Tampere University

Cuong Nguyen

BIBLIOMETRICS

Integrating the Computing Sciences Unit of Tampere University into CSRankings

Bachelor of Science Thesis
Faculty of Information Technology and Communication Science
Examiners: Prof. Billy Bob Brumley
M.Sc. Nicola Tuveri
April 2023

ABSTRACT

Cuong Nguyen: Bibliometrics
Bachelor of Science Thesis
Tampere University
International Bachelor degree of Science and Engineering
April 2023

Universities have gained increasing influence on the social development through their teaching and research duties. Thus, it is crucial to have a concrete approach to evaluate the academic performance of universities. Besides, as a result of the globalization, a benchmark tool for universities in the international level is unavoidably needed. There are global university rankings that have been widely referenced in recent years. However, many of them are reputation-based rankings which use biased methodologies and subjective data for evaluating the academic performance of universities. To solve this problem, metric-based rankings which utilize transparent approaches and objective data appeared. In this thesis, I focus on CSRankings, a metric-based ranking which ranks universities in the computer science field. In addition, I integrated the Computing Sciences Unit of Tampere University within the Faculty of Information Technology and Communication Science to the database of CSRankings. As a result, Tampere University is now presented in CSRankings; thus, prospective students, researchers, and investors can have better tools to evaluate the academic performance of Tampere University in the international level within the computer science field.

Keywords: CSRankings, dblp, university ranking, metric-based ranking, GOTO-ranking

PREFACE

This Bachelor's thesis represents the culmination of my undergraduate studies in the International Science and Engineering program, majoring in Signal Processing and Machine Learning and minoring in Information Security, at Tampere University. I feel fortunate to have worked with my supervisors, Prof. Billy Bob Brumley and M.Sc. Nicola Tuveri, on the project that has helped me to deepen my knowledge in the computer science field.

I would like to express my gratitude to Prof. Billy Bob Brumley and M.Sc. Nicola Tuveri for their guidance and tremendous support throughout the research and writing process. Their insightful comments and feedback have been invaluable in shaping the content and structure of this thesis. Moreover, I would also like to thank all NISEC members who have inspired me to pursue a career in information security through their expertise, dedication, and passion.

Tampere, 26th April 2023

Cuong Nguyen

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF ABBREVIATIONS

AES	Academic Excellence Survey
CD	Continuous Delivery
CI	Continuous Integration
CS	Computing Sciences
CSV	Comma-separated values
CVCS	Centralized Version Control System
CWTS	Centre for Science and Technology Studies
DVCS	Distributed Version Control System
GRAS	The Global Ranking of Academic Subjects
HTTP	Hypertext Transfer Protocol
NTU	National Taiwan University
PID	Person ID
PR	Pull Request
SHA1	Secure Hash Algorithm 1
TAU	Tampere University
TOCs	Table of Contents
URL	Uniform Resource Locator
URPA	University Ranking by Academic Performance
USN&WR	U.S. News and World Report
VCS	Version Control System
WoS	Web of Science
XML	Extensible Markup Language

1. INTRODUCTION

I joined the *Bibliometrics* project through the *Research Trainee* program in which Bachelor students can freely choose an available project to obtain experience in a research environment and earn credits. The first stage of the *Bibliometrics* project is to integrate personnel of the Computing Sciences (CS) Unit within the Faculty of Information Technology and Communication Science at Tampere University (TAU) into the CSRankings database. I selected this project as it matched my academic background and caught my interest as well. In later paragraphs, I will introduce the motivation of this work and outline the structure of this thesis.

For ages, universities have played an essential part in the evolution of civilizations overseas through their teaching and research missions [46]. In addition to performing these duties, they also design development strategies and contribute in increasing graduate employment, raising the standard of education in society, expanding opportunities for individuals, and developing knowledge and technology [46]. Universities that are conceptualized as knowledge-explorers, along with enterprises known as knowledge-exploiters, generate new regional inventive capacities [4].

As higher education institutions play a critical role in the social development, it is necessary to have a robust method to measure their academic performance. Hence, university rankings have been used since 1925, based on reputation of universities within a nation [60]. In addition, the third era of globalization started in 1989 and continues today, leading to the need of worldwide university rankings. Hence, global university rankings first appeared in 2000s. For example, The Academic Ranking of World Universities (ARWU) was first published in 2003 followed by THE (Times Higher Education) World University Rankings in 2004. They have flourished in recent years in spite of the criticism about their biased methodologies and objectives [58, 32]. As a consequence of this proliferation, global university rankings influence the means of people — policy makers, prospective students, investors, and university presidents — evaluating the quality of higher education institutions [33, 31, 46]. In addition, university rankings have become an indicator of economic competitiveness among countries since the more universities in a particular nation or region that are rated among top 10, 50, or 100, the better the country's or territory's economic repute and inventive potential [36]. As a result, both developed and developing countries have invested a large portion of their annual Gross Domestic Prod-

uct (GDP) into higher education and research and development (R&D). For example, the European Union (EU) spent 328 billion euros on R&D, 2.27% of GDP in 2021, while China and United States expended 2.4% and 3.45% respectively [21]. It can be inferred that high-ranked universities in global rankings are likely to receive greater grants or investments. Another impact of rankings on universities is that these benchmarks encourage the internationalization, globalization, or multi-regional collaboration among educational institutions as many rankings introduced the internationalization as a primary factor [30]. In order to promote internationalization, several universities have started to use English as their main instructional language [60]. In addition, universities are encouraged to attract international academic staff and students to not only boost their rankings but also increase their financial grants. With respect to prospective students, university rankings are a vital factor they consider when they are choosing a university [6]. Moreover, rankings are an effective tool for international students to select an university to study abroad as it is challenging to visit an institution based overseas. Given the above arguments, it is reasonable to state that global university rankings have a significant impact on national systems, higher education institutions, and students.

However, major global university rankings — Academic Rankings of World Universities (ARWU), Times Higher Education (THE) World University Rankings, US News Best Global Universities Rankings — are reputation-based rankings whose methodology is subjective and unreliable. Ranking indicators utilized by ARWU and THE rankings are most direct contributors to reputational bias [57]. Hence, in this thesis, CSRankings, a metric-based ranking, is examined. CSRankings evaluates academic performance of institutions in the computer science field based on their contributions at top conferences. Furthermore, the process of merging personnel of the CS Unit of TAU into CSRankings is also presented in this thesis. Once the CS Unit of TAU is available on CSRankings, their academic performance will attract more prospective students and researchers to join their research community.

The remainder of the thesis is structured as follows. Chapter 2 describes the motivation, methodology, and technical background of CSRankings and its supporting software. In Chapter 3, I will introduce The Global Ranking of Academic Subjects (GRAS) and emphasize the similarities and differences between it and CSRankings. Chapter 4 presents in detail the process of integrating the CS Unit of TAU into CSRankings. Chapter 5 provides the final result of the integration process and data analysis. Finally, Chapter 6 summarizes the contents of mentioned chapters.

2. BACKGROUND

Before we go to the implementation section, we need to be familiar with some core concepts: GOTO ranking, CSRankings, dblp, and Version Control with Git.

GOTO ranking is a metric-based ranking methodology which CSRankings bases on. In addition, CSRankings assesses higher education institutions based on the number of academic papers published at top computer science conferences. To access the database of bibliographic information on these conferences, CSRankings uses dblp, an open bibliography provider. Besides, CSRankings is an open source project which we can contribute to through a specific state capture mechanism. In this case, Git is the version control system that is used by CSRankings to manage snapshots or states along the timeline of it.

2.1 GOTO ranking

Higher education institutions are mostly ranked by for-profit organizations [5]. Hence, the business-first goal of those ranking organizations is inevitable. For-profit ranking enterprises usually tune their methodology in order to make variation in the rankings. The movement in the rankings is needed for making the business viable [64]. Furthermore, the methodology of certain rankings is highly limited. These rankings, reputation-based rankings, depend solely or partly on the reputation surveys which have several limitations. The first disadvantage is that the reputation is prone to change slowly, because it can take years for the reputation of a university to be updated when a department improves [5, 56]. In addition, the assessment, in which department chairs and graduate directors are asked to evaluate each program on the scale of 1 to 5, is subjective [64]. The scores are made based only on the personal opinion without objective data, not showing any indication of whether productivity or reputation is measured [60]. In addition, reputation-based rankings have a considerable influence on assessments responded by faculty members about institutions' reputation, which is known as the anchoring theory: current judgements may be influenced by prior evaluations that were represented in rankings, resulting in duplicated and reinforced evaluations [7, 20]. According to [7], the anchoring theory states that "people use the starting value to inform their judgements, and then they adjust (insufficiently) this value when making their final judgement, even when the starting value is

entirely random". As the following ranking anchors the observers' assessments of each higher education institution, the anchoring theory has a substantial impact on shaping perceptions based on the initial rankings as well as impacting current perceptions. Along with the anchoring theory, the echo effect may have impact on consecutive rankings: universities affect the judgements that are successively used to produce following rankings by using their ability to simplify, and their effective communication networks [57]. Therefore, the subsequent assessments of the reputation of universities can be misled as the initial reviews can be reinforced and widely broadcasted by the media [55]. A circle known as "reputation-ranking-reputation" results when the anchor effect is repeated dynamically and the echo effect is introduced [57]. In other words, it is possible that the position of institutions in many reputation-based rankings does not illustrate the current quality but their prior reputation that has been gained since a long time ago. Furthermore, the halo effect may have a biased influence on the reputation surveys as well. The halo effect, or the halo error, is a cognitive bias in which individuals generate opinions about a quality or attribute of a product depending on their propensity toward another aspect [45]. For instance, rather than utilizing ranking factors that are closely related to the performance in terms of teaching, research, or the so-called third mission including promoting innovation, entrepreneurship, knowledge transfer and social commitment, several rankings are based on metrics that are connected to the size and age of institutions [58]. As a result, the halo effect of renowned universities' parent institutions raises the ranks of their graduate programs [64].

To address aforementioned disadvantages of reputation-based rankings, a new model of ranking factor should be introduced. As mentioned in [5], the Computing Research Association (CRA) has suggested developing new methods which are data-driven, based on meaningful metrics, and at least meet the following criteria:

- **G**ood data: have been cleaned and curated.
- **O**pen: data is available, regarding attributes measured, at least for verification.
- **T**ransparent: process and methodologies are entirely transparent.
- **O**bjective: based on measurable attributes.

The requirement of **G**ood data means that data should be accurate and reflect correctly how research is disseminated in a particular scientific community. For example, in the computer science area, conference publications are the most influential and highly cited peer reviewed articles. However, in 2017, U.S. News and World Report (USN&WR) conducted a ranking of all computer science departments over the world based on journal publications, ignoring academic papers published at conferences. As a result, the output ranking did not reflect precisely how research works were propagated in the computer science community or how academics were rewarded or had an influence [1]. Although objective data was used in this case, the ranking was still implausible due to the shortage

of data coverage. In order to address this issue, in 2019, USN&WR started including conference publications which contributed 7.5% among 12 ranking factors in total [44].

Regarding to the criteria of **O**peness and **T**ransparence, the data used for assessing institutions should be publicly accessible at least for verification from other neutral parties. Furthermore, the methodology should be transparent such that an external user using open source data and published method can reproduce the final result. Following the USN&WR example above, USN&WR used bibliographic data provided by Web of Science (WoS) which is a paid-access platform. Hence, only fee-paying users can access the database. Moreover, the list of venues is not public. It could be seen that is an example of closed source data and questionable methodology.

In terms of the **O**bjectiveness of data, all quantitative indicators should be meaningful metrics and have a logical foundation for comparisons. For instance, publications, citations, honors, and funding are considered objective data [5].

At the time of writing, there are three GOTO rankings for the computer science field: CSMetrics, CSIndexbr, and CSRankings. Without regard to departmental organization or authors' job titles, CSMetrics is a quantitative metric-based ranking focusing on the institution as a whole. CSIndexbr provides transparent and meaningful statistics about the Brazillian scientific production in computer science field. CSRankings is a metric-based ranking that is faculty-centric and based on publications at selected top conferences. There is no a perfect ranking which serves all purposes effectively; thus, those GOTO rankings are created to support each other and solve different types of issues. Among those three GOTO rankings, CSRankings is the one which I focus on in this thesis.

