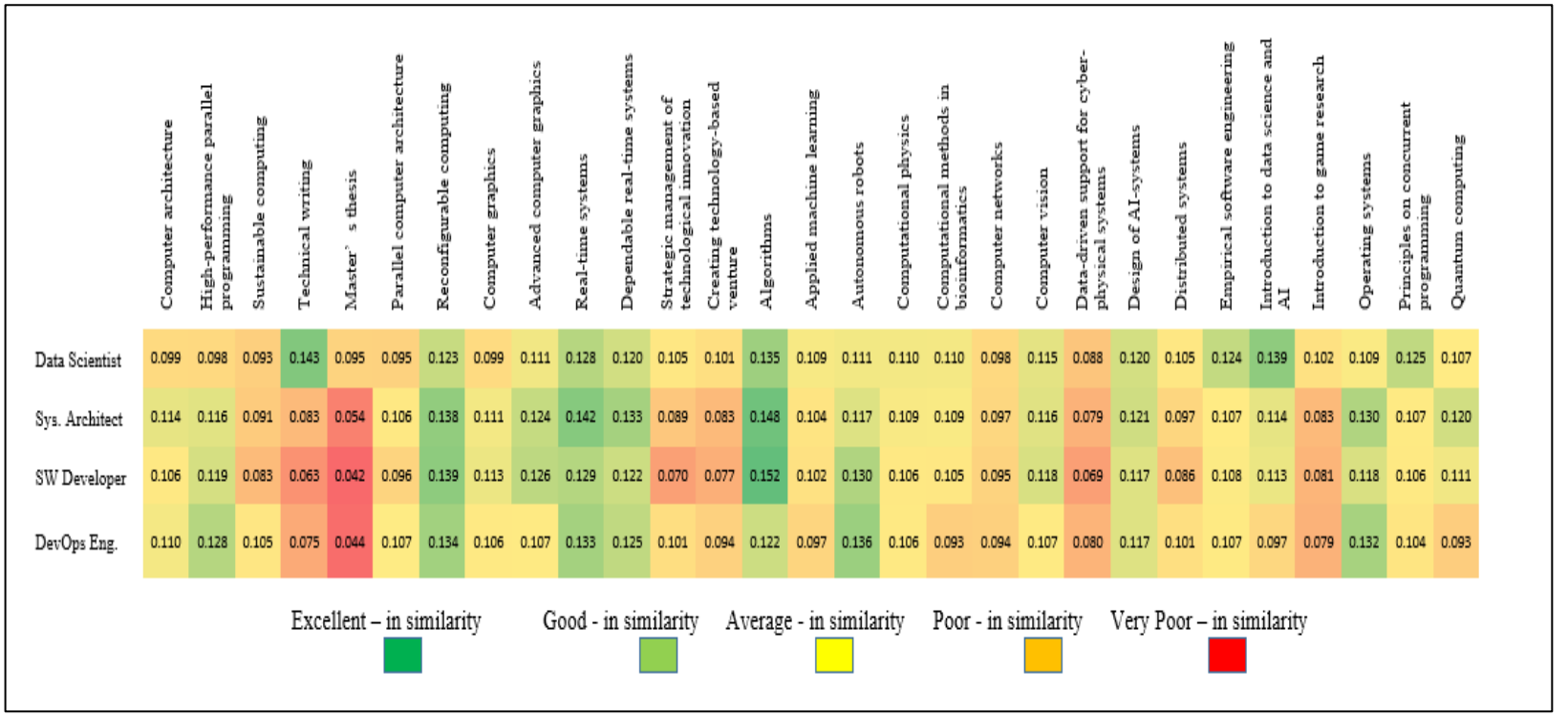

Figure 20: Heatmap for overview – The European university

Reconfigurable computing, real-time systems, dependable real-time systems have closer similarity results for all occupations because in their course definitions similar words, definitions were used to describe the courses.

When the ESCO definitions were examined, it was obvious to see the skills defined in System Architect occupation showed similarity with skills defined in Software Developer occupation.

Similarly, in the general examination, it is seen from the Figure 21 that the Introduction to Scientific Computing course is the most related course and the Machine Learning course is the least related course among the courses taught at the American university.

Figure 21: Heatmap for overview – The American university

The results showed that, similar definitions in course contents result in closer similarity vectors. C programming and advanced C programming courses were defined using the same words and semantically expressing the same things, so their similarity vectors for each occupation were very high. The same conclusion can be reached for C++ for advanced programmers and advanced C++ courses.

5.2 Identification of the skills gaps and coverage of the industry in the MSc education

Educational institutions should take the lead in managing technological developments. On the one hand, they are expected to provide educational and training opportunities to fulfill the demands of students and the labor market at the time; on the other hand, they must be adaptable in order to effectively modify their programs to consider future technological developments. It is critical to include digital knowledge like high performance computing in new and updated curricula to enable the development of skills that are appropriate for future workers.

Text mining or natural language processing techniques has been widely utilized by academics to examine job descriptions, educational course descriptions to show the current gaps between market and academia. ICT and digital domains have been the main topics of most studies. All of these researches started by gathering and analyzing data from online sources using cutting-edge machine learning and text mining techniques. For some studies, websites with job advertisements are the primary sources of information for the market's demands. These studies' main goals are the examination of market demands, the automatic updating of occupation taxonomies, and the comparison of market needs and educational institution curriculums. Regarding the approaches utilized, the majority of studies have trained a classification algorithm that extracts data from documents on job profiles, skills, and expertise using machine learning. To calculate semantic similarity, various methods have been used in the literature. The outstanding examples are (Almgerbi, 2021) for ICT domain, (Chiarello, 2021) for Industrial Engineering domain, (Maer-Matei, 2019) for digital domain.

Using the most recent machine learning and text mining techniques, these research all started by gathering and analyzing data from web sources.

This thesis aimed to present a novel way to compare the curriculum of selected universities' HPC MSc programs with the ESCO abilities outlined under the chosen occupations; data scientist, system architect, software developer, and devOps engineer by using natural language processing techniques.

5.2.1 A European University

In this thesis, one of the research objectives was to recommend a methodology to determine skills gaps. The work carried out for this purpose were explained in detail in the Chapter 3 METHODOLOGY And CHAPTER 4 IMPLEMENTATION. After determining the methodology to be developed, the box plot in the Figure 22 was drawn using the similarity vectors obtained as output of the written code. The courses of the European university's HPC MSc program were compared with the ESCO occupations - a comparison was made in the skills defined for these occupations in ESCO and the sentences in the weekly curriculum of the courses were visualized with this box plot.

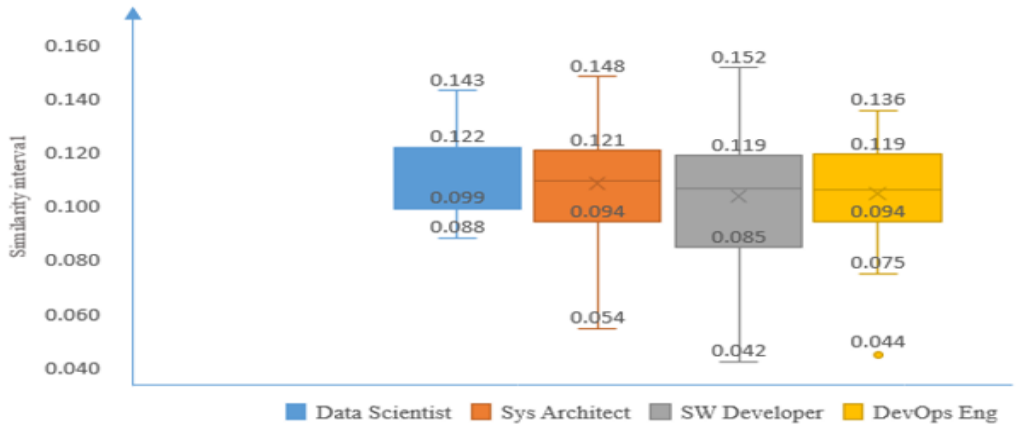

Figure 22: The European university's HPC MSc courses vs. ESCO occupations - similarity coverage

Looking at the distribution on the graph, it is seen that the HPC courses provided by this European university meet the 4 occupations selected from ESCO at more or less the same level. When the skills defined under the data scientist, software developer, system architect and devOps engineer occupations as defined in the weekly schedules of HPC courses are compared, the accumulation of the distribution in approximately the same range shows that the general view of this HPC program is consistent and stable.

This result demonstrates how well the semantic similarity method can generalize and identify concepts that are similar, but expressed differently. However, it should not be overlooked that HPC domain specific occupations and skills are not included in ESCO. If there were more appropriate occupations at ESCO that are HPC domain specific, this analysis might yield better results in terms of accuracy and completeness.

Here the results showed that courses offered at this European university's HPC MSc programme provide more or less the same degree of knowledge and skills for the 4 chosen occupations.

5.2.1.1 The European University's HPC MSc programme vs. Data Scientist occupation

According to the box plot drawn based on similarity results, the relationship between the courses at the European university's HPC MSc programme and the Data Scientist occupation; it is seen that the highest correlation is 0.143 for the Technical Writing course, and the lowest value is 0.088 for the Data-Driven Support for Cyber-Physical Systems course. In addition, it is seen that the average value of the relationship with this occupation is 0.111, the median value is 0.109, the Q1 value accepted as the lower limit of the concentration based on the general average is 0.099, and the Q3 value considered as the upper limit value is 0.122.

In section 4.4.2, the cut off value was determined as Quartile 1 (Q1), so for the comparison of this HPC MSc programme course sentences and ESCO Data Scientist

occupation sentences, the similarity vectors that are above **Q1= 0.099** were accepted as similar.

5.2.1.2 The European University's HPC MSc programme vs. System Architect occupation

According to the box plot drawn based on similarity results, the relationship between the courses at European university's HPC MSc programme and the System Architect occupation; it is seen that the highest correlation is 0.149 for the Algorithms course, and the lowest value is 0.054 for the Master's Thesis course. In addition, it is seen that the average value of the relationship with this occupation is 0.108, the median value is 0.109, the Q1 value, which is considered as the lower limit of the concentration depending on the general average, is 0.094, and the Q3 value, which is considered as the upper limit value, is 0.121. So, the similarity vectors that are above **Q1= 0.094** were accepted as similar.

5.2.1.3 The European University's HPC MSc programme vs. Software Developer occupation

According to the box plot drawn based on similarity results, the relationship between the courses at European university's HPC MSc programme and the Software Developer occupation; it is seen that the highest correlation is 0.152 for the Algorithms course, and the lowest value is 0.042 for the Master's Thesis course. In addition, it is seen that the average value of the relationship with this occupation is 0.104, the median value is 0.107, the Q1 value accepted as the lower limit of the concentration depending on the general average is 0.085, and the Q3 value considered as the upper limit value is 0.119. So, the similarity vectors that are above **Q1= 0.085** were accepted as similar.

5.2.1.4 The European University's HPC MSc programme vs. DevOps Engineer occupation

According to the box plot based on similarity results, the relationship between the courses at European university's HPC MSc programme and the DevOps Engineer occupation; it is seen that the highest correlation is 0.136 for the Autonomous Robots course, and the lowest value is 0.044 as an outlier for the Master's Thesis course. Since the lower outlier threshold (threshold) is 0.056, the value of 0.044 in this graph is considered an outlier. However, despite the outlier, it is seen that the distribution is normal. In addition, the mean value of the relationship with this occupation is 0.105, the median value is 0.106, the lowest non-outlier value is 0.075, the lower limit of the concentration depending on the general average is 0.094 (Q1 value), and the upper limit value is 0,112 (Q3 value). So, the similarity vectors that are above **Q1= 0.094** were accepted as similar.

5.2.2 An American University

The courses of the chosen American university's HPC MSc program were compared with the ESCO occupations. A comparison was made in the skills defined for these occupations in ESCO and the sentences in the weekly curriculum of the courses were visualized with box plot in Figure 23.

Figure 23: The American university's HPC MSc courses vs. ESCO occupations - similarity coverage

When the distribution of results for this university was examined, it is seen that the distribution is not consistent. The average of the results, the lower and upper limit values differ between occupations. Here the following assumptions can be made;

- it covers more skills specified in ESCO for software developer occupation,
- for devOps engineer and data scientist occupations, it covers the skills specified in the ESCO less,
- ESCO occupation definitions are not in line with MSc courses descriptions and contents,
- course definitions are not detailed enough and there are no definitions for skills and knowledge that students should be able to after completion of the course,
- there is no standard for defining courses in course descriptions and course contents in universities and also in ESCO, so the compared sentences are really different from each other.

5.2.2.1 The American University's HPC MSc programme vs. Data Scientist occupation

According to the box plot drawn in section 5.2.2, it is seen that the highest correlation is 0.193 as an outlier for the Introduction to Scientific Computing course, and the lowest value is 0.105 for the C++ for Advanced Programmers course. Since the upper outlier threshold (threshold) is 0.160, the value of 0.193 is considered as an outlier. Due to this outlier, it is seen that the distribution is not normal. In addition, the average value of the relationship with this occupation is 0.128, the median value is 0.126, the lower limit of the concentration depending on the general average is 0.115 (Q1 value), the upper limit value is 0.114, and the highest non-outlier value is 0.156. So, the similarity vectors that are above **Q1= 0.115** were accepted as similar.

5.2.2.2 The American University's HPC MSc programme vs. System Architect occupation

The highest correlation with system architect occupation is 0.207 for the Introduction to Scientific Computing course, and the lowest value is 0.094 calculated for the Machine Learning course. In addition, it is seen that the average value of the relationship with this occupation is 0.145, the median value is 0.146, considered as the lower limit of the concentration depending on the general average is Q1 value is 0.129,

and the Q3 value considered as the upper limit value is 0.161. So, the similarity vectors that are above **Q1= 0.129 were accepted as similar.**

5.2.2.3 The American University's HPC MSc programme vs. Software Developer occupation

The highest correlation is 0.208 for the Introduction to Scientific Computing course, and the lowest value is 0.097 for the Bioinformatics for Computer Scientists course. In addition, it is seen that the average value of the relationship with this occupation is 0.146, the median value is 0.147, the Q1 value which is considered as the lower limit of the concentration depending on the general average is 0.120, and the Q3 value which is considered as the upper limit value is 0.174. So, the similarity vectors that are above **Q1= 0.120 were accepted as similar.**

5.2.2.4 The American University's HPC MSc programme vs. DevOps Engineer occupation

The highest correlation is 0.171 for the Introduction to Scientific Computing course, and the lowest value is 0.076 for the Machine Learning course. In addition, it is seen that the average value of the relationship with this occupation is 0.127, the median value is 0.129, the Q1 value accepted as the lower limit of the concentration based on the general average is 0.113, and the Q3 value considered as the upper limit value is 0.143. So, the similarity vectors that are above **Q1= 0.113 were accepted as similar.**

5.3 Comparison of the ESCO skills with university curriculums

It was tried to investigate how the ESCO skills of *"Data Scientist, System Architecture, Software Developer, and DevOps Engineer"* occupations and the curricula of universities overlap with each other. The semantic similarity method – USE Model and a pair-wise comparison was used to compare ESCO skills' sentences and HPC MSc course contents. The output of NLP provided n-dimensional vectors calculated from cosine of the angle between two sentences; smaller cosine angles meant more

similar. To determine the cut off value, sentences were clustered with a cosine similarity greater than Q1 value for each occupation. The final set obtained (greater than Q1);

- Within the European university's HPC MSc course sentences compared with;
    - Data Scientist occupation; there were 15161 similarities,
    - System Architect occupation; there were 17945 similarities,
    - Software Developer occupation; there were 21379 similarities,
    - DevOps Engineer occupation; there were 6559 similarities,
- Within the American university's HPC MSc course sentences compared with;
    - Data Scientist occupation; there were 9018 similarities,
    - System Architect occupation; there were 11100 similarities,
    - Software Developer occupation; there were 13471 similarities,
    - DevOps Engineer occupation; there were 4029 similarities

In order to measure whether and how much the courses are aligned with each ESCO skill, the bar graphs shown below were created on the basis of occupations.