2.2 CSRankings

In terms of measuring the quality of academic papers, citation-based metrics are considered suitable tools provided that they are accurate and used with care and competence [43]. The logical basis is that papers with a high number of citations have a greater influence and consequently higher value [3]. However, this principle is found unreliable and prone to control. A study shows that 32% of the group of highly cited articles on clinical trials produced results which were later contradicted [35]. Additionally, citation data is not freely available, and studies showed that bibliometric indicators offered by citation measuring systems, e.g., Google Scholar, may be easily manipulated [19, 38]. On top of this, there is a phenomenon known as "citation cartels": small groups of researchers band together to cite each other's papers, misleading the citation system into assessing that their publications are highly influential [47]. This phenomenon could affect the scientific community by rewarding quantity over quality [47].

Avoiding the mentioned disadvantages, CSRankings ranks higher education institutions

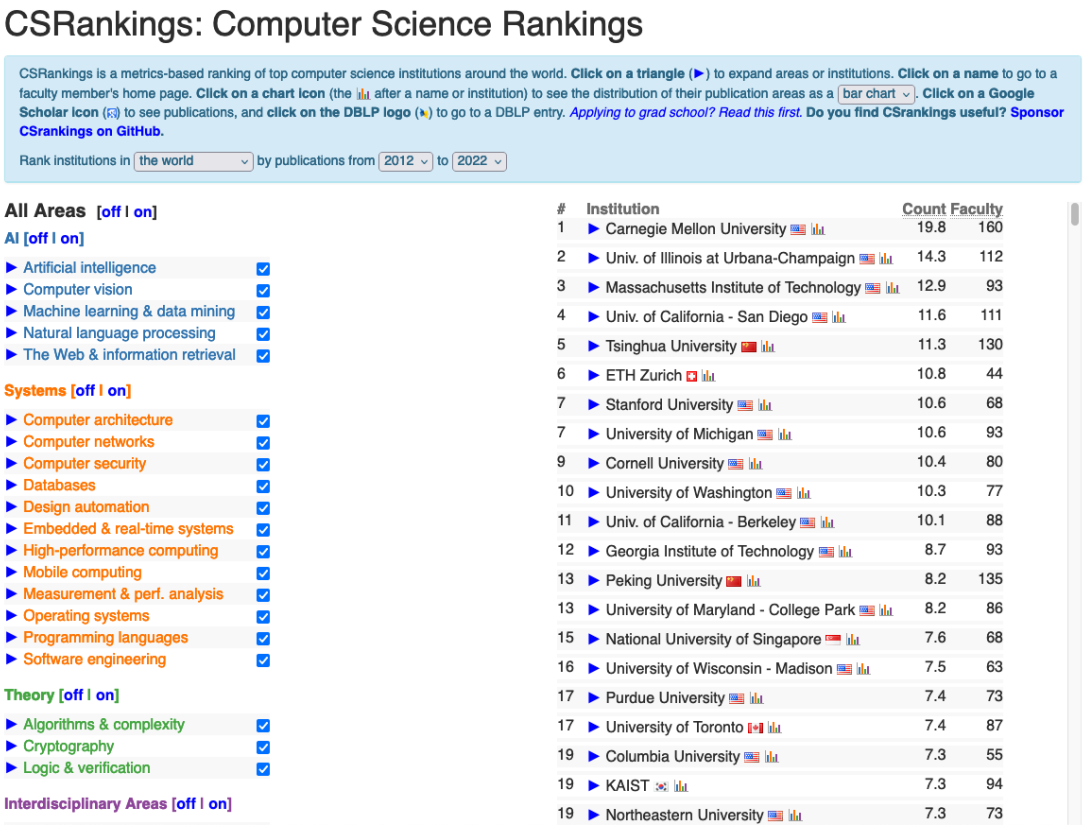


Figure 2.1. *Institution ranking within all areas in the world.*

by their presence at top esteemed conferences. This approach is considered not only incentive-aligned but also difficult to game, encouraging faculties to publish high-quality articles at top venues [13]. Furthermore, as a GOTO ranking, CSRankings is entirely metric-based and transparent; therefore, all source code and data are published so that anyone can freely access through this link: https://github.com/emeryberger/csrakings. Figure 2.1 and Figure 2.2 show the user interface of CSRankings.

2.2.1 Methodology

CSRankings ranks institutions based on two indexes: *'adjusted counts'* and *'average counts'*. From the viewpoint of faculty, the *'adjusted counts'* is computed as the sum of *'adjusted credit'* which is divided uniformly across all co-authors in each publication. Hence, simply adding authors to an article cannot increase rankings. In more detail, once an article is published at a selected-top-tier conference, a faculty member obtains $\frac{1}{N}$ credit for that paper, where N is the number of authors regardless of their presence in the database of CSRankings. Thus, the achieved credit is stable. If all authors of the published paper are in the database, the maximum point of the paper is recorded as 1 credit. However, researchers pointed out that the technique of adjusting count can lead to a situation discouraging undergraduate students involved in research works. Furthermore, this

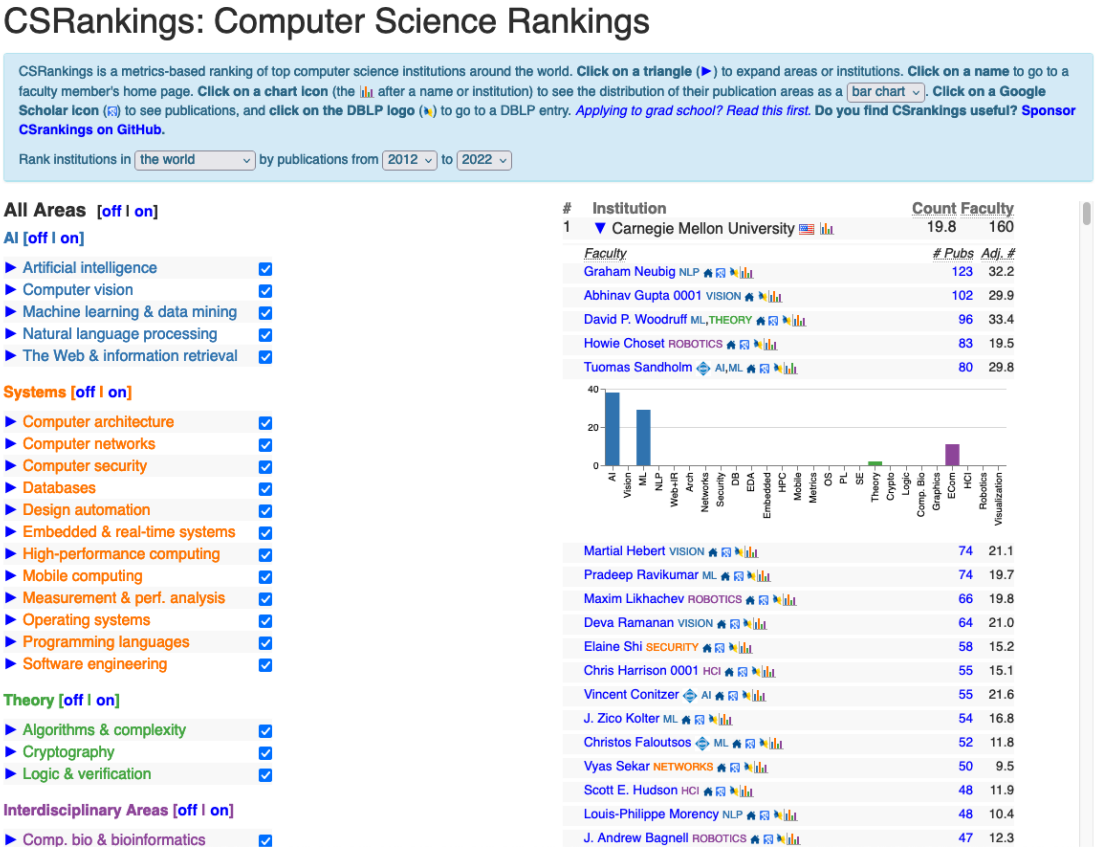


Figure 2.2. *Institution ranking within all areas in the world — showing adjusted count and areas of each author.*

approach could create a disincentive against collaborating across institutions [28]. Many contributors proposed an approach that only divides credits over authors who are in the database. According to [13], however, such approach could have several disadvantages:

- It would be challenging to manually calculate the authorships.
- The credits of authors would be unstable, changing over time. When someone is removed from the database, the credit of other authors would increase.
- It would be an encouragement for senior researchers to collaborate with junior students who are not offered tenure.
- It would encourage to collaborate with industrial researchers, who are not presented in the database, instead of academic people, since authors do not have to share their credits.
- It would create a disincentive for senior faculty to see their junior co-authors get faculty appointments as they have to share credits when their co-authors present in the database.

From the viewpoint of institution, *'average counts'* is an output of the geometric mean of *'adjusted counts'* per area:

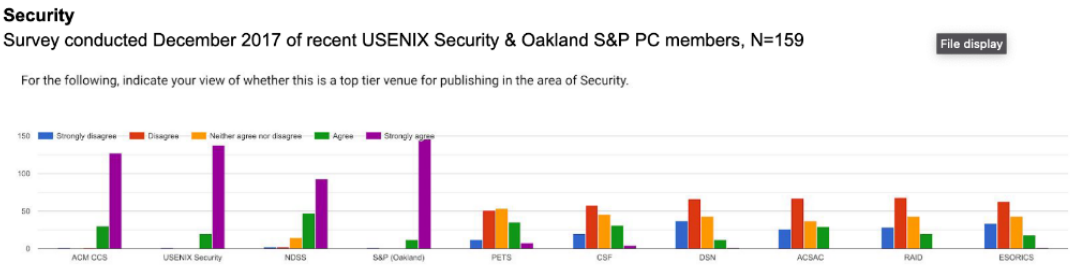


Figure 2.3. *The survey result indicating top tier venue in Security field.*

$$averageCount = \sqrt[N]{\prod_{i=1}^{N}(adjustedCounts_i + 1)},$$

where N is an amount of areas selected. By using geometric mean, the only correct method of averaging normalized results, the publication rates and sizes of areas are normalized [22]. Moreover, only articles at least 6-pages long are taken into account. The rationale behind this constraint is stated by the maintainer of CSRankings, Emery Berger: "Conference proceedings often include things like keynotes, posters, and demos, which are all shorter than normal papers. These are generally captured by this page limit." [27].

Since only papers accepted at top conferences are recorded in the CSRankings system, the policy of selecting those prestigious conferences is unavoidably controversial. The list of selected conferences in each area is constituted based on the results of surveys recording assessments of faculty across a variety of universities. Each participant of the survey is requested to grade pre-selected conferences on the scale of 1 to 5, *Strongly disagree to Strongly agree* respectively (see Figure 2.3). There is a range of opposite viewpoints around this venue-selecting approach. In the scope of this thesis, I do not conclude which method is better. As CSRankings is an open source project, everyone is encouraged to contribute their viewpoints. For instance, through an issue reporting system, researchers requested inserting a top-tier venue [26], or suggested removing a conference reconsidering its top-tier status [29].

2.3 dblp

dblp, a web server providing bibliographic metadata and linking to electronic versions of computer science articles, is a bibliometric data provider for CSRankings. The dataset of dblp is freely accessible and high-quality; thus, it has become a widely-used tool for measuring the academic performance of researchers or institutions [51]. Initially, it was a

digital library operated by the *database systems and logic programming (dblp)* research group at University of Trier from 1993 to the end of 2010 [17]. From 2010 to 2018, dblp was a joint service of the University of Trier and Schloss Dagstuhl — Leibniz Center for Informatics [16]. Since November 2018, Schloss Dagstuhl has been the main operator collaborating with the University of Trier. In addition to the two aforementioned contributors, dblp has received grants by several donors: Deutsche Forschungsgemeinschaft (German Research Foundation), Leibniz Gemeinschaft (Leibniz Association), Klaus Tschira Stiftung gGmbH (Klaus Tschira Foundation), Allen Institute for Artificial Intelligence (AI2), Microsoft Research, and the Very Large Data Base Endowment [16]. At some times, the name “Digital bibliography & Library Project” was seen as a backronym of dblp, but this name is not in use now [17]. The correct name of dblp is “dblp”, or “dblp computer science bibliography” [17].

With respect to the completeness of dblp, there are at least two scientific studies regarding it [51, 54]. However, their analyses were conducted more than 10 years ago, not reflecting the current degree of coverage of dblp database. A study shows that approximately 24% of all publications in the computer science field are indexed by dblp [51]. In terms of sub-field coverage, dblp was supposed to store publications only from the area of database systems and logic programming at the beginning [17]. However, it has extended continuously and included all areas of computer science [17]. This scope extension started after the year 2000 with the support of large number of common authors [54]. As publications at conferences have a main impact on bibliometric studies in the computer science field, the coverage of conference is a vital factor of any bibliographic data services focusing on the computer science field [39]. In case of dblp, the conference coverage of all sub-fields was around 65% in average at the end of 2005 [39]. However, dblp is still incomplete, and true comprehensiveness can never be reached [14].

At the time of writing, dblp contains more than 6,500,000 bibliographic entries. From 2019 to 2022, dblp inserted 500,000 new entries per year in average (see Figure 2.4). Anyone can freely access to the raw dataset of dblp via https://dblp.org/xml/dblp.xml.gz.

2.3.1 Structure of records

Being started as a simple server for testing web technology, dblp was a minimal collection of Table of Contents (TOCs) of significant publications from the fields of database systems and logic programming [42]. These TOCs were inserted and formatted in static HTML, with a few introduction pages manually added [42]. The system design of dblp was lean, emphasizing the simplicity and manageability. It was operated with a minimal amount of custom scripts and without the support of a database management system.

Since publication data was stored as a collection of HTML-based TOCs, the data structure was homogenous [42]. Each person in the dblp database has their own author page.

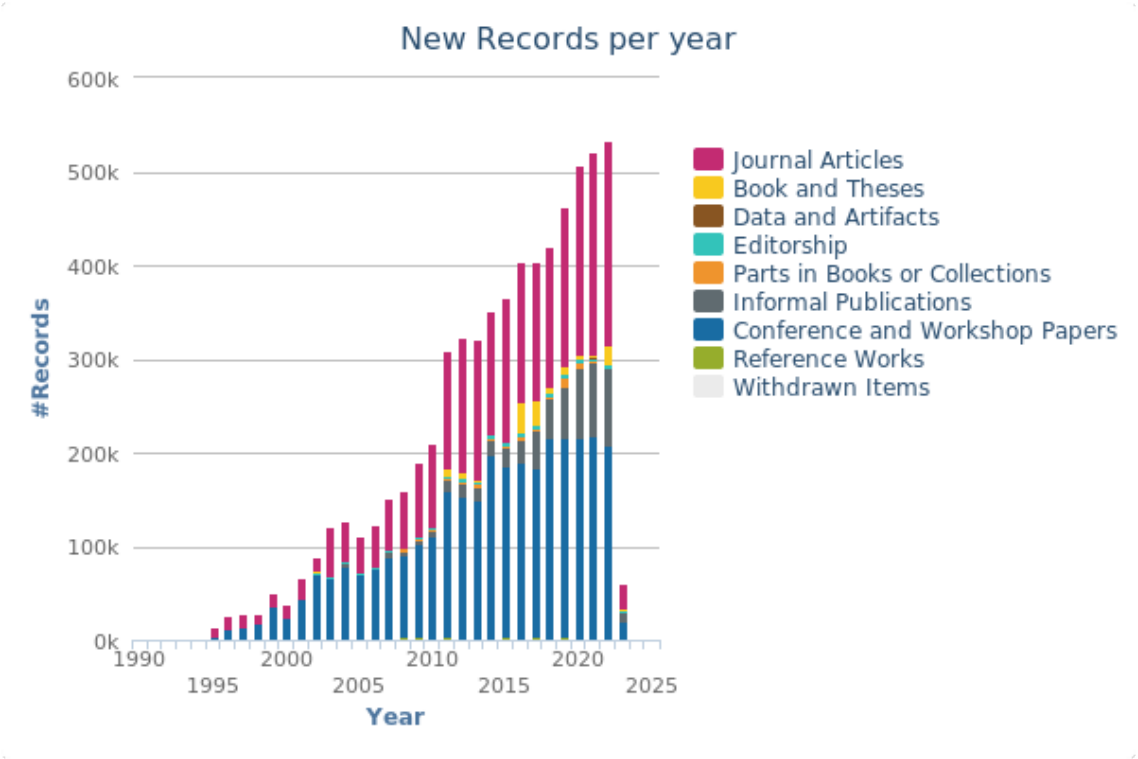


Figure 2.4. *New dblp records per year [15].*

Author pages listing all academic papers (co)authored by a person were generated in two steps. In the first stage, the HTML parser based on the discontinued xmosaic browser was compiled with a shell script, in which identifying information was hard-coded. The output of the compilation was a large line oriented file called “TOC_OUT”. In the next stage, the program “mkauthors” read “TOC_OUT” into a main memory data structure and produced HTML files of author pages, an index of all author pages, and the “AUTHORS” file storing all author names [42]. According to [41], the typical HTML page looked like this:

```
<h2>Keynote Addresses</h2>
<ul>
<li><cite key="conf/vldb/Jhingran06" style=ee>
<li><cite key="conf/vldb/Sikka06" style=ee>
</ul>
<h2>Ten-Year Best Paper Award Talk Session</h2>
<ul>
<li><cite key="conf/vldb/HalevyRO06" style=ee>
</ul>
<h2>Research Sessions</h2>
<h3>Continuous Query Processing</h3>
<ul>
<li><cite key="conf/vldb/LiCTACH06" style=ee>
```

```
<footer>
```

Due to the growth of the amount of bibliographic information, HTML-based tables of contents became an unreliable design. In order to manage the increasing amount of citation linking, annotated bibliographies, etc., publications assigned with unique IDs should be stored in the more bibliographic records [42]. The appearance of the new Extensible Markup Language (XML) format at that time, in 1994, was the solution for the need of changing the structure of files storing bibliographic records. As a result, the dblp XML record was able to easily adapt to the new XML framework with only minor syntactic modifications. A dblp XML record looked like this:

```
<article key="journals/sigmobile/AthalyeBKMZ20" mdate="2020-11-04">
<author pid="203/8790">Anish Athalye</author>
<author pid="14/8279">Adam Belay</author>
<author pid="k/MFransKaashoek">M. Frans Kaashoek</author>
<author pid="82/11191">Robert Tappan Morris</author>
<author pid="99/5780">Nickolai Zeldovich</author>
<title>Notary: A Device for Secure Transaction Approval.</title>
<pages>34-38</pages>
<year>2020</year>
<volume>24</volume>
<journal>GetMobile Mob. Comput. Commun.</journal>
<number>2</number>
<ee>https://doi.org/10.1145/3427384.3427395</ee>
<url>
db/journals/sigmobile/sigmobile24.html#AthalyeBKMZ20
</url>
</article>
```

In addition to a publication record as shown above, a person record, storing supplemental information of author, is also generated as the following format:

```
<person key="homepages/99/5780" mdate="2016-11-11">
<author pid="99/5780">Nickolai Zeldovich</author>
<note type="affiliation">
Massachusetts Institute of Technology, Cambridge, USA
</note>
<url>
https://scholar.google.com/citations?user=DJ6--hMAAAAJ
</url>
</person>
```

2.3.2 Author name ambiguity

The author name ambiguity is a common and unavoidable problem in digital libraries [66, 37, 23]. Names may not be adequate to differentiate one person from another in many circumstances due to two problems: homonym and synonym [61].

Homonyms are a collection of words that have similar spelling and pronunciation but diverse meanings [11]. In the context of dblp, when an author's name is queried, it is not guaranteed that only one matching result is returned. It is common in dblp that two or more authors share the same name. For example, Thomas Olsson, a common Swedish name, matches two different person records in dblp. In order to make homonyms distinguishable, a string of four digits is appended to the names:

```
<author>Thomas Olsson 0001</author>
<author>Thomas Olsson 0002</author>
```

Another issue while dealing with personal names is the synonym problem, different words have similar or identical meanings. In case of dblp, an author can have many different names. The variant of name is classified by the permutations of name parts **(Li Chen ∼ Chen Li)**, the diacritical mark **(René ∼ Rene)**, or the expanded initial **(M. Ley ∼ Michael Ley)** [41]. To represent multiple alias names, the name variants are inserted in addition to the primary name, the first <author> element, in the person record:

```
<person key="homepages/d/MargaretHDunham" mdate="2020-07-09">
<author pid="d/MargaretHDunham">Margaret H. Dunham</author>
<author pid="d/MargaretHDunham">Margaret H. Eich</author>
<note type="affiliation">Southern Methodist University, Dallas,
Texas, USA</note>
<url>http://engr.smu.edu/~mhd/</url>
<url>https://dl.acm.org/profile/81409595451</url>
</person>
```

To identify the name issues described above, dblp uses simple heuristics on the collaboration indexing [41, 23, 66]. dblp colors the author and co-authors with the same color. If there is no direct collaboration between two authors, or indirect collaboration through a common co-author, they are assigned different colors. If the co-author list is monochrome, the main name entry represents a single author [41, 66]. Otherwise, if a name entry includes multiple authors with the same color, it could be a candidate for homonym, and further investigation or splitting may be necessary. The requests for splitting are typically initiated by authors who notice their academic papers mixed with publications of other authors. Besides, in some cases, maintainers of dblp can also identify the need to split a name entry. However, homonyms still remain undetected in dblp database. To address issues with synonyms, dblp uses a software to compare the names of two authors in many

variants. If two authors have never collaborated directly but have common publications, their names are marked as synonyms [23].

The author name ambiguity may not be a significant problem for users who query authors' name for occasional purposes. However, it is a crucial issue for those who use bibliographic information platforms for accumulating academic knowledge to conduct a research and acquiring practical insights to make decisions about recruiting or funding [2, 34].

Despite the fact that many ambiguous names are still not properly distinguished, dblp shows good performance in disambiguating author names [37].

2.3.3 Person IDs

So far, dblp has been widely used as a source of reference by conference servers, preprint servers, publishers, and universities [41]. In order to be an optimal data provider for these clients, the URLs of person records, which cannot contain variable elements such as person name, should be stable. Hence, creating a unique ID for each person in dblp database is necessary. Person ID (PID) without the "homepages/" prefix, also an ID of each person record, is used to map to an author page via the following URL:

```
https://dblp.org/pid/<PID>.xml
```

where *<PID>* is the person record ID of the target author.

2.4 Version control with Git

CSRankings is an open source project which is hosted on GitHub; therefore, anyone can contribute to it through the pull-request protocol. In this section, essential terms and definitions related to version control models will be introduced. Firstly, Version Control is a system that logs the changes of files over time so that these files can be reverted to a specific state later. In addition to the changes over time, a Version Control System (VCS) also records who modified the file and when, who introduced the issue and when, and more [10].

According to a survey of 820 developers [8], in 2014, 65% utilized Distributed Version Control System (DVCS) and 35% used Centralized Version Control System (CVCS). In CVCS, there is a single server that hosts all versioned files within a project, and users work on files from that central codebase. There are some products implementing that client-server approach such as CVS, Subversion, and Perforce. On the other hand, DVCS is a peer-to-peer approach in which each user stores the full history of a project on their local devices. According to [8], this distributed system offers many advantages over CVCS:

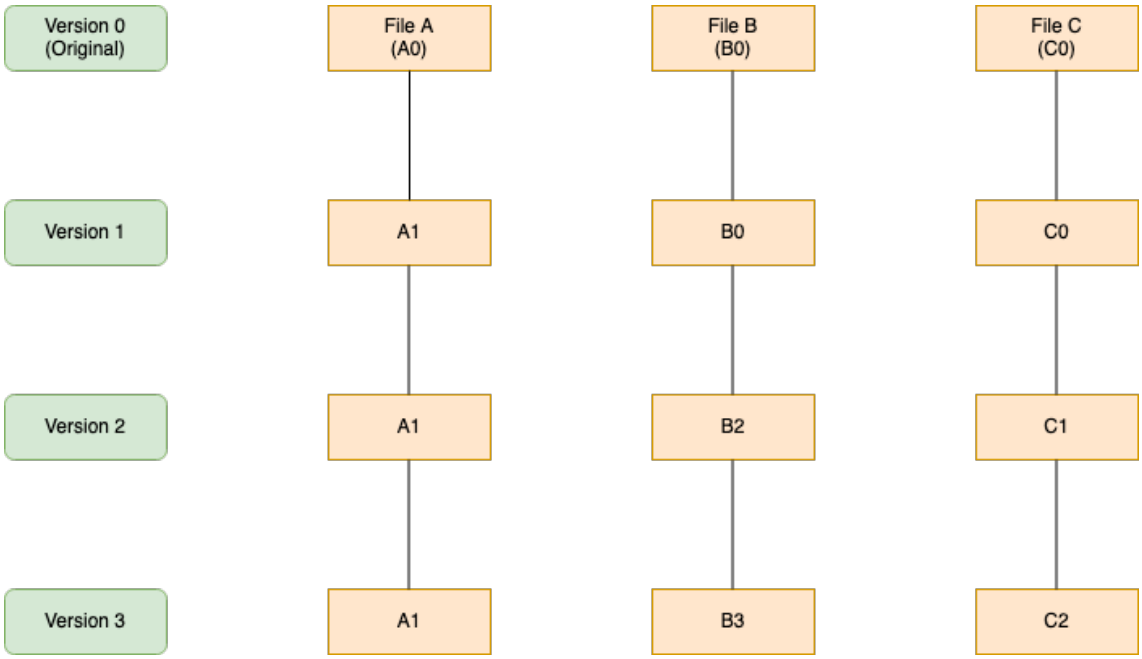


Figure 2.5. *Git stores a series of snapshots.*

- Developers can work independently on their local copies of repositories, allowing them to work offline while maintaining the full project history.
- Developers can effortlessly create and merge branches at a low cost.
- Developers can commit only the lines of a file that have changed, as opposed to having to commit the entire file as in CVCS.