In the Figure 24, it is seen that the most overlapping course with the skills of the data scientist occupation defined in ESCO is the introduction to data science and AI course in the European university. This result can be considered as proof that the NLP method gave proper and reliable results. This figure showed that the courses with more than 50 percent similarity are in the HPC domain for data scientist occupation. However, it should be evaluated here without ignoring that ESCO has not yet included specific occupations in the HPC domain.

The least overlapping course is seen as Master's Thesis course in the European university's HPC MSc program with compared to all occupations. This course was defined with only one sentence which is "thesis work in an industrial context or within a research group at this programme". Since there is not a sentence with sufficient detail to make the comparison properly, it is at the bottom of the bar graphs as the course that has the least semantic similarity with respect to skills of occupations.

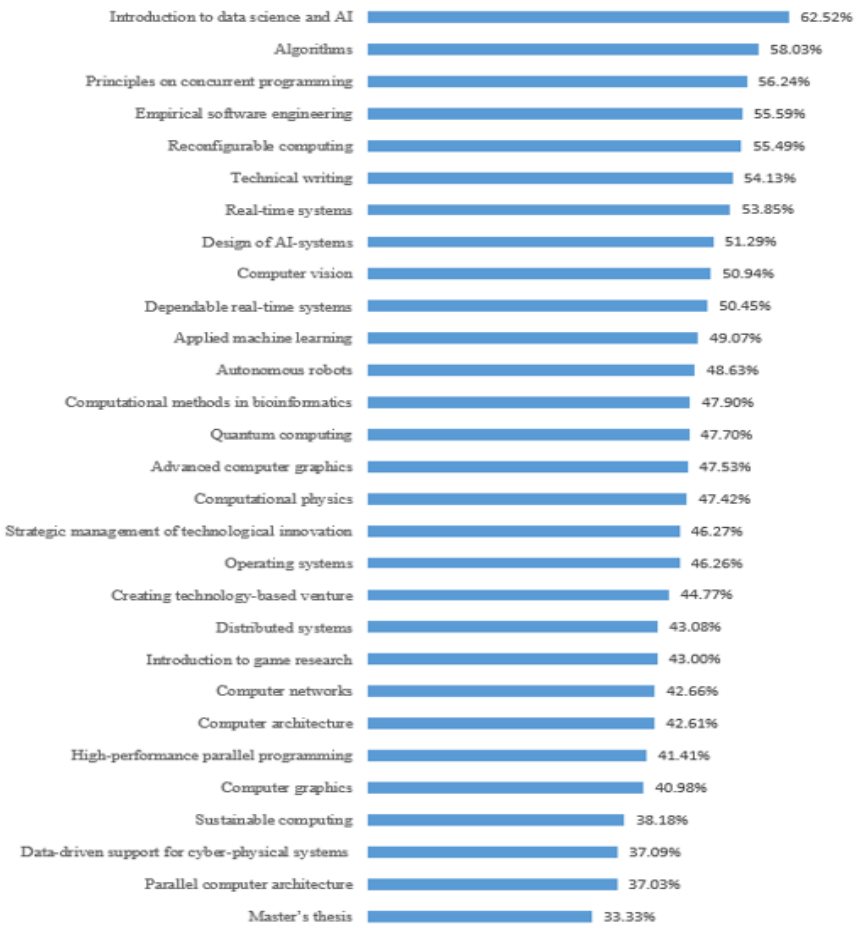

Figure 24: The European university's HPC MSc courses coverage of ESCO skills for Data Scientist occupation – in percentage.

In Figure 25, System Architect occupation was evaluated. This figure proved that the courses at the top of the graph are the courses in the HPC domain for system architecture occupation. It is seen that the courses in the middle part of the graph are general courses in the basic software engineering domain. Considering that the people who will be included in the HPC master's program will not only be from the computer or software engineering domains, it seems meaningful that basic courses are also given in this master's program.

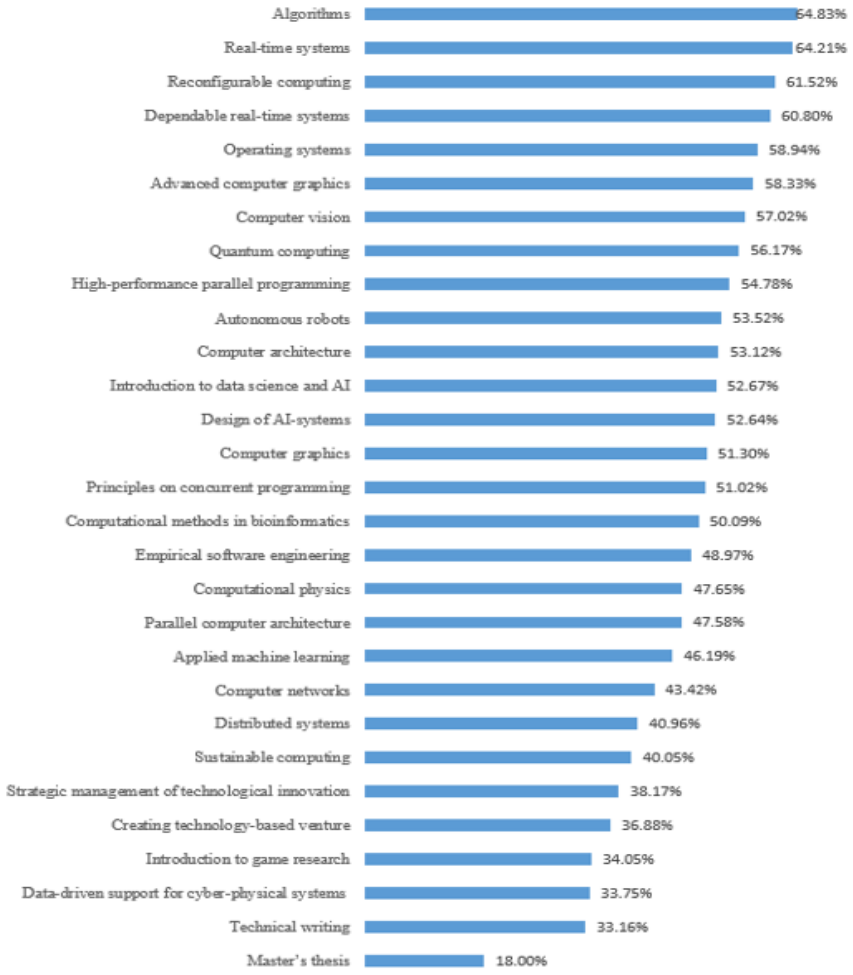

Figure 25: The European university's HPC MSc courses coverage of ESCO skills for System Architect occupation – in percentage

Figure 26, as a result of the comparison made with the software developer occupation, is a figure that states that the highest rate of similarity is usually in the courses that contain coding. In ESCO, Software Developer occupation is also named as "application developer, application programmer, solutions developer, programmer, and software engineer", and the essential and optional skills of these occupations include software engineering basic skills in general like *"computer programming, object-oriented programming, software design patterns, query languages"*. So, the results proved that if ESCO had included HPC domain specific occupations and the skills required for them, the comparison would have produced more accurate results.

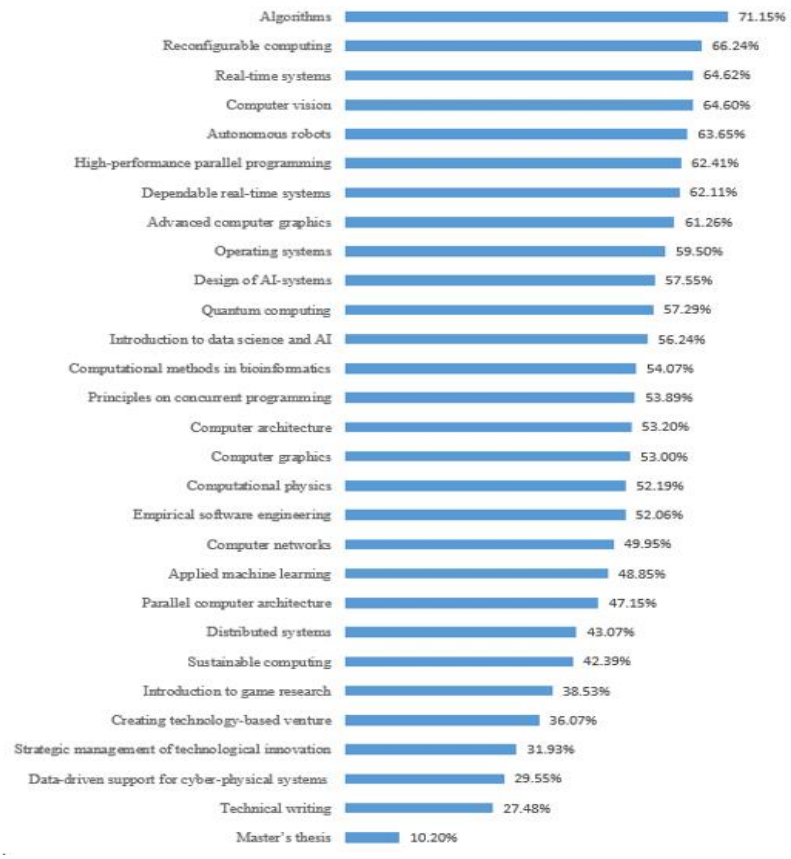

Figure 26: The European university's HPC MSc courses coverage of ESCO skills for Software Developer occupation – in percentage

In ESCO, there is no occupation called "DevOps Engineer". The existing occupation is "ICT Change and Configuration Manager", and the alternative label for this occupation is devOps engineer. So, the essential and optional skills required for devOps engineer includes mainly skills for ICT change and configuration manager that are;

- manage ICT virtualization environments,
- integrate system components,
- use scripting programming,
- deploy ICT systems,
- manage changes in ICT system,
- develop automated migration methods,

- DevOps,
- design cloud architecture,
- utilize computer-aided software engineering tools etc.

Optional knowledge defined in ESCO for devOps engineer;

- embedded systems,
- integrated development environment software,
- computer programming etc.

According to given information above, it is understandable from Figure 27, most of the similarity was provided by the optional knowledge part of ESCO.

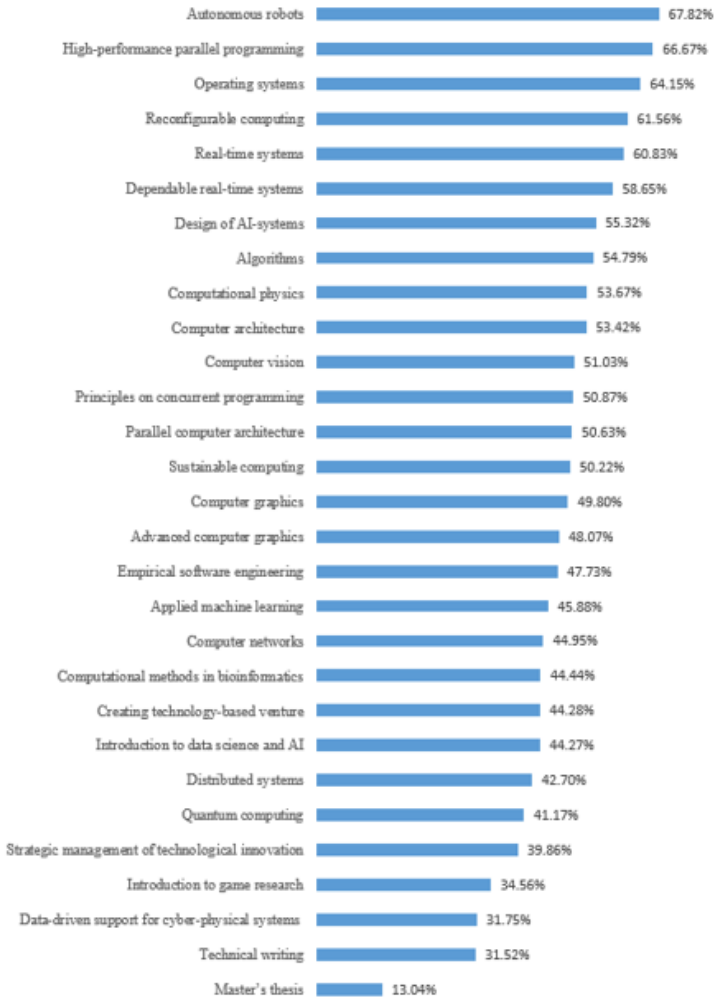

Figure 27: The European university's HPC MSc courses coverage of ESCO skills for DevOps Engineer occupation – in percentage

With the help of Figure 28, Figure 29, Figure 30 and Figure 31, it is seen that in all four occupations, introduction to scientific computing, introduction to computer systems, and programming courses have high similarity percentages, whereas the least similarity percentages are in machine learning and bioinformatics for computer scientists courses.

The remarkable detail here is that the *advanced C++* and the *C++ for advanced programmers* courses have quite different similarity values from each other for each occupation. This is because the two courses are defined quite differently from each other. Another surprising detail is that the machine learning course appears to be the

course with the least similarity. From this, it can be concluded that machine learning skills are not included in the selected occupations.

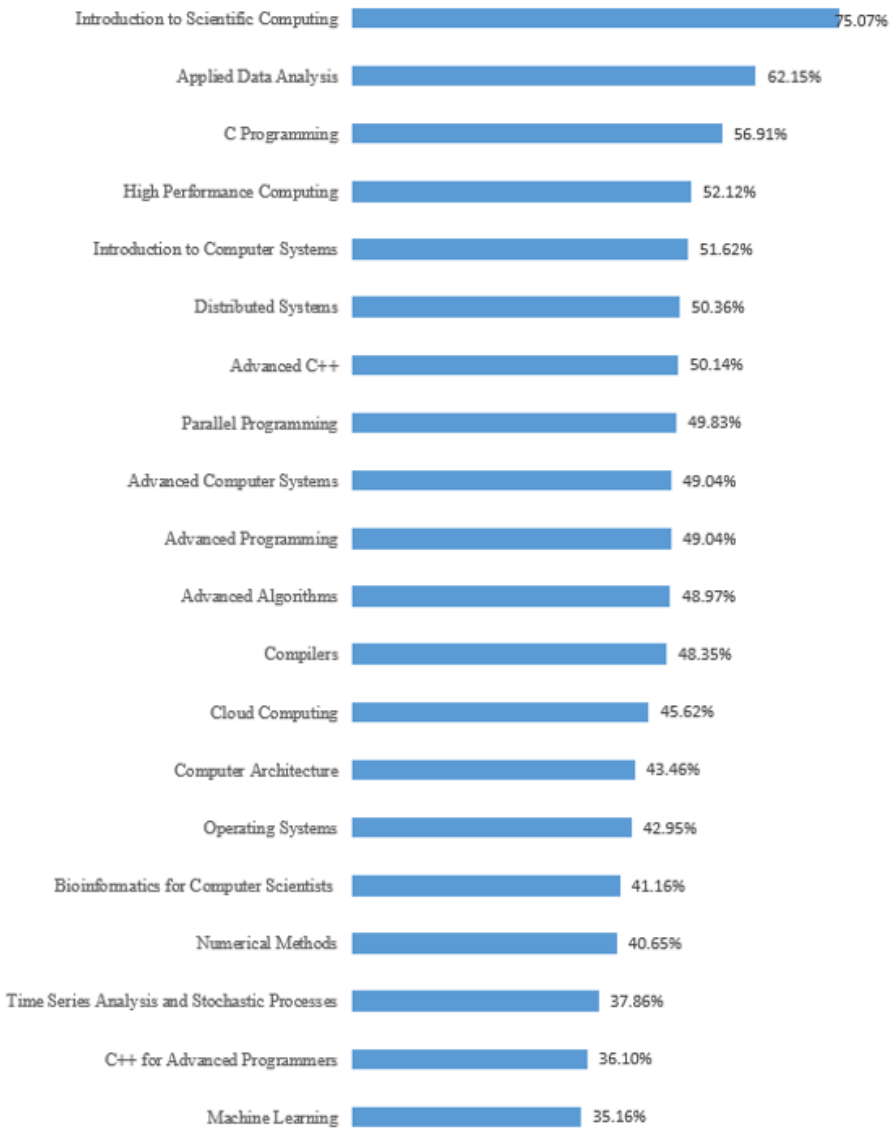

Figure 28: The American university's HPC MSc courses coverage of ESCO skills for Data Scientist occupation – in percentage

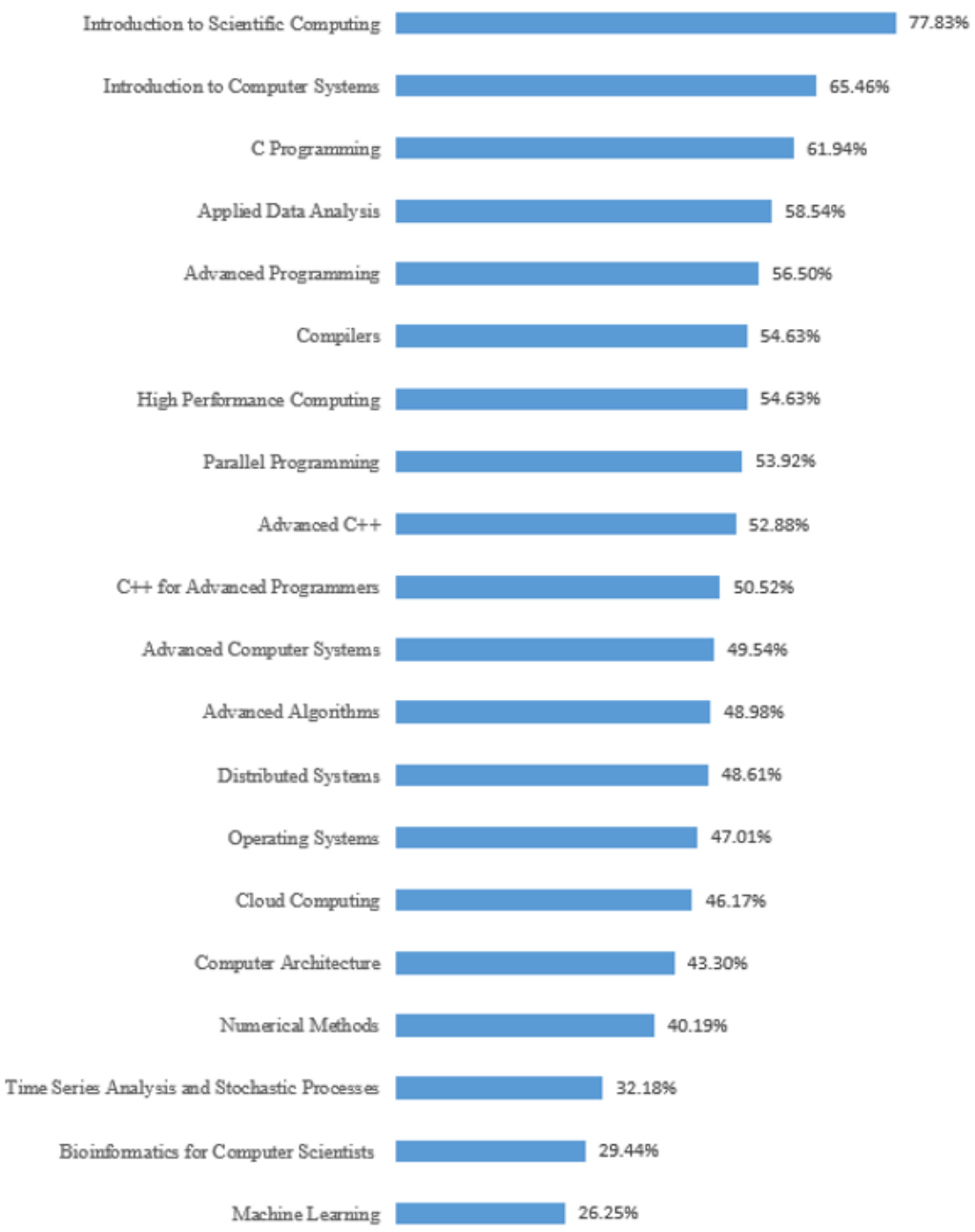

Figure 29: The American university's HPC MSc courses coverage of ESCO skills for System Architect occupation – in percentage

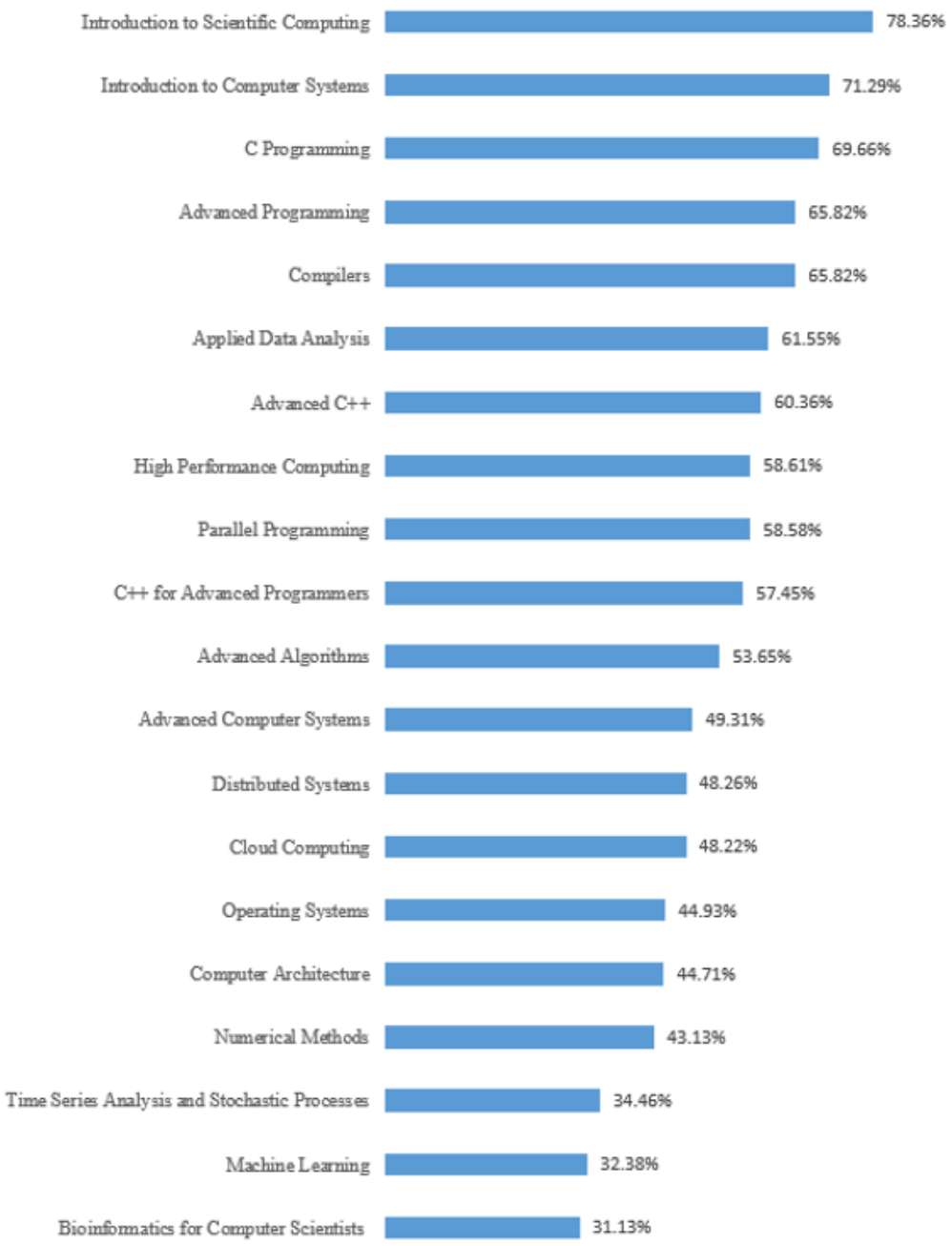

Figure 30: The American university's HPC MSc courses coverage of ESCO skills for Software Developer occupation – in percentage

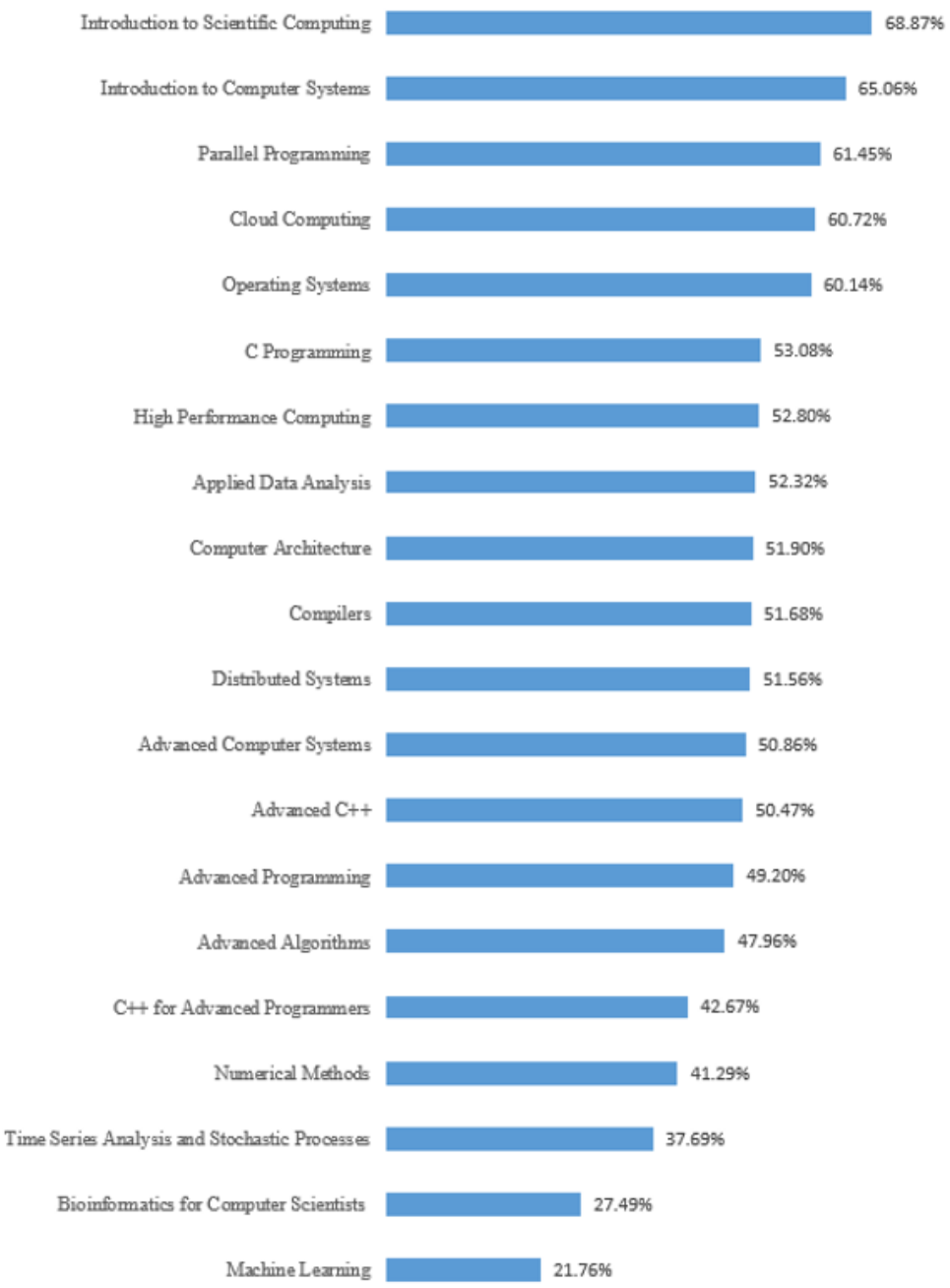

Figure 31: The American university's HPC MSc courses coverage of ESCO skills for DevOps Engineer occupation – in percentage

To sum up the findings given in this section, it can be seen from the results that the distribution of skills taught in the European university's HPC MSC programme seems more aligned with four occupations from ESCO. In addition, when the courses are divided into two categories as soft skills and hard skills, in these two HPC MSc

programs, the low number of soft skill related courses is striking. This proves that these programs especially focus on technical courses.