There are many widely-used DVCSs such as Git, Darcs, Mercurial, and Bazaar. In this thesis, I chose Git to focus on, as component files of CSRankings are versioned with Git.

2.4.1 Git

According to [50], the usage of Git was reported by 93.87% of the participants surveyed in 2022. This indicates beyond doubt that Git is the most widely used DVCS. By using Git, contributors fully mirror the shared repository rather than capturing only the latest changes of files [48, 10]. It does not keep track of version controlled files as a series of changes, but instead as a sequence of snapshots. This means that Git takes snapshots of all files under a version controlled project once a user saves the state of the project. Figure 2.5 visualizes the way Git manages versions of files.

Git has a significant amount of commands which can be studied further through its document. In the scope of this thesis, the following commands were mainly used:

- **git fetch**: download the latest version of a remote repository for examining without applying the latest changes to the local repository. A user can fetch from any configured, or even non-configured, remote. Explicitly pointing to which remote to

be fetched is optional as Git uses some heuristics to figure out what is the default remote from the context in which **'git fetch'** was called.

- **git add**: begin tracking the latest local changes (newly added files, last modified files) which are not currently captured by Git.
- **git commit**: with this command, a commit is created, recording a snapshot including all changes tracked via **'git add'**. A commit records a snapshot of a version controlled project which can be reverted to a previous stage or compared to later. The snapshot is still on the local machine that creates the commit. To sync the local snapshot to a remote repository, **'git push'** needs to be executed. The commit is labeled by its Secure Hash Algorithm 1 (SHA1) checksum which is calculated from the state of a repository, including the hash of all files in the repository, the hash of the previous commit, date and time, etc.
- **git pull**: copy remote changes to the local repository, detecting that there are no conflicting modifications after local changes are applied.
- **git push**: upload local commits to a remote repository, from which they can further be distributed to the local repositories of other developers.

With regard to the stability of a project, branching is utilized to avoid accidentally raising errors in the working version of the project. The idea of branching is to separate the main line of development from what changes are going to be generated. When **'git commit'** is executed, Git saves a commit object containing the reference to previous commits (parent commits) [10]. Thus, a branch is simply a portable reference to one of existing commits [10]. In terms of tracking which local branch a contributor is working on, Git maintains a *HEAD* pointer. In addition, "main", or "master", is the default branch initialized when a new repository is created. These default branches are functionally equivalent to other branches. In other words, all branches are technically identical within a project. For illustrative purposes, commits are labeled as C_i, where *i* is $1, \ldots, 4$, in Figure 2.6.

There are several branching workflows which are widely used in practice. However, topic branch is the reliable and effective workflow used in this work. A topic branch is a short-lived branch that is created and used for a single particular function or related features [10]. For example, a new log-in function is developed in a web application. A topic branch called "log-in_function" is created and diverges from the "main" branch, the official working version of the application (see Figure 2.7). On the new branch, developers can make changes without damaging the "main" line.

Once the newly developed feature, log-in function in this example, is ready to be integrated into the web application, a merge command is executed. As a result, a merge commit, which has at least two predecessors, is created on top of the stack of commits (see Figure 2.8).

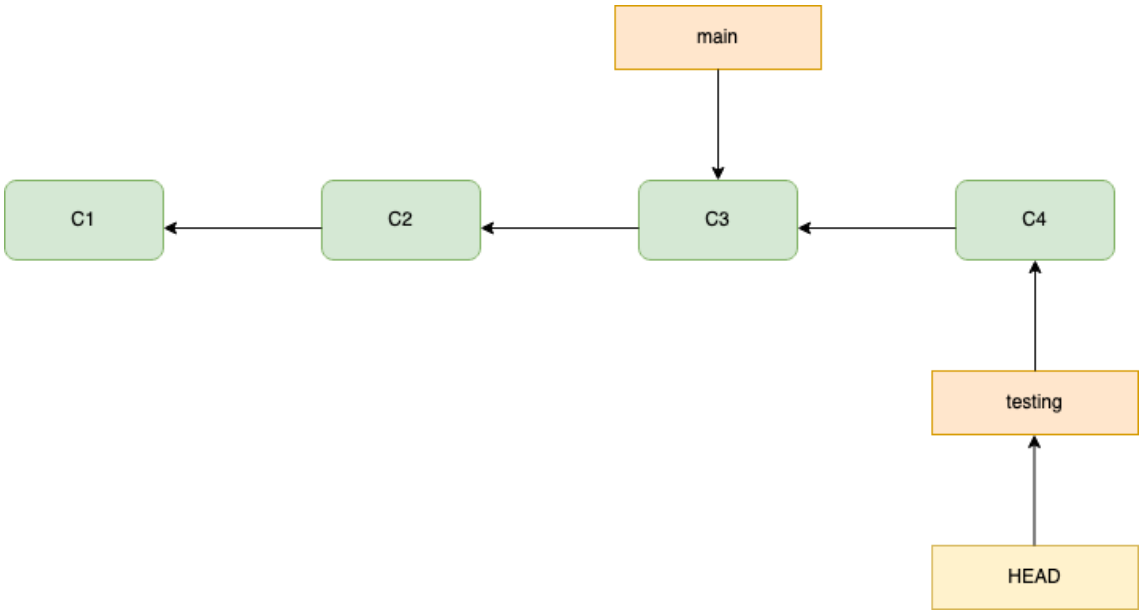


Figure 2.6. *Two branches pointing to different commits.*

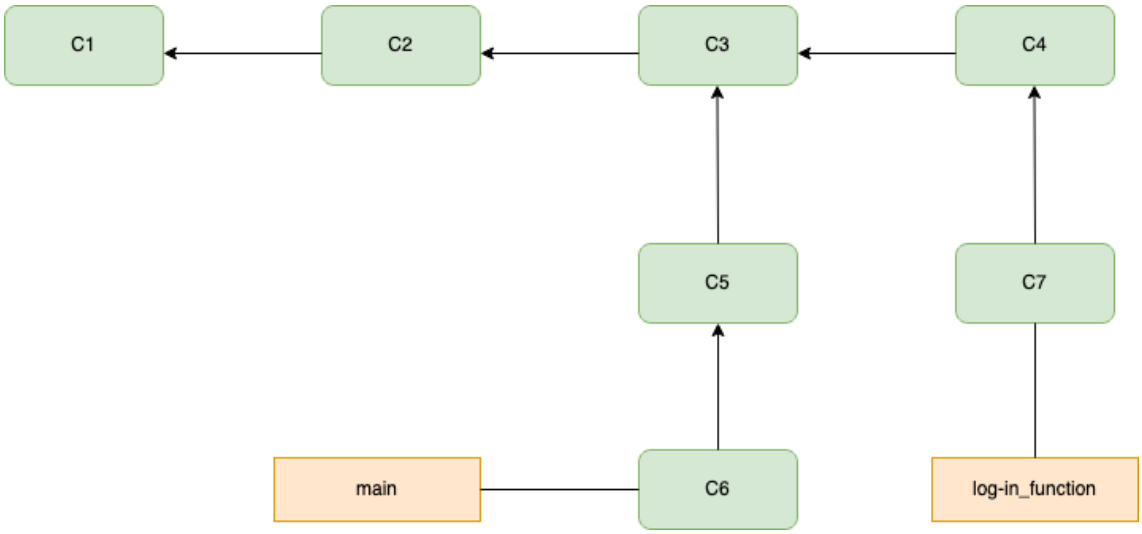


Figure 2.7. *A topic branch besides the main (default) branch.*

2.4.2 GitHub

GitHub, a web-based hosting service for version control using Git, is utilized by many open source projects for Git hosting, access control, bug tracking, and collaborative coding [10]. Beside hosting Git repositories, GitHub provides additional features: collaborative coding, automation and Continuous Integration (CI)/Continuous Delivery (CD), security scanning, project management, and team administration. Founded in 2008, GitHub has been a subsidiary of Microsoft since 2018.

Git and GitHub are sometimes interchangeably used, but they are different in a number of aspects. Git is a DVCS which is installed on local devices. Developers use Git to keep history of changes on their local storage. In contrast, GitHub is a cloud-based service

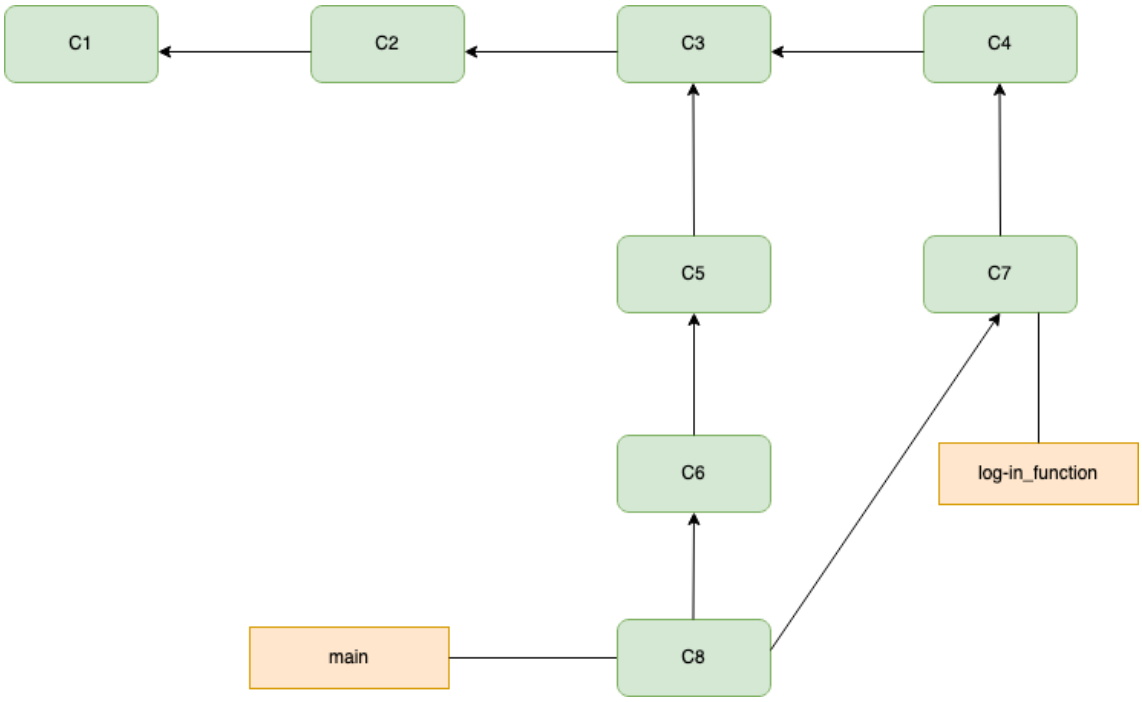


Figure 2.8. *History after merging "log-in_function".*

in which developers can share their code to others, and other developers can contribute to others' code as well. GitHub can be seen as a centralized location for hosting copies of local Git repositories. In terms of user command, Git focuses on command-line tools, including **git add**, **git commit**, **git push**, etc. On the other hand, GitHub serves graphical interfaces where tasks are performed. In addition to functions of version control, GitHub offers various administration tools, collaboration features, integration tools, and a wide range of external plugins. GitHub is designed to work with Git, meaning that GitHub cannot operate properly without Git.

In this work, since the main task is to contribute to the CSRankings GitHub repository, I will discuss the *collaborative coding* feature in more detail. A Pull Request (PR) is the mechanism through which contributors can submit changes for inclusion in a repository owned by another user or organization [25]. Each PR has its own discussion forum where contributors and owner can comment on it, illustrated in Figure 5.1. As a result, the repository owner can accept or reject a PR.