5.4 Comparison of different MSc programs' curriculums

According to the Q1 values calculated in the box plots presented in Figure 22 and Figure 23, the following heat maps, Figure 32 and Figure 33, are redrawn. It has been shown that semantic similarity decreases as you go from dark green to dark red in heatmap coloring. Courses with a similarity value close to Q1 are represented on the orange scale. In other words, as you go from Q1 to 1 (means that two sentences are exactly similar), coloring starts from the dark tone of orange to the tones of yellow, and as the value increases, the transition from the light tone of green to the darker tone is made.

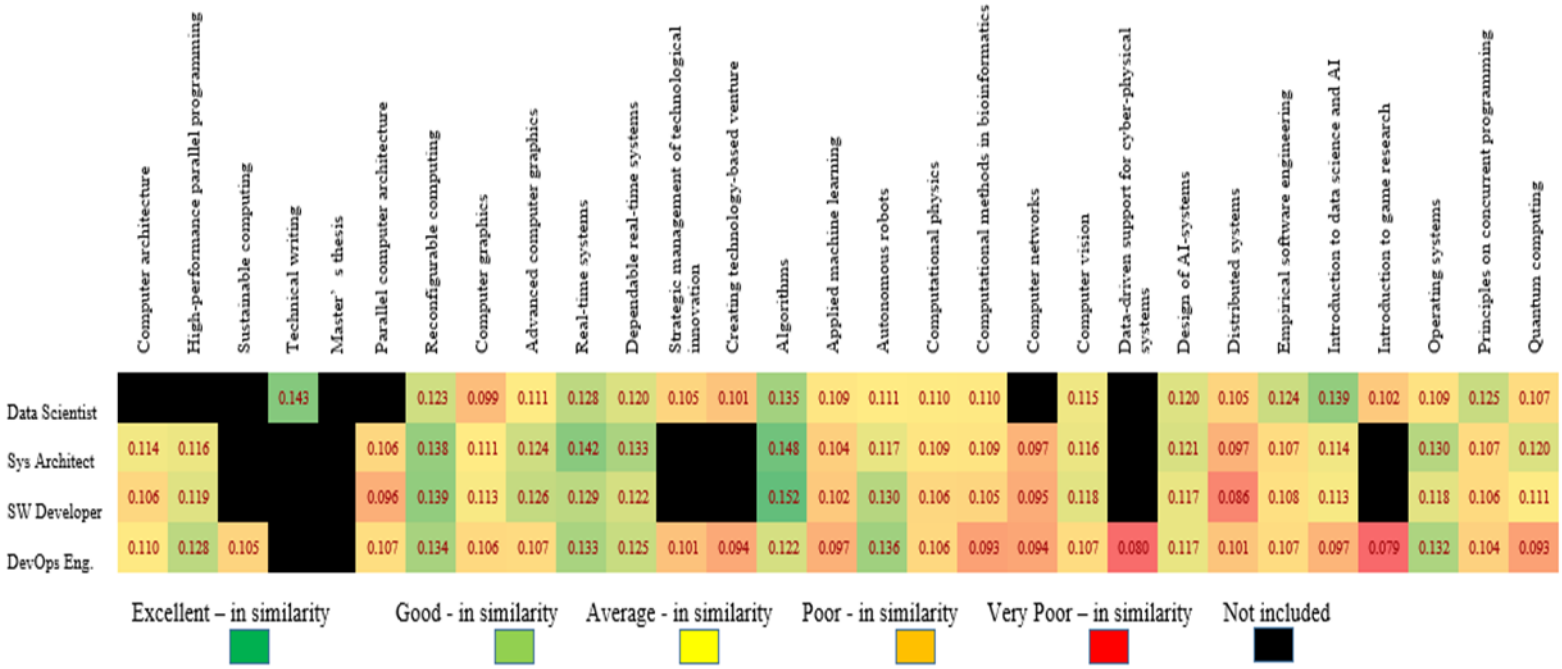

Figure 32: The European university's HPC MSc courses - Heatmap with higher similarity values than Q1 (cut off value)

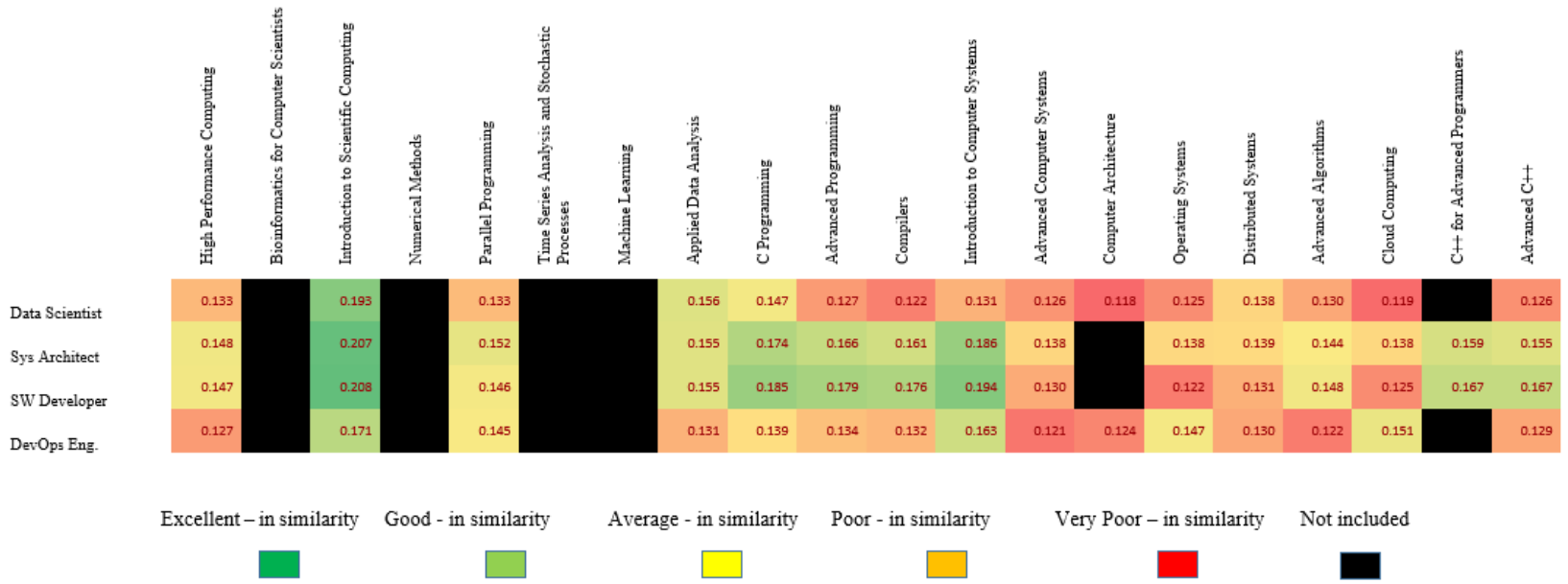

Figure 33: The American university's HPC MSc courses - Heatmap with higher similarity values than Q1 (cut off value)

As a conclusion, with the help of developed methodology in this thesis, the results were evaluated for one of the European and American universities with respect to data scientist, system architect, software developer, and devOps engineer occupations. According to the findings, as a result of the comparison of the European university with ESCO skills, it was concluded that it gave more stable and consistent results, while the American university gave more scattered results. It can be concluded that the semantic similarity comparison gave more consistent results for the European university because it is one of the contributing partners of EUMaster4HPC project and opens the HPC graduate program in this context. It is considered that the course selections and course definitions are made in accordance with the project outputs. Since this program is a new program, the curriculum has been newly created, the curriculum design phase has benefited from the outputs of the EUMaster4HPC project, and the number of courses is more than the American university's HPC MSc programme, it performed better in semantic similarity comparison. However, the point that should not be ignored here is that ESCO and the selected European university are located in Europe and act in line with the European Union's 2020 strategy. (Chiarello, 2021) concluded that ESCO strives for continuous improvement to gather comprehensive information about emerging technologies and how they affect the skills/knowledges undertaken at occupations. The authors stated that ESCO has been successful to some extent in this regard. There was a statistically significant improvement from ESCO v.1.0 to and 1.1. in alignment with technology trends related to I4.0. As they concluded, there are still some gaps that need to be filled in ESCO like HPC specific occupations and skills.

In this context, since ESCO was chosen as the only data source for occupations and related skills, in this thesis it was not aimed to compare the universities with each other.

5.5 Evaluation of ESCO whether it is up-to-date on the basis of occupation and skill according to market requirements for HPC field

In order to ensure the adequate adoption and exploitation of new technology, staff with required skillset and knowledge is crucial. To make use of new technologies, businesses frequently seek new personnel with advanced knowledge and skills. In order to satisfy changing labor market demands, educational systems should alter their curricula. The lengthy time this upgrade might take, however, contrasts with how quickly technology is developing. This may affect the alignment between the skills both offered by the educational institutions and demanded by the labor market. So, it is essential to provide timely and impartial information on the most recent workforce requirements in order to assist the transition in education. The field of HPC has been deemed ideal for establishing a methodology to evaluate skills gaps since it is thought to be one that has not yet reached its full potential in the industry. The created methodology is not dependent on the field or the course outline. This methodology is made to be used for many educational levels, initiatives, and curricula. However, given the dearth of data and the challenge of acquiring current data for HPC, it is thought that this methodology may function better in other areas given the newly emerging field of HPC that was chosen for this study.

The ESCO database, a dynamic resource that is always being updated and expanded, served as the source for the occupations. Since, ESCO lacks specific HPC-related skills, the dataset used in this thesis can be improved by employing a list of the most important terms from applicable scientific articles, new deliverables from EUMaster4HPC, job openings, or course descriptions. The investigation conducted for this thesis demonstrated that ESCO has not yet identified any jobs or skills linked to HPC (supercomputing, quantum computing, etc.). Hence, for the research, the HPC skill tree was employed as a reference in order to differentiate which abilities listed in ESCO were connected to the HPC field.

When the job postings were analyzed, it was seen that there were high performance computing related occupations like;

- High performance computing scientist,
- High performance computing platform engineer,
- HPC architect,
- HPC software consultant,
- HPC computational scientist,
- HPC system administrator,
- Machine learning engineer,
- Data curator,
- HPC performance engineer,
- Cybersecurity software developer etc.

However, these occupations are not listed in ESCO.

So, the analysis in this thesis is focused on four relevant occupations from the HPC field: data scientist, software developer, system architect, and devOps engineer which are thought to be the most related occupations with HPC. Methodology was applied for the chosen universities.

When ESCO was analyzed for four occupations through their essential and optional skills, and essential and optional knowledge, it was clear that these four occupations are mainly from the computer engineering or software engineering domain. If ESCO had included HPC domain-specific occupations and skills, the frequency analysis results shown in below would have changed. The most frequent words describing the skills of four occupations were subjected to frequency analysis. Figure 34 shows the frequency analysis results as tags cloud for the Data Scientist occupation. The words "data" account for 46% of the total number of occurrences, making it clear that data related tasks dominate the occupation.

Figure 34: Frequency analysis for the most common words in Data Scientist occupation

Figure 35 shows the frequency analysis results as tags cloud for the Software Developer occupation. It is obvious that the most common words to describe this occupation are “programming, software, computer, and develop”.

Figure 35: Frequency analysis for the most common words in Software Developer occupation

Figure 36 shows the frequency analysis results as tags cloud for the System Architect occupation. It shows that this occupation is defined with the words “programming” (10%), “system” (7%), “computer”, and “ICT” (6%) in general.

Figure 36: Frequency analysis for the most common words in System Architect occupation

As mentioned in the previous sections, devOps engineer is not listed in ESCO's occupations. It was listed under "Alternative Labels" of "ICT change and configuration manager" occupation. Figure 37 shows the frequency analysis results as tags cloud for this occupation. From the word frequency analysis results shown, it is easily understood that these words are the words that describe the ICT configuration manager occupation.

Figure 37: Frequency analysis for the most common words in DevOps Engineer occupation

5.6 Individualized recommendations for improvement of syllabuses

The analysis presented here attempts to assist academia and businesses in addressing the challenge faced with skills gaps. By creating a methodology that employs NLP for

semantic sentence similarity, a data-driven approach was presented to evaluate the alignment between the educational and labor market components in the HPC sector. English descriptions of master's degree programs and the ESCO database were used to identify skills gaps using NLP. Skills in both data sources were assessed using pair-wise sentence similarity, which allowed comparison of two textual sources. The approach can be used to explore additional linguistic and educational contexts.

In order to help increasing the relationship of curricula with occupations and ensuring that MSc students can graduate readier for working life, produced heat maps and graphs have been analyzed course by course to make concrete recommendations.

To remind of what was explained in Section 4.2, the curriculum at the selected European university was structured so that it begins with classes that lay the groundwork for high performance computer systems. Computer architecture, high performance parallel programming, and sustainable computing are these required courses. Students can select academic specialties like machine learning, computational sciences, operating systems, and networking through elective courses. Therefore, to begin the master's degree in high performance computer systems, the institution prefers students who have taken courses in basic computer organization, machine-oriented programming, concurrent programming principles, mathematical modeling, and problem solving. Students might choose to specialize in entrepreneurship, computer graphics, computer systems, or real-time systems as part of their required optional courses. Advanced computer graphics, real-time systems, and reliable high-performance computing are just a few of the high-performance computing-focused topics that are covered in some of the required elective courses. Another notable point is CPU problem optimization. This issue receives relatively little attention outside of the high-performance sector and is consequently poorly taught in traditional computer science classes. However, as this information is now required for effective high-performance computing, it ought to be covered more in this program.

The curriculum requires to be updated approximately an equal number of credits in mathematics, computer science, and the application field, and also it should include

more courses related with soft skills. Faculty from many departments collaborate to teach multidisciplinary courses in this program. High performance computing is a focus of various courses the department offers.

According to the analysis results, the most overlapping course with the skills of the data scientist occupation defined in ESCO is the introduction to data science and AI course in the European university's HPC MSc programme. The least overlapping course is seen as Master's Thesis course with compared to all occupations. If this course is explained using sentences that describe the course such as thesis titles, thesis topics that can be prepared within the scope of the course, the match will be higher. In addition, it was seen that the computer science and engineering department at this university serves as the primary inspiration for the high-performance computing courses there. In addition, the relationship between the qualifications framework for higher education in Europe should be established while preparing course syllabuses and defining the courses.

As mentioned in Section 4.2, the second selected university is from U.S.A. High performance computing is one of the program's compulsory courses. Other required compulsory courses include bioinformatics for computer scientists, introduction to scientific computing, numerical methods, parallel programming, time series analysis and stochastic processes, and machine learning or sometimes known as applied data analysis or applied machine learning. Core systems courses include compilers, introduction to computer systems, advanced computer systems, computer architecture, parallel programming, operating systems, and distributed systems. Core programming courses include C and advanced programming. Advanced algorithms, cloud computing, C++ for advanced programmers, and advanced C++ are the final suggested elective courses. Modern courses in software engineering; machine learning, high speed computing, web development, cloud computing, big data analytics, and application development are available as electives to help students keep up with the rapidly changing world of technology.

From the course list of high-performance computing MSc program in the American university, it is seen that the program strikes a realistic mix between applied courses and foundational courses in areas like algorithms, databases, and programming languages.

According to the findings of the methodology presented in this thesis, it seems that the European university's HPC MSc programme covers more of the selected occupations, while the American university's HPC MSc programme seems to cover at the lower level, especially for devOps engineer and data scientist. While interpreting the results, it should also be considered that both compulsory and elective courses were examined. Therefore, the courses analyzed include basic computer-software engineering courses along with HPC domain specific courses. This is expected, since students of these programs can come from a wider background with undergraduate degrees not related to computer or software engineering domains.

Although the courses offered by the program are from both the computer science domain and the high-performance computing domain, when the results of the sentence similarity methodology were examined, it was seen that the HPC MSc programme of the selected American university did not overlap enough with the occupations selected for HPC and it was seen that it performed worse when compared to the European university. However, the underlying reason for this was the fact that the definitions of the skills required for the occupations selected in ESCO and the sentences in the course definitions of the program are structurally different from each other.

The analysis performed better for software developer occupation, and performed worst for data scientist occupation with respect to courses of the American university. The interesting thing about the sentence similarity results of this university was how varied the similarity values between the advanced C++ and the C++ for advanced programmers courses are for same occupation. This is thus because the two courses have very distinct definitions from one another. The fact that the machine learning course seemed to be the one with the least similarities was another unexpected finding. According to the findings, it can be assumed that ESCO occupation definitions do not

correspond to the descriptions and contents of MSc courses, course definitions are not sufficiently detailed, and there are no definitions for the skills and knowledge that students should possess upon successful completion of the course. In universities and even in ESCO, there is no established standard for defining courses and skills, consequently the compared sentences in this methodology were very dissimilar from one another. In addition, when the course list was examined, it was seen that no soft skill related course was given under this program. All the courses selected for the program were completely focused on hard skills.

According to the THE university rankings, the results of the analysis with this thesis would be expected to be much better for the American university than for the European university, because it ranks between 10-50 in THE world rankings, while the European university ranks between 250-300.

A few useful ramifications of the findings were emphasized in this part to help academia deal with misalignments with business. The results showed that educational institutions need to explain courses and learning objectives in a clearer, more thorough manner. Course outlines can be more precisely defined by adhering to the Bologna Process, and, to further close the gap between industry and education, universities should use the ESCO database (if its version is up-to-date) as a guide when creating educational programs that adhere to industry standards.

The question of how well a current major curriculum qualifies the students as candidate practitioners in the HPC domain to handle real-world difficulties poses a significant challenge to academics. To bridge the skills gap revealed by this study, academics should reconsider their curricula. Owing to the quick changes in industrial needs, this adaption should be futuristic, have a quick response to changes, and support current practices in the industry. In order to meet industrial expectations in full through mutual understanding, industry and academia should collaborate to develop an up-to-date curriculum. In addition, curriculums should be updated via the addition of new courses or the revamping of current ones to ensure that soft skills are covered more thoroughly across all curricula. In order to satisfy the demands of the software

industry, it is also required to examine particular needs based on HPC domain. Academics should place more emphasis on teaching real-world software examples by concentrating on themes that are pertinent to industry. There is a gap between the theory taught and the practical issues with high performance computer systems (Akdur, 2021). Since the courses examined in the analysis presented in this thesis did not provide information on whether the courses are taught in a classroom or laboratory environment, a comparison could not be made such as to what extent practical issues were met.

CHAPTER 6

6 CONCLUSIONS

In order to assist academia dealing with misalignments with the business, a few practical consequences of the findings were highlighted in this section. First, the findings indicated that institutions must be more thorough and explicit when presenting the courses and their learning objectives. The Bologna Process can be followed to define course syllabuses in more rigid way.

This thesis was concentrated on the gaps between ESCO and the courses offered in MSc programs (specialized in HPC) and explored the skills gap from a larger viewpoint in order to create an effective curriculum for HPC. Even though numerous new technologies have been introduced, some skills/competences change or evolve along with the new technology, while others remain the same over time and are still considered to be crucial. The outcomes demonstrated that ESCO is yet to adopt HPC related changes in the sector. Educational institutions often develop their curricula from a competency perspective or focus on the role people want to take on after the course. In the EUMaster4HPC project, the main goal is to design a curriculum that will meet the business' requirements in the field of HPC.

This study provided a common reference scheme for the competencies, skills, knowledge and competence levels needed by those who want to work in the HPC fields related to data science, computer architecture, parallel programming, devOps engineer occupations in line with the ongoing EUMaster4HPC project. The case was chosen as high-performance computing, however, the purpose of the methodology developed in this study is to draw a reference frame that is adaptable not only for the specified fields, but also for all fields and all educational programs. In this context, four profiles in the field of HPC were selected in order to explain and apply the methodology. It also presents the basic data in order to provide support for designing the curricula, which is the aim of the thesis.

In this thesis, the most important skills in the high-performance computing field are analyzed, the adequacy of ESCO skills was scrutinized, and the coverage of these skills in the curriculums of one of the European and American universities' MSc programs on high performance computing were assessed by using the NLP USE model. This thesis proposed an assessment methodology to provide information about the skills gaps between ESCO and MSc curricula, specifically for the selected universities.

The idea of descriptive statistics was used to analyze the similarity vectors and gain a better understanding of what the data means. For the European university, a total of 146124 comparisons were made via 297 skill sentences in ESCO and 492 course sentences in MSc program. For the American university, a total of 82269 comparisons were made via 297 skill sentences in ESCO and 277 course sentences in MSc program. It was necessary to determine a cut off value in order to evaluate the NLP score. In this thesis, the lower quartile value was taken by using descriptive statistics for this value. If this value is changed or a different statistical method is used, the results can change. In this thesis, two universities were chosen to test the developed methodology. In the field of HPC, there are still limited MSc programs in the universities, and there is an ongoing progress in defining the curriculum of HPC related MSc programs. For the existing programs, it has been observed that the explanations in the syllabus and the skills/knowledges expected to be gained by the student after completing the course are not written at a sufficient level.

With this study, there was a contribution in the way to provide information for the universities involved in the EUMaster4HPC project and trying to develop new curricula in the field of HPC. ESCO was evaluated in the way of its skills and competences. Knowledge pillar has not been included in this study. The results showed that ESCO may not contain sub-skill level data in sufficient detail. Considering that HPC is a new growing field in the sector, it is seen that both industry and academia have not completed their development in this field yet, and there are missing areas as shown in this work. This work described the skills/competences environment in accordance with ESCO in order to promote knowledge and acknowledgment of the issues with curriculum design.

During this work, a demo code was developed to demonstrate the methodology. After the syllabuses are prepared as csv in the desired format, comparisons can be made with the desired skills/competences defined in ESCO. The code takes as input the syllabuses csv and the occupations of the ESCO to be compared as well as a cut off value and produces as output the comparison results.

The course syllabuses were examined in the way of compulsory, compulsory elective, and elective courses. Note that, although course types were categorized in the course syllabuses, in this thesis all courses were included to test the methodology, because the focus was on what the programs can offer, rather than what the students actually get. If there is no other option available due to course quota restrictions, a graduate student may take an optional course from other programs. This may happen voluntarily or involuntarily, so it has not been estimated in the calculations.

6.1 Limitations

In this section, potential validity issues for the developed methodology are discussed. There are certain limitations of this study that must be recognized. Additionally, these restrictions bring up a number of new study avenues. The only data sources used for the developed methodology was ESCO, and the selected universities' MSc programme on HPC course syllabuses. Considering that ESCO lacks specialized HPC-related occupations, the dataset can be enhanced with the list of the most pertinent terms from relevant scientific articles, using EUMaster4HPC new deliverables, using job vacancies or course descriptions. By the help of the analysis done in this thesis, it was shown that in ESCO, HPC related occupations and skills (supercomputing, quantum computing, etc.) are not yet defined. So, for the research, the HPC skill tree (HPC Competence Framework, 2022) was used as a reference in order to distinguish which skills defined in ESCO were related to the HPC field. It was assumed that "*Data Scientist, Software Developer, DevOps Engineer, and System Architect*" occupations were the most related occupations in the field of HPC according to (EuroHPC JU, 2022) deliverable 2.2.

Since the field of HPC is seen as a field that has not yet completed its development in the sector, this field has been evaluated as suitable for developing a methodology. The developed methodology is independent of the type of the field and the course syllabus. This methodology is designed to be used for different education levels, programs and syllabuses. However, since the newly developing area of HPC was chosen in this study, it is considered that this methodology could work better in different areas, given the lack of sufficient data and the difficulty of accessing existing data for HPC. Moreover, ESCO requires definitions for the HPC domain. Sentences from the wider software engineering domain that are not focused on HPC, cannot be compared to generate relevant findings. In addition, the data that was posted to the ESCO is referred to in the limitations of this thesis. The ESCO portal is thought to have data that is more pertinent to the European labor market because the majority of EU Member States enter data there.

For sentence similarity methodology to provide more accurate results, the definitions of courses provided in university web sites should be made as detailed as possible and in accordance with ESCO or other qualification frameworks. In addition, it is essential that the skills and the knowledge the students will acquire after taking a course is specified in course definitions, in order to get more meaningful results.

In addition, there is a need for HPC domain specific definitions in ESCO. Comparisons made with sentences taken for the general software engineering domain, which is not specialized for HPC cannot provide the meaningful results. For this reason, it is normal that the basic computer or software engineering courses and skills come out at the top percentages, while those specialized in HPC come out in the middle of the bar graphs presented in figures between Figure 24 and Figure 31. If there were defined occupations like high performance computing system administrator, HPC computational scientist, high performance computing architect, machine learning engineer etc. and related skills in ESCO, and the course definitions were in line with these skills in universities, then the semantic similarity method would have given more accurate results.

When universities with HPC graduate programs in Europe are examined, the data obtained could not be used in this study because most of them do not provide access to their syllabuses; only the students enrolled in the university are given access, usually in the country's native language. For universities that will open HPC programs in the 2022-2023 academic year, the situation seems to be that the curricula have not yet been defined at a detailed level, the syllabus has not been created, and they do not provide information on what gains the students will achieve.

EUMaster4HPC has released deliverable 2.3 and 2.4 as this thesis was nearing completion. These deliverables could not be examined due to time constraints in this thesis.

6.2 Implications for Further Research

From the viewpoint of academia and business, this thesis offered quantitative analysis about the degree to which course syllabuses in selected universities cover the skills/competences related with HPC in ESCO. Hence, in the future, different fields, universities, degree programs can be compared with ESCO to bridge the gaps between the academia and the business, and also the results can be evaluated by comparing the programs of the universities with the worldwide accepted university ranking lists.

REFERENCES

Akdur. (2021). Skills Gaps in the Industry: Opinions of Embedded Software Practitioners.

Al-Alak, B. A. (2012). Assessing the relationship between higher education service quality dimensions and student satisfaction. *Australian Journal of Basic and Applied Sciences*, 156-164.

Almgerbi, M. D. (2021). A systematic review of data analytics job requirements and online-courses. *J. Comput. Inf. Syst.* , 1–13.

Brunello. (2019). Skill Shortages and Skill Mismatch in Europe: A Review of the literatue.

Carayannis, E. M.-J. (2022). 2022. The futures of Europe: Society 5.0 and Industry 5.0 as driving forces of future universities. .*J. Knowl. Econ. 1–27.*, 1-25.

Chiarello. (2021). Towards ESCO 4.0- Is the European classification of skills in line with Industry 4.0? A text mining approach. *Elsevier*.

Chowdhary, K. (2020). *Natural Langguage Processing In: Fundamentals of Artificial Intelligence.* New Delhi: Springer.

DA.RE. Project: https://dare-project.eu adresinden alındı, (2022, 11 09).

Data Science Learner. (2022, December 9). Data Science Learner: https://www.datasciencelearner.com/how-to-install-en_core_web_lg-spacy-model/#:~:text=What%20is%20en_core_web_lg%3F,size%20is%20only%2013%20MB. adresinden alındı

Devlin, J. C. (2019). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding .

Di Luozzo, S., D'Orazio, L., & Schiraldi, M. (2021, September). Skills Mismatch in Operations & Supply Chain Mannagement roles: perceptions from the European Skills, Competences, Qualifications and Occupatios database. *ResearchGate*.

D'Orazio. (2020).

E.Rata. (2021). The Curriculum design coherence model in the knowledge-rich school project. *Rev. Educ.*, 448-495.

EUMaster4HPC. (2022). EUMaster4HPC: https://eumaster4hpc.uni.lu/the-project/ adresinden alındı

EuroHPC JU. (2022, 10 16). https://eurohpc-ju.europa.eu/participate/our-projects/eumaster4hpc_en adresinden alındı

European Commission ESCO. (2022, 09). European Commission: https://esco.ec.europa.eu/en adresinden alındı

European e-Competence Framework. (2022, November 09). European e-Competence Framework: https://ecfexplorer.itprofessionalism.org/ adresinden alındı

Fernandez. (2017). e-Skills Mismatch: A framework for mapping and integrating the main skills,knowledge and competence standards and models for ICT occupations. *Elsevier*.

Floydhub. (2022, December 15). Floydhub: https://blog.floydhub.com/when-the-best-nlp-model-is-not-the-best-choice/ adresinden alındı

Grolemund, G. W. (2018). *R for Data Science.*

Guoyan Li, C. Y. (2021). Data Science skills and domain knowledge requirements in the manufacturing industry: A gap analysis. *Journal of Manufacturing Systems*, 692-706.

Gupta V., L. G. (2009). A survey of text mining techniques and applications. *J. Emer. Technological Web Intelligence*, 60-76.

HPC Competence Framework: https://www.hpc-certification.org/cs/ adresinden alındı, (2022, 10 20).

HPC CF: https://www.hpc-certification.org/cs/ adresinden alındı, (2022, 11 1).

Handel, M. J. (2016). The O* NET content model: Strengths and limitations. Journal for. 157.

Harris, Z. (1954). Distributional structure. 10 (2–3). Z. Harris içinde, *Word* (s. 146–162).

joinUp. (2022). https://joinup.ec.europa.eu/collection/labour-market-interoperability/about-esco-european-classification-skills-competences-qualifications-and-occupations-esco adresinden alındı

Josten, C. &. (2020). Robots at work: Automatable and non-automatable jobs. *Springer International Publishing*.

Kahlawi. (2020). A Similarity matrix approach to empower ESCO interfaces for testing, debugging and in support of users' experience. *ResearchGate*.

Khodayari, F. &. (2011). Service quality in higher education. *Interdisciplinary Journal of Research in Business*, 39-40.

Kipper, L. I. (2021). Scientific mapping to identify competencies required by Industry 4.0. *Technology Society*, 64.