According to [10], there is a collaboration workflow which GitHub users follow while contributing to an open source project:

1. Fork the project. The term "fork" indicates the act of copying a project, which a contributor do not have push access, to their namespace for later contributing [10]. After pushing changes to the personal fork of the project, the contributor can open a PR to inform others about the changes. Once it is opened, the contributor can discuss and review the potential changes with collaborators and add supplemental

commits if necessary before these changes are merged into the original repository [24].

2. Create a topic/function branch from the **"main"** branch.
3. Edit files and create commits.
4. Push this branch to the personal GitHub repository of the contributor.
5. Open a PR on GitHub against the original 'upstream' project.
6. Discuss, and continue committing if it is necessary.
7. The owner of the original 'upstream' project merges or closes the PR.
8. Sync the updated **"main"** branch back to the fork of the contributor.

The next subsection covers other DVCS hosting services which are similar to GitHub.

2.4.3 GitHub and its alternatives

Besides GitHub, there are many other Git hosting services, including, for example, GitLab, Bitbucket, and SourceForge. Among them, GitLab has gained significant attention. GitLab, an open source web-based service for Git repositories, provides collaborative and end-to-end software development platform with built-in version control. Developed by GitLab Inc, GitLab was launched in 2011. In addition to source code management, GitLab offers other key features: team planning, continuous integration, package registry, code review workflow, fuzz testing, continuous delivery, error tracking, Kubernetes management, vulnerability management, etc.

Both GitHub and GitLab are based on Git, making it simple for developers to migrate their code seamlessly across the two platforms. In addition, they share many common features. GitHub and GitLab both provide issue tracking tools that allow for status modifications and the assignment of owners to each issue. On both platforms, developers can add a description or comment to issues or Merge/Pull Requests. Both of them maintain a separate system for documentation known as Wiki and is built into each project as a separate Git repository. On the other hand, both platforms offer similar functions but with different terminologies (see Table 2.1). There are still many similarities of both services from different aspects, but they are beyond the scope of this thesis.

GitLab and GitHub have many key differences which fulfill requirements of different projects. Hence, developers need to consider the scope, structure, and resource of their products before selecting the appropriate platform.

GitHub	GitLab	Meaning
Pull Request	Merge Request	Request to integrate a branch to another branch.
Gist	Snippet	Instantly shared code.
Repository	Project	Container storing all project's files and each file's revision history, and project-specific settings.
Organization	Group	Shared accounts to manage many related-projects at once.

Table 2.1. *Different terminologies have similar functions in GitHub and GitLab.*

3. RELATED WORK

In addition to CSRankings, there are other university rankings that are based solely on bibliometric data, including, most notably, NTU World University Ranking, University Ranking by Academic Performance (URPA), CWTS Leiden Ranking, and GRAS [49]. However, these rankings are based on other indicators — number of citations, h-index, etc. — as well, not only on the number of published articles.

NTU World University Ranking assesses institutes based on three criteria, including *Research Productivity*, *Research Impact*, and *Research Excellence* [62]. These criteria contribute different weights, ranging from 10% to 15%, and each has a unique set of indicators [62]. A full detail of its methodology can be found at http://nturanking.csti.tw/methodoloyg/indicators.

URPA ranks higher education institutions based on six indicators, including *Article*, *Citation*, *Total Document*, *Article Impact Total*, *Citation Impact Total*, and *International Collaboration* [63]. These indicators constitute a set of different weights, ranging from 10% to 21%, on the overall ranking [63]. A full detail of its methodology can be found at https://urapcenter.org/Methodology.

CWTS Leiden Ranking assesses universities based on four group of indicators , including *Scientific impact*, *Collaboration*, *Open access*, and *Gender* [40]. Only core publications are counted while calculating indicators [40]. A full detail of its methodology can be found at https://www.leidenranking.com/information/indicators.

Among the above rankings, I selected GRAS as the ranking which is studied in this thesis. The reason behind this selection is that GRAS is the most influential university ranking which is non-reputation-based, and its methodology is transparent and stable [59, 18].

3.1 ShanghaiRanking's Global Ranking of Academic Subjects (GRAS)

GRAS, conducted by ShanghaiRanking Consultancy, was published for the first time in 2009. In the first version, it ranked institutions in 5 subjects, including Mathematics, Physics, Chemistry, Computer Science, and Economics/Business. Over the years, the range of subjects has extended to 54 subjects in the latest publication, GRAS 2022.

However, in order to build a correlation to CSRankings which is a ranking focusing on computer science field, only the computer science subject is described thoroughly in this thesis. Similar to CSRankings, GRAS utilizes third-party data providers, WoS and InCites, as a bibliometric database. With regard to the computer science area, each database has its unique advantages. For example, dblp indexes the significant number of unique publications, while WoS provides high-quality indexing and bibliographic records in terms of accuracy, consistency, and relevance [9].

3.1.1 Methodology

GRAS 2022 only examines universities that have a certain number of publications during the period of 2016–2020. The publication threshold varies in subject. In the case of computer science, only universities that published at least 150 articles in the period of 2016–2020 are eligible to be examined. According to [53], GRAS measures the academic performance of universities through the following objective indicators:

- **Research output (Q1)**: the number of articles published by an institution in the particular subject in journals with Q1 Journal Impact Factor Quartile during a period.
- **Research influence (CNCI)**: the ratio of citations of articles, published by an institution in the respective subject during a period, to the average citations of articles in the same field, the same year and same type of journal publication. The value of 1 indicates that the citation performance is at world-average level. CNCI less than 1 represents the below-average level, while CNCI greater than 1 demonstrates above-average citation performance.
- **International influence (IC)**: the percentage of internationally co-authored articles conducted by an institution in the particular subject during a period.
- **Research quality (Top)**: the number of academic papers published in top journals or top conferences in the respective subject for an institution during a period.
- **International academic awards (Award)**: the number of researchers of an institution achieving a prestigious award in the particular subject since 1981.

GRAS considers a wider set of indicators in calculating the score of each university. In case of CSRankings, only the number of academic papers published at top venues is taken into account. This indicator can be considered as the counterpart of GRAS's *Research quality (Top)* indicator.

GRAS allocates different weights to the indicators for different subjects. The weights in Computer Science field are listed in Table 3.1. The final scores of institutions are calculated as the sum of the score of each indicator in a respective subject:

Subject	Q1	CNCI	IC	Top	Award
Computer Science	100	100	20	100	100

Table 3.1. *Indicator weights of Computer Science subject*

$$\sum_{i \in \mathbb{L}} \sqrt{P_i} \cdot W_i,$$

where W_i is the weight of each indicator; P_i is the percentage of the top scorer for each indicator, and $\mathbb{L}$ is the set of indicators $\{Q1, CNCI, IC, Top, Award\}$.

The webpage of GRAS presents the total scores and component scores for each indicator. They illustrate the academic performance of the department in general. On the other hand, CSRankings shows total score and individual score (score of each researcher). Hence, the users can have information about the academic performance of an individual professor and the department as well.

In terms of conducting surveys, GRAS is different from CSRankings in a number of aspects. Top journals, top conferences, and top academic awards are selected as the result of the Academic Excellence Survey (AES). The participants are professors from the top 100 universities, and many of them are chairs or heads of departments. In contrast to the survey conducted by CSRankings, ShanghaiRanking AES publishes the names and affiliated institutions of all respondents, producing the transparent result of the survey. The survey consists of two optional questions. The first question asks the respondents to list the top journals and conferences in their primary subjects. In the second question, participants are required to propose the most influential and internationally prestigious academic awards in their primary subjects. According to [52], the selection of a journal, a conference, and an award is based on the answers of participants along with the following criteria:

- **Top journal**: if it acquires more than one vote in one subject, and it has minimum 50% votes on a particular subject, or it was a top journal in the previous year.
- **Top award**: if it obtains at least one vote in one subject, and it has 50% or more votes on a particular subject, or it was a top award in the previous year.
- **Top conference**: if it is nominated by at least 10 participants, or it was a top conference in last year.

Considering the requirement of selecting top journals/awards/conferences raises a question: “Is it possible that a journal/award/conference will be considered as a top venue, automatically and indefinitely for the following years, after it has been selected once?”

For example, a conference was selected in 2017 as it was voted by 10 survey participants. In 2018, only 5 survey respondents selected it, but it was still considered a top conference as it was selected in 2017. The same situation could happen in 2019 or later. The AES was first published in 2017; therefore, this problem could be investigated in more detail if the methodology of AES 2017 can be examined. However, at the time of writing, methodologies of AES from 2017 to 2020 are not publicly accessible; thus, it is not possible to answer this question in this thesis. Nonetheless, it is worth highlighting the potential issue here to further remark the intricacies of defining objective and unbiased methodologies for these rankings.

The full list of journals — conferences, awards, and participants — is published on this page: https://www.shanghairanking.com/activities/aes/method/2022

3.1.2 Ideology

GRAS aims to provide a reliable university ranking in a wide range of subjects across Natural Sciences, Engineering, Life Sciences, Medical Sciences, and Social Sciences. The target audience of GRAS are both undergraduate and graduate students who want to pursue a Bachelor's or Master's program. In a particular subject field, GRAS ranks the department based on its academic performance in general.

CSRankings differs from GRAS in a number of aspects. CSRankings ranks higher education institutions in a wide range of areas across Computer Science. The target audience of CSRankings is post-graduate students who want to pursue a doctoral program. The goal of CSRankings is to help prospective students to have more information about the research activities of professors, or research groups, who they are interested in. Thus, it is assumed that the users of CSRankings have a clear understanding of their research area and are seeking to identify professors whose research direction aligns with their own.

In conclusion, both rankings aim to provide a reliable ranking system through which student and university can benefit each other. Students can find their best universities by researching the ranking, and universities can treat these performance-based rankings as a motivation to improve themselves and attract more talents.

4. IMPLEMENTATION

In order to integrate the CS Unit of TAU into the CSrankings system, many requirements must be followed. All detailed instructions are publicly accessible through the Git repository of CSRankings. According to [12], there are four important guidelines that should be noticed:

GL1 Do not modify any files except *csrankings-[a-z].csv* or (if needed) *country-info.csv*.

GL2 Make sure the Google Scholar IDs are just the alphanumeric identifier (not a URL or with &hl=en).

GL3 Check to make sure the home page is correct.

GL4 Check to make sure the name corresponds to the dblp entry (look it up at http://dblp.org).

GL5 Insert new faculty in alphabetical order (not at the end) in the appropriate *csrankings-[a-z].csv* files. Do not modify *csrankings.csv*, which is auto-generated.

In this work, I mainly analyzed these Comma-separated values (CSV) files: *csrankings.csv*, *master.csv*, *csrankings-[a-z].csv*, and *country-info.csv*. Among these files, *csrankings-[a-z].csv* is a set of alphabetical-sorted files starting from *csrankings-a.csv* to *csrankings-z.csv*. In case of *master.csv*, we created it to keep track of the information of personnel in the CS Unit within the Faculty of Information Technology and Communication Science at TAU. Moreover, *master.csv* was created in a separate repository meant to host the work related to this project, not being part of CSRankings repository. On the other hand, other CSV files are managed by the maintainers of CSRankings. However, everyone can contribute to them through a PR which is also the final step in the integration phase. The links of aforementioned files are listed in Table 4.1.

In this chapter, I will discuss the contributing guidelines, integration steps, and their motivation. Each integration step that is introduced below addresses the corresponding guideline.

4.1 Data collecting and processing

According to GL1, a new entry needs to be inserted into *country-info.csv* when the home institution is not in the USA. Since TAU locates in Finland, I needed to add its entry to

File	Link
master.csv	https://gitlab.com/nisec/bibliometrics/-/blob/main/master.csv
csrankings.csv	https://github.com/emeryberger/CSrankings/blob/gh-pages/csrankings.csv
country-info.csv	https://github.com/emeryberger/CSrankings/blob/gh-pages/country-info.csv
csrankings[a-z].csv	https://github.com/emeryberger/CSrankings

Table 4.1. *Links of aforementioned CSV files.*

country-info.csv. Table 4.2 shows the format of entries in *country-info.csv*.

institution	region	countryabbrv
AUEB	europe	gr
Aalborg University	europe	dk
Aalto University	europe	fi
Aarhus University	europe	dk
Aberystwyth University	europe	uk

Table 4.2. *The first 5 records of the country-info.csv.*

Following the above format, I inserted the TAU entry as:

```
Tampere University,europe,fi
```

With regard to faculty inclusion, each entry included in the CSRankings system must contain required fields: *name (dblp name)*, *affiliation*, *homepage*, and *scholarid*. In this work, *affiliation* of all faculty entries are “Tampere University”. In addition, each eligible faculty must be a full-time, tenure-track member who can solely supervise PhD students in the computer science field [12]. To help clarifying this requirement, I added the *job title* field beside required fields in our internal *master.csv* file.