Legčević, J. (2009). Quality gap of educational services in viewpoints of students. *Ekonoska Misao Praksa DBK. GOD*, 279-298. .

Liu, K. S. (2016). Good skills in bad times: Cyclical skill mismatch and the long-term effects of graduating in a recession. *European Economic Review*, 8-15.

Maer-Matei, M. M. (2019). Skill needs for early career researchers—a text mining approach. *Sustainability 11*.

Male, S. (2010). Generic engineering competencies: a review and modelling approach. *Educ. Res. Perspect*, 25-51.

Manacorda. (1999). Skill Mismatch and Unemployment in OECD Countries.

Marwedel. (2020). How can we educate students such that they will be able to design CPS systems? .

Md. Mamun-ur-Rashid, M. Z. (2017). Quality of higher education in bangladesh: application of a modified servqual model. *Problems of Education in the 21th century Vol. 75, No.1*, 1-20.

Moldovan, L. (2019). State-of-the-art analysis on the knowledge and skills gaps on the topic of Industry 4.0 and the requirements for work-based learning. *Procedia Manufacturing*, 294-301.

Neutel. (2021). Towards Automatic Ontology Alignment Using BERT.

O*NET. (2023, 2 11). https://www.onetonline.org/help/onet/ adresinden alındı

O*NET Online. (2022, 9). O*NET: https://www.onetonline.org/help/online/ adresinden alındı

OECD. (2019). *OECD Works on Education and Skills*. OECD.

Parasuraman, A., Zeithaml, V., & Berry, L. (1988). SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality. *Journal of Retailing*, 12-40.

Passow, H. P. (2017). What competencies should undergraduate engineering programs emphasize? A systematic review. *J. Eng. Educ.*, 475-526.

Pellizzari, M. a. (2017). A new measure of skill mismatch: theory and evidence from PIAAC. *IZA Journal of Labor Economics, IZA Journal of Labor Economics*, 10-15.

Pınar Özdemir, A. S. (2023). Closing the gap between present and future through education: MINE-EMI project. *Elsevier*.

Radenko Milojević, M. R. (2022, November 11). *Assessment of higher education service quality: integration of servqual model and ahp method.* LinkedIn: https://www.linkedin.com/pulse/assessment-higher-education-service-quality-servqual-ventura-phd- adresinden alındı

Smedt, D. (2020). ESCO: Towards a Semantic Web for the European Labor Market.

Spada, I. (2022). Are universities ready to deliver digital skills and competences? *Elsevier*.

Thomas Sterling, M. A. (2018). *High Performance Computing.* Morgan Kaufmann.

Thomas Sterling, M. A. (2018). *High Performance Computing Modern Systems and Practices*. MK.

Times Higher Education. (2023, February 12). https://www.timeshighereducation.com/ adresinden alındı

Times Higher Education. (2023, February 13). https://www.timeshighereducation.com/world-university-rankings/ adresinden alındı

Travelyen. (2014). The Making of an Expert Engineer.

Trevelyan. (2019). Transitioning to engineering practice.

Vasilicia. (2019). Competences between labor market and higher education through ESCO.

Wikipedia. (2022, November 11). https://en.wikipedia.org/wiki/SERVQUAL adresinden alındı

Yousapronpaiboon, K. (2014). Servqual: Measuring Higher Education Service Quality in Thailand. *Procedia, Social and Behavioral Science*, 1088-1095.

APPENDICES

Appendix A

The sentences in the courses' weekly schedule were redesigned and structured to be understandable by the NLP.

The European University's HPC MSc programme

	Course	DETAIL
1	**Computer architecture**	architectural techniques essential for achieving high performance for application software. simulation-based analysis methods for quantitative assessment of the impact a certain architectural technique has on performance and power consumption. trends that affect the evolution of computer technology including Moore s law, metrics of performance (execution time versus throughput) and power consumption, benchmarking as well as fundamentals of computer performance such as Amdahl s law and locality of reference. how simulation based techniques can be used to quantitatively evaluate the impact of design principles on computer performance. various techniques for exploitation of instruction level parallelism (ILP) by defining key concepts for what ILP is and what limits it. dynamic and static techniques. Tomasulo s algorithm, branch prediction, and speculation. loop unrolling, software pipelining, trace scheduling, and predicated execution. memory hierarchies. attack the different sources of performance bottlenecks in the memory hierarchy such as techniques to reduce the miss rate, the miss penalty, and the hit time. victim caches, lockup-free caches,

		prefetching, virtually addressed caches. main memory technology is covered in this part. multicore, multithreaded architectures. programming model and how processor cores on a chip can communicate with each other through a shared address space. micro architecture level it deals with different approaches for how multiple threads can share architectural resources. fine-grain, coarse-grain and simultaneous multithreading. master concepts and structures in modern computer architectures. understand the principles behind a modern microprocessor; advanced pipelining techniques that can execute multiple instructions in parallel in order to be able to establish performance of computer systems;. understand the principles behind modern memory hierarchies in order to be able to assess performance of computer systems. proficiency in quantitatively establishing the impact of architectural techniques on the performance of application software using state-of-the-art simulation tools.
2	**High-performance parallel programming**	parallel computer architectures and parallel programming models and paradigms. mechanisms for synchronization and data exchange. performance analysis of parallel programs is covered. tools and techniques for developing parallel programs in shared address spaces. popular programming environments such as pthreads and OpenMP. development of parallel programs for distributed address space. Message Passing Interface MPI. programming approaches for executing applications on accelerators such as GPUs.

		CUDA (Compute Unified Device Architecture) programming environment. parallelize sample programs over a variety of parallel architectures, and use performance analysis tools to detect and remove bottlenecks in the parallel implementations of the programs. List the different types of parallel computer architectures, programming models and paradigms, as well as different schemes for synchronization and communication. List the typical steps to parallelize a sequential algorithm. List different methods for analysis methodologies of parallel program systems. Apply performance analysis methodologies to determine the bottlenecks in the execution of a parallel program. Predict the upper limit to the performance of a parallel program. Given a particular software, specify what performance bottlenecks are limiting the efficiency of parallel code and select appropriate strategies to overcome these bottlenecks. Design energy-aware parallelization strategies based on a specific algorithms structure and computing system organization. Argue which performance analysis methods are important given a specific context

3	**Sustainable computing**	the energy efficiency aspects of computer systems and computing, ranging from the electronic circuits up to the applications for systems ranging from small IoT devices to large data centers. measuring and estimating the energy consumption of different architectural components as well as architecture and software techniques to save energy in the system. describe the key aspects in sustainable computing and how computer systems can be used to improve sustainability. describe techniques used in large data-centers to improve efficiency and sustainability. describe the electrical mechanisms that cause power to be dissipated. describe circuit techniques for reducing power dissipation and the impact on performance. describe computer architecture, memory, and secondary storage techniques for reducing energy consumption. describe techniques at the operating system, runtime, and application for reducing energy consumption. explain what affects the energy consumption of computer systems especially concerning their architecture. use specific devices to directly measure energy consumption of the whole system. use performance counters to measure the energy consumption of certain components in the system. use simulation tools to estimate the energy consumption of different system configurations. identify key trends in research and industry that lead to more efficient and sustainable systems. identify the strengths and weaknesses of different classes of computer system components like processor and memory, with respect to

		energy efficiency. evaluate and compare different architecture and system techniques in terms of the energy efficiency. explain the methods for evaluating and reporting the energy consumption in computer systems and how these can be used to optimize the system. judge the importance of energy consumption from societal and ethical perspectives. interpret requirements on the architecture of computer systems to meet societal needs for sustainability.
4	**Technical writing**	underlying structure of scientific and engineering research papers. Improve proficiency in reviewing and writing scientific research papers. Presenting such papers in public. Ethical issues in connection with scientific writing, plagiarism and authorship.
5	**Master's thesis**	thesis work in an industrial context or within a research group at university.

6	**Parallel computer architecture**	a review of fundamental concepts in computer architecture. basic multiprocessor designs for the message passing and shared memory programming models. interconnection networks, an essential component in chip multiprocessors and scalable parallel computer systems. how to correctly support parallel algorithms in shared memory hardware. last years' recent transition towards chip multiprocessors (also known as multicores). cost tradeoffs with respect to performance, power, energy, verifiability, programmability, and maintainability. memory bottleneck, and the importance of efficient resource management. The lectures are complemented with several exercise sessions. Via three lab assignments, participants learn how to develop software using models such as C++ threads and OpenMP, they develop and analyze synchronization algorithms, and they learn how to use performance analysis tools. The course also contains a written assignment in which the participants take the role of the computer architect who will survey and discuss solutions to a particular problem in the field of parallel computing. describe current approaches to parallel computing. explain the design principles of the hardware support for the shared memory and message passing programming models. describe the implementation of different models of thread-level parallelism, such as core multithreading, chip multiprocessors, many-cores or GPGPU. implement

		synchronization methods for shared memory and message passing parallel computers. design scalable parallel software and analyze its performance. analyze the trade-offs of different approaches to parallel computing in terms of function, performance and cost.
7	**Reconfigurable computing**	Reconfigurable Computing Hardware. Reconfigurable Devices. Reconfigurable Architectures. Reconfigurable Systems,. Programming Reconfigurable Systems. Compute Models e.g. streaming, SIMD. VHDL Programming. High Level Synthesis. Partial and Dynamic Reconfiguration. Mapping Designs to Reconfigurable Platforms. FPGA Applications. Application characteristics. Hardware or Software Partitioning. network processing, stream processing, Machine learning. Reconfigurable Computing. Memory centric. Defect and Fault Tolerance. describe reconfigurable devices, architectures and systems. recognize the function and uses of reconfiguration techniques. identify application characteristics that can be supported well in reconfigurable devices. identify different compute models for reconfigurable systems and

		recognize how they fit particular application characteristics. approach a design problem targeting reconfigurable computing (application analysis, design solution, implementation choice). use modern tools to design and implement in reconfigurable hardware. use modern tools to perform hardware reconfiguration. measure the performance and energy costs of a hardware design mapped in reconfigurable hardware. evaluate the advantages and disadvantages of reconfigurable computing compared to other computing alternatives. compare different types of reconfigurable devices, architectures and systems. compare different compute models for reconfigurable systems. critically evaluate and judge an application implementation mapped to a reconfigurable system.
8	**Computer graphics**	implementing 3D graphics algorithms using C, C++ or possibly Java. principles used to create images through computer algorithms. real-time rendering and photo realistic rendering. real-time rendering, and techniques for illumination, special effects, shadows and reflections will be studied. Design of graphics hardware and speedup algorithms will also be treated. generating photo-realistic images and includes studying of ray tracing and global illumination. the corresponding mathematics will be revealed. Describe the fundamental algorithms and processes used to create computer graphics in 3D-games and movies. Utilize the functionality of dedicated hardware support for graphics through programming

		interfaces. Implement efficient algorithms to generate 2-dimensional images from 3-dimensional models. Implement algorithms to generate real-time renderings and photo realistic renderings.
9	**Advanced computer graphics**	Describe more advanced algorithms and processes used to create computer graphics in 3D-games and movies. Implement more advanced algorithms to generate real-time renderings and photo realistic renderings. Computer graphics. knowledge in 3D graphics. Each student selects an advanced graphics technique, algorithm ambient occlusion, hair rendering, GPGPU applications, ray tracing and global illumination, GPU-ray tracing, hard and soft shadows, real-time indirect illumination, spherical harmonics, wavelets for CG etc. project around 3D-graphics (individually or in groups), for instance implementing a specific advanced extensive technique or implementing several advanced but smaller techniques. render engine of a game or stand alone programs.

10	**Real-time systems**	real-time computing systems. Characteristics of real-time systems. application constraints, design methods, task models, run-time mechanisms, architectures. Evaluation of real-time systems. performance measures, evaluation methodologies. Single and multiprocessor scheduling. problem definition, terminology, and algorithms. Complexity theory and NP-completeness in the context of real-time scheduling. Real-time communications. protocols and end-to-end delay guarantees. Fault-tolerance techniques for real-time systems. models, algorithms and architectures. Formulate requirements for computer systems used in time- and safety critical applications. Demonstrate knowledge about the terminology of scheduling, dependability and complexity theory. Describe the principles and mechanisms used for scheduling of task execution and data communication in real-time systems. Design real-time systems and apply techniques to verify whether the real-time requirements are met or not. Derive the theoretical performance limitations of a given real-time system. Reason about advantages and disadvantages regarding the choice of the optimal design for a real-time systems given certain conditions.
11	**Dependable real-time systems**	The course covers the following topics. Background. motivation for, and definition of, real-time computing systems. Characteristics of real-time systems. application constraints, design methods, task models, run-time mechanisms, architectures. Evaluation of real-time systems.

		performance measures, evaluation methodologies. Single and multiprocessor scheduling. problem definition, terminology, and algorithms. Complexity theory and NP-completeness in the context of real-time scheduling. Real-time communications. protocols and end-to-end delay guarantees. Fault-tolerance techniques for real-time systems. models, algorithms and architectures. Formulate requirements for computer systems used in time- and safety critical applications. Demonstrate knowledge about the terminology of scheduling, dependability and complexity theory. Describe the principles and mechanisms used for scheduling of task execution and data communication in real-time systems. Design real-time systems and apply techniques to verify whether the real-time requirements are met or not. Derive the theoretical performance limitations of a given real-time system. Reason about advantages and disadvantages regarding the choice of the optimal design for a real-time systems given certain conditions.
12	**Strategic management of technological innovation**	Industry dynamics of technological innovation, e.g. sources of innovation and types and patterns of innovation. Formulating technological innovation strategy, e.g. defining the organization's strategic direction, collaboration strategies and protecting innovation (intellectual property rights). Implementing technological innovation strategy, e.g., organizing for innovation and managing the new product development process. Apply tools

		and theories of strategic management of technological innovation. Convey insights from the analysis of strategic management of technological innovation. Participate in decision-making linked to strategic management of technological innovation.
13	**Creating technology-based venture**	theories, methods, and practical tools for creating technology-based ventures. illustrate how theoretical concepts are used in the practice of new business creation. Be familiar with modern startup theories including customer development, lean startup, and effectuation. Know the value and limits of business planning. Know the difference between a startup and a company. Know how to elicit valuable feedback through customer interviews. Know how to work with experiments and prototypes to speed up learning. Know how to work with metrics and key process indicators in startups. Know how to manage the transition from early adopters to mainstream users. Be familiar with sources of venture finance. Have learned about a range of specific tools and methods for efficiently identifying and testing business model assumptions. Have improved their analytic, interpersonal, creative and presentation skills. Know what startup support is available at university. Potentially have kick-started their entrepreneurial career.
14	**Algorithms**	Tools for analysis of algorithms. O-notation. Analyzing loops and recursive calls. Solving recurrences. Data structures and algorithms. Review of basic data structures. Combining data structures. Merge-and-find. Graph