Following GL2 and GL3, I manually searched Google Scholar IDs and homepages of faculty through the Google search engine. By scanning the homepage, I can collect the *job title* of a particular researcher as well. Most researchers at TAU have their homepages in the form as:

```
https://www.tuni.fi/en/<full-name>
```

e.g. https://www.tuni.fi/en/karen-eguiazarian. However, some people have their homepages in the alternative form as:

`https://researchportal.tuni.fi/en/persons/<full-name>`

e.g. https://researchportal.tuni.fi/en/persons/frank-emmert-streib. In order to ensure that these two Uniform Resource Locator (URL) patterns are not mistakenly used, I built a script that checks the status code of the Hypertext Transfer Protocol (HTTP) request. If the returned status code is 200, the recorded homepage is accessible.

In order to double-check Google Scholar IDs, I utilized the Python module known as *scholarly*[1]. This open source module allows users to retrieve author and publication information from Google Scholar. In this work, *scholarly* helped me return a corresponding author name with a given Google Scholar ID. By comparing the returned author name from *scholarly* and the full name, I can detect whether or not the collected Google Scholar ID refers to the intended author.

According to GL4, the *name* field of faculty is the name corresponding to a dblp entry. In some cases, the name of a faculty member is not consistent with their name in the dblp database. For example, Prof. **Davide Taibi** has his corresponding dblp name as **Davide Taibi 0001**. dblp appends numbers to a person name to solve the homonym problem in which many people have the same full name. On the other hand, a person can have more than one corresponding full names, known as the synonym problem. For example, the homepage displays **Karen Eguiazarian** as his full name, but his name in dblp is recorded as **Karen O. Egiazarian**. To address these issues, I maintained two separate fields of name, one for the "normal" name and another for the name in dblp system, to track the consistence between them.

dblp provides an efficient searching function so that I can easily find a dblp author entry by inputting the full name of a particular researcher. In most cases, there is only one matching result returned. On the other hand, in some cases, dblp returns more than one entry. In these cases, I selected the entry corresponding to the Tampere University *affiliation*. If the field of *affiliation* was undefined, I scanned the list of published articles of each returned entry and compared to the list on author's Google Scholar page. I resolved conflicts by considering the author entry that had more articles matching those on the author's Google Scholar page to be correct.

Once a dblp author record was defined, I effortlessly collected the PID of an author through the corresponding URL of the dblp homepage:

`https://dblp.org/pid/e/KOEgiazarian.html`

where the PID is "e/KOEgiazarian". To avoid capturing the wrong PID by mistake, I built a script that validated each PID and its *dblp name*. Firstly, it requested an XML version of a *Person Record* through the following pattern:

[1] https://github.com/scholarly-python-package/scholarly

```
https://dblp.org/pid/e/KOEgiazarian.xml
```

where the dblp server stored:

```
<dblpperson name="Karen O. Egiazarian" pid="e/KOEgiazarian" n="360">
    <person key="homepages/e/KOEgiazarian" mdate="2022-02-15">
        <author pid="e/KOEgiazarian">Karen O. Egiazarian</author>
    <! -- more irrelevant tags are skipped here -->
</dblpperson>
```

Then by comparing the value of the attribute *name* in the *dblpperson* element to the *dblp name* returned from the search function, I can know whether or not the collected *dblp name* and PID point to the same dblp entry.

4.2 Preparing a Pull Request

After all required fields were collected and checked, I inserted faculty entries following the GL5.

Since a certain amount of dblp person names contains non-unicode characters, records cannot be simply sorted by the built-in Python sorting method. Fortunately, CSrankings is an open source project; thus, its sorting method is publicly accessible. By utilizing the script *split-csrankings.py*[2], the sorting step becomes unchallenging. The job of *split-csrankings.py* is to apply the *strip_accents* function to each record in *csrankings.csv* and then sort them in alphabetical order. Therefore, I only needed to append all eligible entries to the end of *csrankings.csv* and execute *split-csrankings.py* script to generate a sorted range of *csrankings-[a-z].csv*. Following the GitHub workflow mentioned in Subsection 2.4.2, I synced-up the local changes to my remote fork of CSRankings repository. At this stage, a pull request is ready to be opened.

Receiving feedback from the maintainer of CSRankings, I created the follow-up commits to address the following issues:

- 7cf758c9: Update homepages which show both titles showing both titles: Academy Research Fellow and Associate Professor (tenure track).
- 8acb67f0: Replace dead homepage.
- 136e8e1a: Remove teaching faculty and visitors.
- 7e449d5d: Remove non-CS faculty.

As a result, the history of commits in my fork of CSRankings diverged, illustrated in Figure 4.1.

[2]split-csrankings.py — https://github.com/emeryberger/CSrankings/blob/gh-pages/util/split-csrankings.py

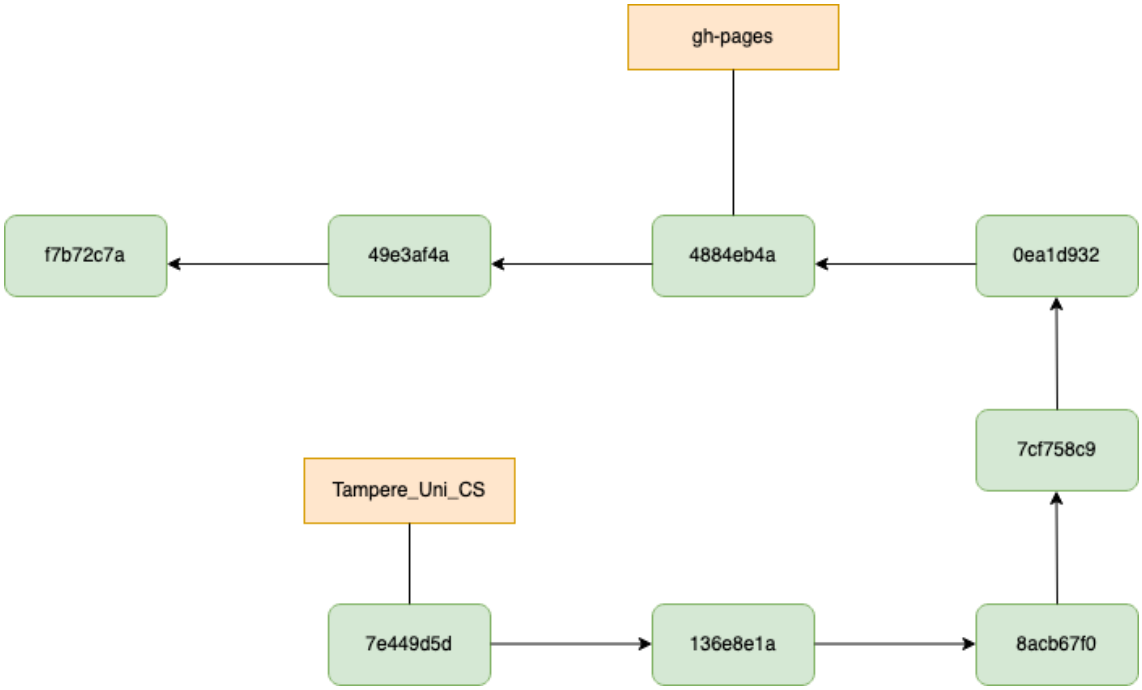


Figure 4.1. *History in my fork of the CSRankings repository.*

5. RESULT

After the problems mentioned in Section 4.2 were solved by the follow-up commits, the pull request was accepted (see Figure 5.1). As a result, Tampere University is now visible in CSRankings (see Figure B.1). By utilizing visualization features of CSRankings (see Figure B.1), I obtained an overview of research areas in which faculty members at the CS Unit of TAU have been involved. Researchers at TAU have published articles in a wide variety of fields, such as artificial intelligence, computer vision, machine learning & data mining, computer security, databases, logic & verification, human-computer interaction, robotics, and visualization. Three most-published areas are presented in Table 5.1.

Furthermore, 16 researchers have academic papers published in top selected conferences during the period 2000–2022, which accounts for 39.02% of the total of 41 eligible people. From 2000 to 2022, the geometric mean count of publications across all areas indexed by Tampere University is 1.4. The selected time frame, between 2000 and 2022, is reasonable as outdated publications that do not reflect precisely the research activity of institution in the current time are excluded. Moreover, 22 years is a sufficient period of time for covering a wide range of academic papers. Depending on the scope of ranking, the normal rank and the percentile rank[1] of TAU fluctuate considerably (see Table 5.2).

The national ranking for Finland is not originally shown by CSRankings since only countries that have at least 5 institutions in its database are eligible for displaying a national ranking. In addition, CSRankings does not use the percentile rank as its benchmarking result. However, the normal rank alone cannot present entirely the academic performance of higher education institutions. For example, in the national ranking for Finland, TAU shows a high-quality performance according to its normal rank (Rank 3), but an opposed result is illustrated by its percentile rank (Rank 16.7). To implement these two supplemen-

[1] The percentile rank of a given score indicates the percentage of scores that are less than that score in its frequency distribution[65].

Area	Computer Vision	HCI	Robotics
Publications	22	18	13

Table 5.1. *The three most-published areas during the period 2000–2022.*

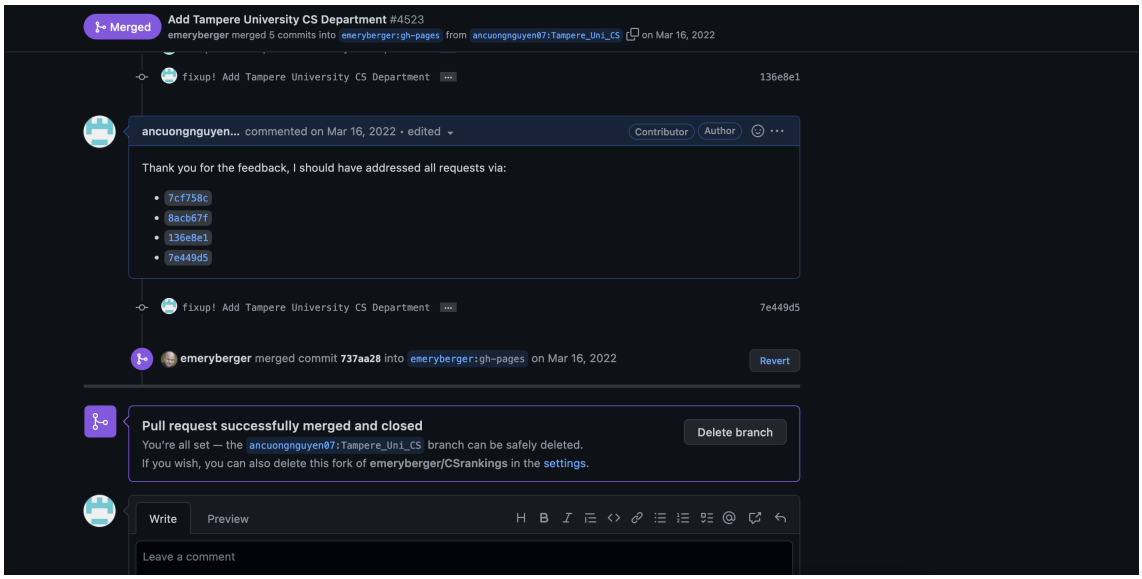


Figure 5.1. *The accepted Pull Request.*

	World Ranking	Europe Ranking	Finland Ranking
Normal rank	337	124	3
Percentile rank	39.1	38.0	16.7

Table 5.2. *Normal ranks and percentile ranks of TAU during the period 2000–2022.*

tal functions, together with my co-supervisor, Nicola Tuveri, we created two userscripts[2] that are executed whenever the CSRankings webpage is visited.

[2]https://gitlab.com/nisec/bibliometrics/-/snippets/2274508

6. CONCLUSION

The goal of this work was to merge the CS Unit of TAU within the Faculty of Information Technology and Communication Science into the database of CSRankings and retrieve statistical results in different scopes of ranking. Following the instructions in the guiding document, I inserted, in alphabetical order, the required information fields corresponding to each faculty member into alphabetical-split CSV files. However, in several cases, the required *dblp name* field does not match the full name of faculty members. To help maintain the correlation between the full name and the dblp name, an additional field was added. Besides, I also applied scripts to validate other required fields. The CS Unit of TAU is now merged on CSRankings after an approved PR.

Regarding to statistical results, CSRankings provides meaningful indexes for evaluating the academic performance of each faculty member and the whole CS Unit in the international level. However, additional statistics were needed to better analyze the academic performance of the CS Unit of TAU at the national level. Thus, we created additional userscripts to generate a national ranking and percentile ranks, yielding more meaningful data for assessment.