		algorithms. Greedy algorithms. Divide-and-conquer. Dynamic programming. Short introduction to local search and approximation algorithms. Basic complexity theory. Complexity classes P, NP, and NPC, reductions. Examples of NP-complete problems. Coping with hard problems. describe your algorithms and their qualities. explain algorithms in writing, so that others can understand how they work, why they are correct and fast, and where they are useful. recognize that non-trivial computational problems, which need to be solved by algorithms, appear in various real-world computer applications and to formalize them. intractability. recognize intractable problems and other classes of problems like P, NP, NPC. prove the correctness of algorithms. . design. apply the main design techniques for efficient algorithms (for instance greedy, dynamic programming, divide-and-conquer, backtracking, heuristics) to problems which are similar to the textbook examples but new. perform the whole development cycle of algorithms. problem analysis, choosing, modifying and combining suitable techniques and data structures, analysis of correctness and complexity, filling in implementation details, looking for possible improvements, etc. perform simple reductions between problems, explain NP completeness, recognize various computationally hard problems which tend to appear over and over again in different applications, cope, at least in principle, with computationally hard problems, using heuristics,

		refinements of exhaustive search, approximative solutions, etc. critically assess algorithmic ideas and demonstrate the ability to resist the temptation to create obvious and seemingly plausible algorithms (which often turn out to be incorrect). analyse. explain why the time efficiency of algorithms is crucial, express the time complexity in a rigorous and scientifically sound manner, analyze the time complexity of algorithms (sum up operations in nested loops, solve standard recurrences, etc.) i.e. perform an objective evaluation of the performance and be able to compare it to other algorithms performance. main focus is on the design of algorithms from a given problem specification and the analysis of efficiency of these algorithms.
15	**Applied machine learning**	supervised learning, such as linear models for regression and classification, or nonlinear models such as neural networks, and in unsupervised learning such as clustering. The use cases and limitations of these algorithms, and their implementation. Methodological questions pertaining to the evaluation of machine learning systems , as well as some of the ethical questions that can arise when applying machine learning technologies. the real-world context in which machine learning systems are used. The use of machine learning components in practical applications will be exemplified, and realistic scenarios will be studied in application areas such as ecommerce,. business intelligence, natural language processing, image processing, and bioinformatics. The importance of

		the design and selection of features, and their reliability, will be discussed. Describe the most common types of machine learning problems,. explain what types of problems can be addressed by machine learning, and the limitations of machine learning. account for why it is important to have informative data and features for the success of machine learning systems,. explain on a high level how different machine learning models generalize from training examples. . apply a machine learning toolkit in an application relevant to the data science area,. write the code to implement some machine learning algorithms,. apply evaluation methods to assess the quality of a machine learning system, and compare different machine learning systems. . discuss the advantages and limitations of different machine learning models with respect to a given task,. reason about what type of information or features could be useful in a machine learning task,. select the appropriate evaluation methodology for a machine learning system and motivate this choice,. reason about ethical questions pertaining to machine learning systems.

16	**Autonomous robots**	Survey of robot related hardware. Modern software development for autonomous robots. Kinematics and dynamics for autonomous robots. Simulation of autonomous robots. Perception and sensor fusion for autonomous robots. Behaviour modeling for autonomous robots. Practical work related to autonomous robots. Describe properties of common types of robotic hardware, including sensors, actuators, and computational nodes. Apply modern software development and deployment strategies connected with autonomous robots. Set up and use equations of motion of wheeled autonomous robots. Apply basic sensor fusion. Set up and use computer simulations of autonomous robots. Apply global and local navigation of autonomous robots. Apply the basics of behavior-based robotics and evolutionary robotics. Apply methods for decision making in autonomous robots. Discuss the potential role of autonomous robots in society, including social, ethical, and legal aspects. Discuss technical challenges with autonomous robots in society.
17	**Computational physics**	the programming language C. ordinary differential equations, molecular dynamics simulation. random numbers, random processes, Brownian dynamics. discrete and fast Fourier transforms, power spectrum analysis. Monte Carlo integration and the Metropolis algorithm. Variational and diffusion Monte Carlo. use C to solve numerical problems. explain and numerically apply the basic idea behind the molecular dynamics simulation method. explain how

		random numbers can be used to treat static and dynamic phenomena and numerically apply the methodology. explain and numerically apply the Metropolis Monte Carlo method. integrate knowledge in modeling physical systems with various numerical techniques. write well-structured technical reports where computational results are presented and explained. communicate results and conclusions in a clear way.
18	**Computational methods in bioinformatics**	Computational methods and concepts featured. dynamic programming; heuristic algorithms; graph partitioning; image skeletonisation, smoothing and edge detection; clustering, sub-matrix matching, geometric hashing; constraint logic programming; Monte Carlo optimisation; simulated annealing; self-avoiding walks. Biological problems featured in this course include. sequence alignment; domain assignment; structure comparison; comparative modelling; protein folding; fold recognition; finding channels; molecular docking; protein design. Describe and summarise problems that have been addressed in the bioinformatics literature, and computational approaches to solving them. design and implement computational solutions to problems in bioinformatics. critically discuss different bioinformatics methods that address the same task or related tasks, and to discuss differences in the tasks addressed, or differences in the computational approaches. identify situations where the same computational methods are

		applied in addressing different problems, even across different application areas
19	**Computer networks**	learning experiences that involve hands-on experimentation and analysis as they reinforce student understanding of concepts and their application to real-world problems. API programming for fault-tolerance network systems, and Internet interconnections and services from a practical perspective, and protocols' design and analysis with a strong emphasis on self-stabilizing algorithms. fundamental issues in the design of methods for computer network protocols. Describe and analyze basic protocols and their limitations on networks such as the Internet. analyze and discuss network issues, such as software-defined networks (SDNs), TCP connections, contention, performance, and flow control. show the ability to define and analyze a computer network in terms of communication graphs and as a distributed system. critically to analyze the effect of failures, such as transient faults, message omission, and topology changes, on the system and how can such failures propagate and affect computer networks. develop small-scale network applications using fundamental

		networking techniques.design and develop your own network-oriented program and then test and demonstrate it in the lab. The written communication skills, the write up of lab reports and the demonstration of protocol correctness. explain and demonstrate the correctness of the studied protocol as well as clearly describe the network algorithms. skillful and knowledgeable demonstration of these software developments for advanced fault-tolerant client-server and peer-to-peer architectures. design distributed algorithms for computer networks and to show why they work. . describe, design and analyze existing and new algorithms for network protocols with a very strong emphasis on self-stabilizing algorithms for computer networks.
20	**Computer vision**	Projective geometry. Geometric transformations. Modelling of cameras. Feature extraction. Robust estimation. Minimal solvers in computer vision. Stereo vision. 3D-modelling. Rigid and non-rigid structure-from-motion. Bundle adjustment. Geometry of surfaces and their silhouettes Be able to clearly explain and use basic concepts in computer vision, in particular regarding projective geometry, camera modelling, stereo vision, and structure and motion problems. be able to describe and give an informal explanation of the mathematical theory behind some central algorithms in computer vision (the least squares method and Newton based optimization). . in an engineering manner be able to use computer packages to independently solve problems in computer vision. be able to show

		good ability to independently identify problems which can be solved with methods from computer vision, and be able to choose an appropriate method. be able to independently apply basic methods in computer vision to problems which are relevant in industrial applications or research. with proper terminology, in a well-structured way and with clear logic, be able to explain the solution to a problem in computer vision.
21	**Data-driven support for cyber-physical systems**	The content is focused on distributed computing and systems, data processing, information and systems security, networking and computer communication in the context of new cyber-physical systems. There are lectures from faculty to give an overview of the areas of the course, and invited presentations from industry to talk about actual systems, as well as in-depth presentations by the student themselves on specific research topics relating to their projects. Typically, the lectures include an introduction to the new types of cyber-physical systems, e.g. the smart grid. Open research problems in relation to distributed operations, data-processing and cyber security are discussed, e.g. through lectures on streaming, security and privacy, and communication suitable in this domain. Examples of cyber-physical systems important for society are presented, e.g. the smart grid from both on the transmission and distribution perspective. List cyber-physical systems, and in particular ICT methods for supporting adaptiveness, decentralization and cybersecurity based on the

		students chosen area. Discuss current research and development in the area of such cyber-physical systems, in order to meet the requirements of sustainable development in (security, economic and ecological terms). Design and analyse methods, algorithms, protocols for adaptive and cybersecure cyber-physical systems, such as smart power grid networks. Explain complex algorithms and concepts. Plan and organize a small team project and document the work and the result in a report. Identify, combine and use own and others' resources, and deal with uncertainty, with the aim of creating value for others. Relate to idea development through evaluation and selection of ideas, presenting ideas and implementing ideas in relevant context(s). Present complex material to a small audience. Improve skills in running a small team project, practice technical writing. Judge the relevance of the literature in a topic. Reflect on own and others' abilities and roles in relation to the project work - examined in a peer assessment combined with one's own reflections.
22	**Design of AI-systems**	design of AI systems in several different ways. different AI systems and their design (eg. AlphaZero, Watson, systems for self-driving cars). implementation of different simpler AI systems. simpler AI systems. possibilities and limitations of AI, ethics and societal impact. Provide an overview of different applications of AI and related areas. Describe how some different well-known AI-systems work and how they are used. Explain how AI approaches relate to other kinds of advanced

		information processing. Identify problems that can be solved with AI and other advanced computational techniques. Design simpler Ai systems for different applications, including model choices and system design. Implement AI systems with programming in combination with different tools and programming libraries. Discuss advantages and disadvantages of different models and approaches in AI. Reflect over fundamental possibilities and limitations of current AI approaches. Critically analyze and discuss AI applications with respect to ethics, privacy and societal impact. Show a reflective attitude in all learning.
23	**Distributed systems**	basic concepts of distributed systems and the challenges they pose. the required background in communication systems and operating systems. Naming. Mutual Exclusion and Election. Clocks and Time. Consistency and Replication. Fault Tolerance in Distributed Systems. Selected Applications in Distributed Systems. required fundamentals, experience in developing distributed systems and exploring their real-world challenges. learning experiences that involve hands-on experimentation and analysis. develop an understanding of fundamental issues in the design of methods for distributed systems. Recall and apply knowledge of basic concepts of distributed systems and their challenges, naming and synchronization of systems, consistency and replication, and fault tolerance in distributed systems. Describe applications of distributed systems and the mechanisms these use to provide

		their services. Discuss and analyze the challenges and requirements that the different approaches have. Compare and summarize the strength and weaknesses associated with the individual mechanisms. Develop and evaluate small-scale distributed systems using fundamental mechanisms introduced in the lectures. Demonstrate software developments in advanced settings including unreliable links and systems as well as limited bandwidth. Demonstrate lab results in oral and written presentation. Describe and analyze existing and new methods for distributed systems design. In particular, the systems ability for scalability and fault tolerance. Discuss and value the social and ethical aspects of distributed systems and their applications.
24	**Empirical software engineering**	empirical methods applied to the field of software engineering. quantitative and qualitative methods in software engineering with accompanying statistical methods used for analysis. Descriptive and inferential statistical methods applied to software engineering. Conducting qualitative and quantitative methods in software engineering. Methods for analysing quantitative and qualitative data in software engineering. Usage of statistical tools. Describe, understand, and apply empiricism in software engineering. Describe, understand, and partly apply the principles of case study research , experiments , surveys. Describe and understand the underlying principles of meta-analytical studies. Explain the importance of research ethics. Recognise and

		define code of ethics for when conducting research in software. engineering. State and explain the importance of threats to validity and how to control. said threats. Describe and explain the concepts of probability space (incl. conditional probability), random variable, expected value and random processes, and know a number of concrete examples of the concepts. Describe Markov chain Monte Carlo methods such as Metropolis. Describe and explain Hamiltonian Monte Carlo. Explain and describe multicollinearity, post-treatment bias, collider bias, and confounding. Describe and explain ways to avoid overfitting. Assess suitability of and apply methods of analysis on data. Analyse descriptive statistics and decide on appropriate analysis methods. Use and interpret code of ethics for software engineering research. Design statistical models mathematically and implement said models in a programming language. Make use of random processes, i.e., Bernoulli, Binomial, Gaussian, and Poisson distributions, with over-dispersed outcomes. Make use of ordered categorical outcomes (ordered-logit) and predictors. Assess suitability of, from a ontological (natural process) and epsitemological (maxent) perspective, various statistical distributions. Make use of and assess directed acyclic graphs to argue causality. State and discuss the tools used for data analysis and, in particular, judge their output. Judge the appropriateness of particular empirical methods and their applicability to attack various and disparate software

		engineering problems. Question and assess common ethical issues in software engineering research. Assess diagnostics from Hamiltonian Monte Carlo and quadratic approximation using information theoretical concepts, i.e., information entropy, WAIC, and PSIS-LOO. Judge posterior probability distributions for out of sample predictions and conduct posterior predictive checks.
25	**Introduction to data science and AI**	Introduction to data science. Implementation of data science solutions, using Python, basic data analysis and visualization. Introduction of the data science process, and appropriate methodology. Examples of core data science methods with case studies such as in clustering, classification and regression. Data science put in context regarding ethics, regulations and limitations. Statistical methods for data science and AI. Introduction of some common stochastic models with examples of applications in data science and AI (for instance, naive Bayes classifiers, topic models for text and Hidden Markov Models for sequence data). Artificial Intelligence. Introduction to classical AI and machine learning, including the relationship to related areas such as algorithms and optimization, and AI philosophy. Examples of methods and applications of AI, in classical AI (search and constraint satisfaction), and ML-based (search engines, naive Bayes and neural networks). Discussion of ethics and societal impact of AI. Describe fundamental types of problems and main approaches in data science and AI. give examples of data science and AI applications from different

		contexts. give examples of how stochastic models and machine learning (ML) are applied in data science and AI. explain basic concepts in classical AI, and the relationship between logical and data driven, ML-based approaches within AI. briefly explain the historical development of AI, what is possible today and discuss possible future development. use appropriate programming libraries and techniques to implement basic transformations, visualizations and analyses of example data. identify appropriate types of analysis problems for some concrete data science applications. implement some types of stochastic models and apply them in data science and AI applications. implement and, or use AI-tools for search, planning and problem solving. apply simple machine learning methods implemented in a standard library. Justify which type of statistical method is applicable for the most common types of experiments in data science applications. discuss advantages and drawbacks of different types of approaches and models within data science and AI. reflect on inherent limitations of data science methods and how the misuse of statistical techniques can lead to dubious conclusions. critically analyze and discuss data science and AI applications with respect to ethics, privacy and societal impact. show a reflective attitude in all learning.