In conclusion, this thesis shows that CSRankings, a metric-based ranking, is reliable and promising in the computer science area. By adding the CS Unit of TAU into the system of CSRankings, prospective students, researchers, and potential investors now can have more background information about the faculty that they are interested in. As a result, they will have better tools to make choices about prospective supervisors and collaborators.

REFERENCES

[1] Computing Research Association (CRA). *CRA Statement on US News and World Report Rankings of Computer Science Universities*. 2017. URL: https://cra.org/cra-statement-us-news-world-report-rankings-computer-science-universities/ (visited on 02/10/2023).

[2] Alison Abbott, David Cyranoski, Nicola Jones, Brendan Maher, Quirin Schiermeier, and Richard Van Noorden. “Metrics: Do metrics matter?” In: *Nature* 465.7300 (2010), pp. 860–862. DOI: 10.1038/465860a.

[3] Gleb Beliakov and Simon James. “Citation-based journal ranks: The use of fuzzy measures”. In: *Fuzzy Sets and Systems* 167.1 (2011). Special Issue: Aggregation techniques and applications Selected papers from AGOP’2009 dedicated to Jaume Casasnovas, pp. 101–119. ISSN: 0165-0114. DOI: 10.1016/j.fss.2010.08.011.

[4] Paul Benneworth, Lars Coenen, Jerker Moodysson, and Björn Asheim. “Exploring the Multiple Roles of Lund University in Strengthening Scania’s Regional Innovation System: Towards Institutional Learning?” In: *European Planning Studies* 17.11 (2009), pp. 1645–1664. DOI: 10.1080/09654310903230582.

[5] Emery Berger, Stephen M. Blackburn, Carla Brodley, H. V. Jagadish, Kathryn S. McKinley, Mario A. Nascimento, Minjeong Shin, Kuansan Wang, and Lexing Xie. “GOTO Rankings Considered Helpful”. In: *Commun. ACM* 62.7 (June 2019), pp. 29–30. ISSN: 0001-0782. DOI: 10.1145/3332803.

[6] Nicholas Bowman and Michael Bastedo. “Getting on the Front Page: Organizational Reputation, Status Signals, and the Impact of U.S. News and World Report on Student Decisions”. In: *Research in Higher Education* 50 (Aug. 2009), pp. 415–436. DOI: 10.1007/s11162-009-9129-8.

[7] Nicholas A. Bowman and Michael N. Bastedo. “Anchoring effects in world university rankings: exploring biases in reputation scores”. In: *Higher Education* 61.4 (2011), pp. 431–444. DOI: 10.1007/s10734-010-9339-1.

[8] Caius Brindescu, Mihai Codoban, Sergii Shmarkatiuk, and Danny Dig. “How Do Centralized and Distributed Version Control Systems Impact Software Changes?” In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India: Association for Computing Machinery, 2014, pp. 322–333. ISBN: 978-1-4503-2756-5. DOI: 10.1145/2568225.2568322.

[9] Antonio Cavacini. “What is the Best Database for Computer Science Journal Articles?” In: *Scientometrics* 102.3 (Mar. 2015), pp. 2059–2071. ISSN: 0138-9130. DOI: 10.1007/s11192-014-1506-1.

[10] Scott Chacon and Ben Straub. *Pro Git*. New York City: Apress, 2014.

[11] Robert Christman. *Homonyms: Why English Suffers*. Las Cruces, New Mexico: Barbed Wire Publishing, 2002.

[12] CSRankings. *Contributing to CSrankings*. 2022. URL: https://github.com/emeryberger/CSrankings/blob/gh-pages/CONTRIBUTING.md (visited on 03/14/2023).

[13] CSrankings. *FAQ Computer Science Rankings*. URL: https://csrankings.org/faq.html (visited on 09/20/2022).

[14] DBLP. *How complete is dblp?* 2023. URL: https://dblp.org/faq/23593238.html (visited on 02/06/2023).

[15] DBLP. *Statistics - New Records per year*. 2023. URL: https://dblp.org/statistics/newrecordsperyear.html (visited on 02/06/2023).

[16] dblp. *What institution is behind dblp?* URL: https://dblp.org/faq/1474651.html (visited on 02/03/2022).

[17] dblp. *What is the meaning of the acronym dblp?* URL: https://dblp.org/faq/1474577.html (visited on 02/03/2023).

[18] Catherine Dehon, Alice McCathie, and Vincenzo Verardi. "Uncovering excellence in academic rankings: a closer look at the Shanghai ranking". In: *Scientometrics* 83.2 (July 2009), pp. 515–524. DOI: 10.1007/s11192-009-0076-0.

[19] Emilio Delgado López-Cózar, Enrique Orduña-Malea, and Alberto Martín-Martín. "Google Scholar as a Data Source for Research Assessment". In: *Springer Handbook of Science and Technology Indicators*. Ed. by Wolfgang Glänzel, Henk F. Moed, Ulrich Schmoch, and Mike Thelwall. Cham: Springer International Publishing, 2019, pp. 95–127. ISBN: 978-3-030-02511-3. DOI: 10.1007/978-3-030-02511-3_4.

[20] Wendy Nelson Espeland and Michael Sauder. "Rankings and Reactivity: How Public Measures Recreate Social Worlds". In: *American Journal of Sociology* 113.1 (2007), pp. 1–40. ISSN: 00029602, 15375390. URL: http://www.jstor.org/stable/10.1086/517897 (visited on 02/09/2023).

[21] Eurostat. *R&D expenditure*. 2022. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=R%26D_expenditure&oldid=590306 (visited on 02/15/2023).

[22] Philip J. Fleming and John J. Wallace. "How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results". In: *Commun. ACM* 29.3 (Mar. 1986), pp. 218–221. ISSN: 0001-0782. DOI: 10.1145/5666.5673.

[23] Massimo Franceschet. "Collaboration in computer science: A network science approach". In: *Journal of the American Society for Information Science and Technology* 62.10 (2011), pp. 1992–2012. DOI: 10.1002/asi.21614.

[24] GitHub. *About pull requests*. URL: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests (visited on 02/01/2023).

[25] GitHub. *GitHub Glossary*. URL: https://docs.github.com/en/get-started/quickstart/github-glossary#pull-request (visited on 02/01/2023).

[26] *GitHub issue: add NDSS to computer security*. URL: https://github.com/emeryberger/CSrankings/issues/263 (visited on 09/30/2022).

[27] *GitHub issue: Constraint on papers being at least 6 pages long*. URL: https://github.com/emeryberger/CSrankings/issues/316 (visited on 09/30/2022).

[28] *GitHub issue: Divide Credit for a Paper only among professors in the database*. URL: https://github.com/emeryberger/CSrankings/issues/589 (visited on 09/30/2022).

[29] *GitHub Pull Request: Remove INFOCOM from networks rankings, as it is not considered top-tier*. URL: https://github.com/emeryberger/CSrankings/pull/586 (visited on 09/30/2022).

[30] Maruša Hauptman Komotar. "Global university rankings and their impact on the internationalisation of higher education". In: *European Journal of Education* 54.2 (2019), pp. 299–310. DOI: 10.1111/ejed.12332.

[31] Ellen Hazelkorn. "Learning to Live with League Tables and Ranking: The Experience of Institutional Leaders". In: *Higher Education Policy* 21 (June 2008), pp. 193–215. DOI: 10.1057/hep.2008.1.

[32] Ellen Hazelkorn. "Reflections on a Decade of Global Rankings: what we've learned and outstanding issues". In: *European Journal of Education* 49.1 (2014), pp. 12–28. DOI: 10.1111/ejed.12059.

[33] Ellen. Hazelkorn. *Rankings and the reshaping of higher education: The battle for world-class excellence*. eng. Basingstoke: Palgrave Macmillan, 2011. ISBN: 978-0-230-30639-4.

[34] Diana Hicks. "Performance-based university research funding systems". In: *Research Policy* 41.2 (2012), pp. 251–261. ISSN: 0048-7333. DOI: https://doi.org/10.1016/j.respol.2011.09.007.

[35] John P. A Ioannidis. "Contradicted and Initially Stronger Effects in Highly Cited Clinical Research". eng. In: *JAMA : the journal of the American Medical Association* 294.2 (2005), pp. 218–228. ISSN: 0098-7484. DOI: 10.1001/jama.294.2.218.

[36] Barbara M. Kehm. "Global University Rankings — Impacts and Unintended Side Effects". In: *European Journal of Education* 49.1 (2014), pp. 102–112. DOI: 10.1111/ejed.12064.

[37] Jinseok Kim. "Evaluating author name disambiguation for digital libraries: a case of DBLP". In: *Scientometrics* 116.3 (2018), pp. 1867–1886. DOI: 10.1007/s11192-018-2824-5.

[38] Cyril Labbé. "Ike Antkare one of the great stars in the scientific firmament". In: 2010. URL: https://evaluation.hypotheses.org/files/2010/12/pdf_IkeAntkareISSI.pdf.

[39] Alberto H. F. Laender, Carlos J. P. de Lucena, José Carlos Maldonado, Edmundo de Souza e Silva, and Nivio Ziviani. "Assessing the Research and Education Quality

of the Top Brazilian Computer Science Graduate Programs". In: *SIGCSE Bull.* 40.2 (June 2008), pp. 135–145. ISSN: 0097-8418. DOI: 10.1145/1383602.1383654.

[40] Universiteit Leiden. *CWTS Leiden Ranking: Indicators*. URL: https://www.leidenranking.com/information/indicators (visited on 01/30/2023).

[41] Michael Ley. "DBLP: Some Lessons Learned". In: *Proc. VLDB Endow.* 2.2 (Aug. 2009), pp. 1493–1500. ISSN: 2150-8097. DOI: 10.14778/1687553.1687577.

[42] Michael Ley. "The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives". In: *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*. SPIRE 2002. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 1–10. ISBN: 3-540-44158-1. DOI: 10.1007/3-540-45735-6_1.

[43] Henk F. Moed, Lisa Colledge, Jan Reedijk, Felix Moya-Anegon, Vicente Guerrero-Bote, Andrew Plume, and Mayur Amin. "Citation-Based Metrics Are Appropriate Tools in Journal Assessment Provided That They Are Accurate and Used in an Informed Way". In: *Scientometrics* 92.2 (Aug. 2012), pp. 367–376. ISSN: 0138-9130. DOI: 10.1007/s11192-012-0679-8.

[44] U.S. News and World Report. *How U.S. News Calculated the Best Global Universities Subject Rankings*. URL: https://web.archive.org/web/20191023132902/https://www.usnews.com/education/best-global-universities/articles/subject-rankings-methodology (visited on 02/11/2022).

[45] Juan Luis Nicolau, Juan Pedro Mellinas, and Eva Martín-Fuentes. "The halo effect: A longitudinal approach". In: *Annals of Tourism Research* 83 (2020), p. 102938. ISSN: 0160-7383. DOI: 10.1016/j.annals.2020.102938.

[46] Gokcen Arkali Olcay and Melih Bulu. "Is measuring the knowledge creation of universities possible?: A review of university rankings". In: *Technological Forecasting and Social Change* 123 (2017), pp. 153–160. ISSN: 0040-1625. DOI: 10.1016/j.techfore.2016.03.029.

[47] Ivan Oransky and Adam Marcus. *Gaming the system, scientific 'cartels' band together to cite each others' work*. 2017. URL: https://www.statnews.com/2017/01/13/citation-cartels-science/ (visited on 09/26/2022).

[48] Stefan Otte. "Version Control Systems". In: (Jan. 2009), p. 12. URL: https://www.mi.fu-berlin.de/inf/groups/ag-tech/teaching/2008-09_WS/S_19565_Proseminar_Technische_Informatik/otte09version.pdf.

[49] University of Oulu. *Evaluation based on scientific publishing: Ranking lists*. Sept. 9, 2022. URL: https://web.archive.org/web/20221007183609/https://libguides.oulu.fi/c.php?g=124852&p=3992094 (visited on 01/30/2023).

[50] Stack Overflow. *Stack Overflow Developer Survey 2022*. 2023. URL: https://survey.stackoverflow.co/2022/#section-version-control-version-control-systems (visited on 04/14/2023).

[51] Vaclav Petricek, Ingemar J. Cox, Hui Han, Isaac G. Councill, and C. Lee Giles. "A Comparison of On-Line Computer Science Citation Databases". In: *Research and*

Advanced Technology for Digital Libraries, 9th European Conference, ECDL 2005, Vienna, Austria, September 18-23, 2005, Proceedings. Ed. by Andreas Rauber, Stavros Christodoulakis, and A Min Tjoa. Vol. 3652. Lecture Notes in Computer Science. Springer, 2005, pp. 438–449. DOI: 10.1007/11551362_39.