26	**Introduction to game research**	games in all their forms as well as theoretical. concepts and frameworks to analyze games. different forms of games and different perspectives of gaming. games as systems and focus on board games and card games. introduces general concepts to describe games and gaming. how players perceive and immerse themselves into games, using role-playing and. larps to highlight the play experience as a perceptual stance. how different media forms impacts on games and gaming by focusing on computer. games and online games. how the boundaries between games. and other activities can be obscured by the game design and how games can be used for. other purposes than to entertain, for example to criticize, influence, or teach. theoretical concepts and frameworks through academic texts and. builds on the previous part. Know the academic game terms. show an understanding of different types and approaches to classifying games. show an understanding of different academic approaches to researching games and gaming. be able to explain what characterizes games within the most common classifications. analyze games given a specific research question, research stance, and academic vocabulary. describe games given a specific focus and showing an adequate use of academic game terms. make comparisons between games or parts of games through the use of academic game terms. analyze games in relation to various intended uses. analyze games from several different gaming preferences. be able to choose and combine different academic approaches in order to

		analyze and interpret games given a specific context. identify ethical aspects of a game.
27	**Operating systems**	Introduction to the design and implementation of operating systems. concurrent processes, resource management, deadlocks, memory management techniques, virtual memory, processor scheduling, disk scheduling, file systems, distributed file systems and micro kernels, virtual machines and security and protection schemes. key components of operating systems, design and implementation challenge and their evolution from pioneer to modern mobile-based ones. Examples include Unix, Linux, Windows, mobile-devices operating systems. The core functionality of modern operating systems. Processes , threads, scheduling, virtual memory and file systems, aspects of parallelism, kernels, shells, micro kernels, virtual machines. Key concepts and algorithms in operating system implementations. synchronization, deadlock-avoidance , prevention, memory

		management, processor scheduling, disk scheduling, virtual machines, file systems organization. Implementation of simple OS components. appreciate the design space and trade-offs involved in implementing an operating system. Write C programs that interface to the operating system at the system call level. Implement a piece of system-level code in the C programming language. some programing using multithread synchronization constructs.
28	**Principles on concurrent programming**	Physical vs logical parallelism. Concurrency problems. race conditions, interference, deadlock, fairness, livelock. Mutual exclusion. Shared memory synchronization. using semaphores or fine grained locking. Message-passing synchronization. using message queues. know the academic game terms. show an understanding of different types and approaches to classifying games. show an understanding of different academic approaches to researching games and gaming. be able to explain what characterizes games within the most common classifications. analyze games given a specific research question, research stance, and academic vocabulary. describe games given a specific focus and showing an adequate use of academic game terms. make comparisons between games or parts of games through the use of academic game terms. analyze games in relation to various intended uses. analyze games from several different gaming preferences. be able to choose and combine different academic approaches in order to

		analyze and interpret games given a specific context. identify ethical aspects of a game.
29	**Quantum computing**	Elementary quantum gates and basic quantum computing formalism. Introduction to complexity classes and relevant conjectures. Circuit model for quantum computation. Foundational theorems for quantum computation. Solovey Kitaev theorem; Gottesman-Knill theorem. Other models for universal quantum computation beyond the circuit model. Measurement Based Quantum Computation and Adiabatic quantum computation. Quantum Fourier Transform and Phase estimation algorithms. Shor's algorithm. Quantum Machine Learning. Quantum Cloud Computing exercise. Quantum algorithms for solving combinatorial optimization problems. quantum annealing and QAOA. Variational quantum eigensolver. Sampling models. Boson sampling and Instantaneous Quantum Polynomial. Continuous-Variable (CV) quantum computation, measurement-based quantum computation in CV and GKP encoding. List modern relevant quantum algorithms and their purposes. Explain the key principles of the various models of quantum computation (circuit, measurement-based, adiabatic model). Explain the basic structure of the quantum algorithms addressed in the course that are based on the circuit model, and to compute the outcome of basic quantum circuits. Compare, in terms of time complexity, what quantum advantage is expected from the quantum algorithms

		addressed in the course with respect to their classical counterparts. Program simple quantum algorithms on a cloud quantum computer or a cloud simulator. Understand the basic principles of the continuous variable encoding for quantum information processing. Give examples of the motivation for applying quantum computing to machine learning and of what the obstacles are to achieving an advantage from doing so.

The American University's HPC MSc programme

	Course	DETAIL
1	**High Performance Computing**	Overview of CPU and GPU Architectures. Instruction sets. Functional units. Memory hierarchies. Performance Metrics. Latency and bandwidth. Roofline modeling. Single-core optimization. Compiler-assisted vectorization. data-level parallelism. Design patterns for cache-based optimization. Multi-threaded CPU programming. Worksharing, synchronization, and atomic operations. Memory access patterns, including non-uniform memory access. The OpenMP API. GPU programming. Thread-mapping for optimal vectorization and memory access. Task-scheduling for latency reduction. The CUDA and OpenMP offload APIs. Distributed parallelism. Synchronous and asynchronous communication patterns. Data decomposition. Hybrid models for distributed multi-threaded and GPU programming. The MPI API.
2	**Bioinformatics for Computer Scientists**	Genomics, Bioinformatics and Molecular Biology. A high-level view of increasingly important role of computing in the biological sciences will be presented. Genomes, Sequences and Databases. A survey of the current state of the art in storing, organizing and analyzing large data sets will be discussed. The advantages and disadvantages of these methods will be explored in the context of academic and commercial research initiatives. Sequence Alignment. Fast, reliable alignment of text strings started the bioinformatics revolution. Protein Structure and Function. Spatial assembly and interactions of proteins support life and cause of disease. Protein Motifs and Modeling. Understanding protein function holds the promise developing therapeutics and curing diseases, but the computational complexity of analyzing

		three-dimensional models presents obstacles that have been difficult to overcome. shape analysis and comparison that can be scaled to large data sets. High-Performance Computing for Bioinformatics. Strategies for conducting large-scale analysis of genes and proteins will be presented. Microarray Data Analysis. The technologies used to power these services will be introduced as well as the different approaches used to provide web services to analyze the data. SNPs and Disease. trace the genetic origin of disease. different approaches to cataloging and analyzing these changes. In Silico Drug Discovery. Approaches to using computer models to develop new drugs will be presented.
3	**Introduction to Scientific Computing**	software engineering. data mining. high-performance computing. scientific communication. mathematical models and other areas of computer science.
4	**Numerical Methods**	Properties of Linear Systems. Gaussian elimination with backsubstition. pivoting, stability, operation count, implementing in matlab, C, and Fortran. Floating point representation, roundoff error, stability. Real world linear systems. introduction to first piece of model app. 1d implicit heat equation. Efficient Implemention of implicit 1D heat equation solvers, LU decomposition. 2-3d version. Eigenvalues/vectors: basic theory. Eigenvector factorization. Computing largest Eigenvalue/eigenvector numerically. power iteration. Gramm Schmidt. approaches to calculating eigenvalues numerically. Norm, condition number, stability. QR, rotation matrices, similarity transformations. Complex eigenvalues/eigenvectors. Method of Steepest Descent. Jacobi iteration, Gauss-Seidel, and SOR. Introduction to

		Krylov Solvers. Pre-conditioned Conjugate gradient method. Efficient coding of sparse matrix multiply. Survey Topics. Discrete Fourier Transforms. Systems of ODEs. Proper Orthogonal Decomposition.
5	**Parallel Programming**	Processes and threads. Shared memory. Hardware mechanisms for parallel computing. Synchronization and communication for parallel systems. Performance optimizations. Parallel data structures. Memory consistency and hierarchies for parallel computing. Patterns of parallel programming. Parallel programming on GPUs.
6	**Time Series Analysis and Stochastic Processes**	Overivew of fundamentals of probability. CDFs, PDFs, Central Limit Theorem. Numerical sampling from discrete PDFs and continuous PDFs. Time series models. Principal component analysis and singular value decomposition. Spectral analysis including Fourier transforms. Issues in random number generation. Simulating discrete events. Monte Carlo integration with variance reduction. Markov Chain Monte Carlo. Hastings Metropolis, Gibbs Sampler.
7	**Machine Learning**	data mining, machine learning, and statistical modeling, and the practical know how to apply them to real-world data through Python based software. supervised and unsupervised learning. linear and logistic regression and regularization. classification using decision trees, nearest neighbors, naive Bayes, boosting, random trees, and artificial and convolutional neural networks. clustering using k-means and expectation maximization. dimensionality reduction through PCA

		and SVD. Python and Python libraries such as NumPy, SciPy, matplotlib, and pandas for for implementing algorithms and analyzing data.
8	**Applied Data Analysis**	Elementary Probability Statistics. Probability theory. Random variables. Distributions and densities. Software Platforms. Variables, objects, and functions in Python. Working with data frames. Data pre-processing and visualization. Linear Models/Statistical Inference. Least-squares regression. Logistic regression. Hypothesis testing. Model Assessment and Selection. Machine Learning Models. Perceptron classifier. Neural networks. Decision trees and Random forests. Project pitches. Clustering. Unsupervised clustering. Supervised clustering. Decision trees/Random forests. Support vector machines. Computational Frameworks. Common machine learning frameworks. Big data analytics.
9	**C Programming**	Types, Control Flow, Functions and Program structure. Build tools, preprocessor, compiling and linking. Debugging. Unit testing. Pointers and Arrays. Recursion. Structures, Unions, Bit-Fields, Typedef. Data structures. linked lists, stacks, queues, sets, hash tables, trees, heaps. Algorithms. coding classical sorting and searching algorithms, algorithmic analysis. Multicore Programming. threads & synhronization.
10	**Advanced Programming**	C language. Trees. Hash tables. Union finds. Graphs. Intro to multicore and pthreads programming model. Applications of multicore. Alternative approaches. java task-based, OpenMP, etfc. Building larger codes. software engineering, abstraction, development tools, compilers.

11	**Compilers**	compiler structure. basics of compiler structure and review relevant aspects of computer architecture. Lexical analysis and parsing. textual programming language is processed and introduce tools that make this job easier. Intermediate representations. control flow, dominance, intermediate representations, and related topics to understand how a program is represented inside of the compiler. LLVM. use LLVM to create a compiler. Pointers and optimizations. pointers and data-flow and lattice algorithms, plusfunction inlining and peephole optimizations. loop optimizations. Lower-level things. vectorization, instruction selection, and register allocation. Virtual machines. virtual machines, garbage collection, and related topics. Javascript engine. Debugging. debugging technology, including instrumentation based debugging. LLVM sanitizers.
12	**Introduction to Computer Systems**	Boolean logic, combinatorical chip design, Karnaugh maps, hardware description language. Use a hardware description language to build a basic chip set. Sequential chip design, binary arithmetic. Use a hardware description language to build a sequential chip set and the ALU, CPU, memory for a computer. Machine language, computer architecture. Write and run programs in assembly language. use a hardware description language to build a working computer. Assemblers. Design, implement, test, and debug an assembler, using a programming language. Virtual machine paradigm, stack arithmetic. Design, implement, test, and debug a virtual machine translator for stack arithmetic and memory access commands, using a programming language. Virtual machine language program control. Add function definition, and function call and return commands to the virtual machine translator. high-level object-

		oriented programming languages. Compilers, syntax analysis. Design, implement, test, and debug a tokenizer for an object-oriented language. compiler, using a programming language. Compilers, code generation, Operating systems. Design, implement, test, and debug a code generator for an object-oriented language compiler, using a programming language.
13	**Advanced Computer Systems**	Representing and Manipulating information. unsigned and two's complement representation, IEEE floating point and corresponding arithmetic. Machine level representation of programs. x86-64 assembly, control instructions, translation of basic C control constructs, such as loops and switch statements,common code security vulnerabilities, such as buffer overflows. Processor architecture. pipelined out of order processor. Code optimizationMemory hierarchy. persistent storage. magnetic spinning disks, SSD, RAM and ROM, and caches. Virtual Memory.
14	**Computer Architecture**	design and performance evaluation of modern computer architectures. microprocessors, chip-multiprocessors and memory hierarchy design, including parallel, multicore CPUs. Memory cache designs and optimizations. DRAM technologies. Instruction and data pipelining. Branch prediction for instruction- and data-level paralelism. Dynamic scheduling for instruction and data level parallelism. Mulithreading support in hardware and operating system. Data coherency for efficient multithreading.

		Non-uniform memory access for efficient multithreading.
15	**Operating Systems**	basic concepts and techniques used to implement operating systems. processes and threads, interprocess communication and synchronization, memory management, segmentation, paging, linking and loading, scheduling, file systems, and input/output. implementation of Linux/Unix operating systems. fundamental concepts in operating systems, including processes and threads, interprocess communication and synchronization, memory management, segmentation, paging, linking and loading, scheduling, file systems, and input/output. implementation of an x86 operating system kernel, Pintos instructional kernel. implementing higher-level operating system functionality, thread management, memory management. Threads. interact with the OS via system calls. implementing a virtual memory management system. improve Pintos basic file system.
16	**Distributed Systems**	theory and practice of distributed systems. distributed systems, understand characteristics of distributed systems. review the unique challenges of distributed systems, analyze solutions for common distributed systems problems, and gain practical knowledge of the systems and algorithms for building real distributed systems. Distributed architectures. Processes and threads. Networking and communication mechanisms. Naming and mapping. Synchronization. Distributed time and ordering. Consistency. Fault tolerance.

		Distributed consensus. Distributed data. Data intensive computing.
17	**Advanced Algorithms**	design and analysis of efficient algorithms. solve algorithmic problems in the real world. data structures. Bloom filters, red-black trees, skip lists. discrete and continuous optimization. advanced dynamic programming, gradient descent NP-hardness and methods of dealing with intractability. approximation, randomization, backtracking, branch and bound, local search. computational geometry and advanced graph algorithms with applications in computing, data science, and engineering. designing new algorithms in pseudocode and implementing selected algorithms in Python. tackle algorithmic problems they are likely to encounter in software development.
18	**Cloud Computing**	Distributed computing and clouds. Cloud computing service models and economics. Virtualization and Infrastructure-as-a-Service. Application concurrency and data consistency models. Identity and access control in cloud computing systems. Cloud storage systems. Characteristics and tradeoffs. Inter-service communication. Messaging and notifications. Cloud security, privacy, and policy compliance. Serverless computing, microservices and containerization. Automated deployment and operations techniques.
19	**C++ for Advanced Programmers**	C++. Classes and Object-orientation. C++ Libraries including Boost. Overloading, memory management, and their associated consequences and idioms. Low-level and performance

		programming. Templates, specialization, and concepts. Concurrency in C++, including cache-conscious programming. Type traits and metaprogramming.
20	**Advanced C++**	C++. build enterprise-scale software. broad feature set of C++ fits together to enable powerful programming paradigms. use metaprogramming techniques to implement Software Engineering Design Patterns with the aim of making a metaprogram that simply executes design documents. Advanced concurrency techniques like futures and promises with best practices to avoid the myriad pitfalls of multi-threaded programming. Advanced template techniques like SFINAE, CRTP, and variadics. implementing full featured classes like tuples. Using metaprograms to implement seamlessly Embedded Domain Specific Languages and Design Patterns. read and leverage the C++ standard. Creating STL-style iterators. Customizing I/O streams.