[52] Shanghai Ranking. *ShanghaiRanking Academic Excellence Survey 2022 Methodology*. 2022. URL: https://www.shanghairanking.com/activities/aes/method/2022 (visited on 10/22/2022).

[53] Shanghai Ranking. *ShanghaiRanking's Global Ranking of Academic Subjects Methodology 2022*. 2022. URL: https://www.shanghairanking.com/methodology/gras/2022 (visited on 10/22/2022).

[54] Florian Reitz and Oliver Hoffmann. "An Analysis of the Evolving Coverage of Computer Science Sub-fields in the DBLP Digital Library". In: *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*. Ed. by Mounia Lalmas, Joemon M. Jose, Andreas Rauber, Fabrizio Sebastiani, and Ingo Frommholz. Vol. 6273. Lecture Notes in Computer Science. Springer, 2010, pp. 216–227. DOI: 10.1007/978-3-642-15464-5_23.

[55] Kent V. Rondeau. "The Impact of World Ranking Systems on Graduate Schools of Business: Promoting the Manipulation of Image over the Management of Substance". In: *World Journal of Education* 7 (2017), pp. 62–73. ISSN: 1925-0746. URL: https://eric.ed.gov/?id=EJ1157656.

[56] Vicente Safón. "Can the reputation of an established business school change?" In: *Management in Education* 26.4 (2012), pp. 169–180. DOI: 10.1177/0892020611433081.

[57] Vicente Safón. "Inter-Ranking Reputational Effects: An Analysis of the Academic Ranking of World Universities (ARWU) and the Times Higher Education World University Rankings (THE) Reputational Relationship". In: *Scientometrics* 121.2 (Nov. 2019), pp. 897–915. ISSN: 0138-9130. DOI: 10.1007/s11192-019-03214-9.

[58] Vicente Safón. "What do global university rankings really measure? The search for the X factor and the X entity". In: *Scientometrics* 97.2 (2013), pp. 223–244. DOI: 10.1007/s11192-013-0986-8.

[59] Vicente Safón and Domingo Docampo. "Analyzing the Impact of Reputational Bias on Global University Rankings Based on Objective Research Performance Data: The Case of the Shanghai Ranking (ARWU)". In: *Scientometrics* 125.3 (Dec. 2020), pp. 2199–2227. ISSN: 0138-9130. DOI: 10.1007/s11192-020-03722-z.

[60] Jung Cheol Shin. "Organizational Effectiveness and University Rankings". In: *University Rankings: Theoretical Basis, Methodology and Impacts on Global Higher Education*. Ed. by Jung Cheol Shin, Robert K. Toutkoushian, and Ulrich Teichler. Dordrecht: Springer Netherlands, 2011, pp. 19–34. ISBN: 978-94-007-1116-7. DOI: 10.1007/978-94-007-1116-7_2.

[61] Li Tang and John P. Walsh. “Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps”. In: *Scientometrics* 84.3 (2010), pp. 763–784. DOI: 10.1007/s11192-010-0196-6.

[62] National Taiwan University. *NTU World University Performance Indicators*. URL: http://nturanking.csti.tw/methodoloyg/indicators (visited on 01/30/2023).

[63] *University Ranking by Academic Perfomance: Methodology*. URL: https://urapcenter.org/Methodology (visited on 01/30/2023).

[64] Moshe Y. Vardi. “Academic Rankings Considered Harmful!” In: *Commun. ACM* 59.9 (Aug. 2016), p. 5. ISSN: 0001-0782. DOI: 10.1145/2980760.

[65] Wikipedia. *Percentile rank*. 2022. URL: https://en.wikipedia.org/wiki/Percentile_rank (visited on 10/30/2022).

[66] Yan Wu, Srinivasan Venkatramanan, and Dah Ming Chiu. “A Population Model for Academia: Case Study of the Computer Science Community Using DBLP Bibliography 1960-2016”. In: *IEEE Transactions on Emerging Topics in Computing* 9.1 (2021), pp. 258–268. DOI: 10.1109/TETC.2018.2855156.

APPENDIX A: CSV FILES CONTAINING INFORMATION OF FACULTY MEMBERS

Name	Affiliation	Homepage	ScholarID
A Min Tjoa	TU Wien	http://www.ifs.tuwien.ac.at/tjoa	x8qCMhcAAAAJ
A. Akbari Azirani	IUST	http://ce.iust.ac.ir/page.php?slct_pg_id=6537&sid=14&slc_lang=en	pCil4_cAAAAJ
A. Akbariazirani	IUST	http://ce.iust.ac.ir/page.php?slct_pg_id=6537&sid=14&slc_lang=en	pCil4_cAAAAJ
A. Aldo Faisal	Imperial College London	https://www.imperial.ac.uk/people/a.faisal	WjHjbrwAAAAJ
A. Antony Franklin	IIT Hyderabad	http://www.iith.ac.in/~antony/index.html	LVfqLuoAAAAJ
A. B. M. Alim Al Islam	BUET	https://sites.google.com/site/abmalimalislam	K-AIPzQAAAAJ
A. B. Siddique 0001	University of Kentucky	http://cs.uky.edu/~siddique	pVnchfsAAAAJ
A. Bhaskar	BITS Pilani-Goa	https://www.bits-pilani.ac.in/goa/abaskar/profile	WNrQSakAAAAJ
A. C. Cem Say	Bogaziçi University	https://www.cmpe.boun.edu.tr/~say	rOum2XsAAAAJ
A. C. W. Finkelstein	City University of London	https://finkelstein.uk	n8xuCVkAAAAJ

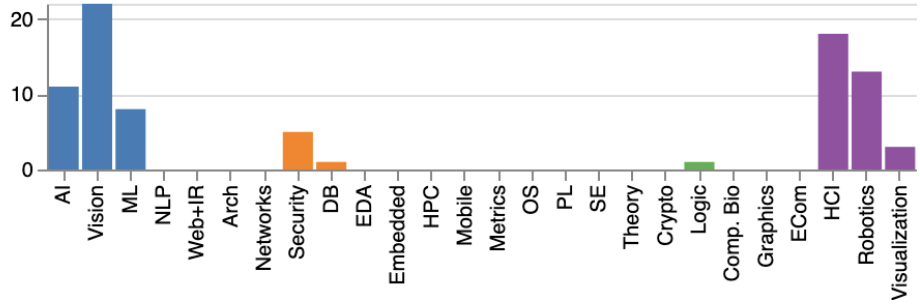
Table A.1. *The first 10 records of* csrankings-a.csv*.*

Name	DBLP Name	Affiliation	Homepage	ScholarID	Job title
Alessandro Foi	Alessandro Foi	Tampere University	https://www.tuni.fi/en/alessandro-foi	zBmF3ZAAAAAJ	Professor of Signal Processing
Annamaria Mesaros	Annamaria Mesaros	Tampere University	https://researchportal.tuni.fi/en/persons/annamaria-mesaros	tOvdEZIAAAAJ	Academy Research Fellow
Antonis Michalas	Antonis Michalas	Tampere University	https://www.tuni.fi/en/antonios-michalas	ovYlEZQAAAAJ	Associate Professor (tenure track) Cyber security
Ari Visa	Ari Visa	Tampere University	https://www.tuni.fi/en/ari-visa	3mVXg3QAAAAJ	Professor of Signal Processing
Atanas P. Gotchev	Atanas P. Gotchev	Tampere University	https://www.tuni.fi/en/atanas-gotchev	quqBZjUAAAAJ	Professor of Signal Processing
Billy Bob Brumley	Billy Bob Brumley	Tampere University	https://www.tuni.fi/en/billy-brumley	7QX0AQYAAAAJ	Associate Professor (tenure track)
David Hastbacka	David Hastbacka	Tampere University	https://www.tuni.fi/en/david-hastbacka	1pgBaqAAAAAJ	Assistant Professor (tenure track)
Davide Taibi	Davide Taibi 0001	Tampere University	https://www.tuni.fi/en/davide-taibi	ToWAPrcAAAAJ	Associate Professor (tenure track)
Eero Hyry	Eero Hyry	Tampere University	https://www.tuni.fi/en/eero-hyry	qmT-ubgAAAAJ	Professor Mathematics
Esa Rahtu	Esa Rahtu	Tampere University	https://www.tuni.fi/en/esa-rahtu	SmGZwHYAAAAJ	Associate Professor (tenure track) Signal Processing (Intelligent Machines)

Table A.2. *The first 10 records of* master.csv.

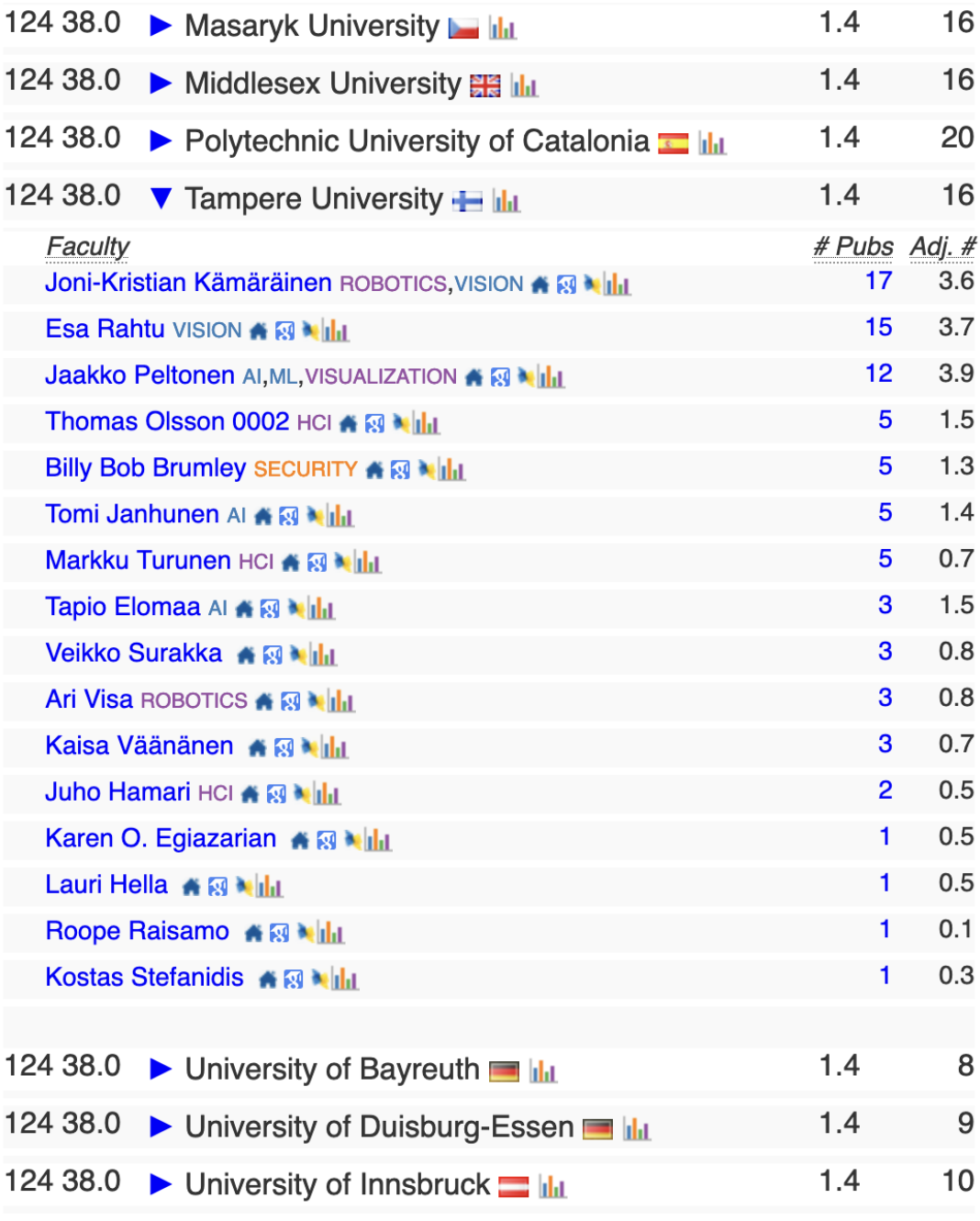
APPENDIX B: TAMPERE UNIVERSITY IN WORLD, EUROPE, AND FINLAND RANKING

337	39.1	National Cheng Kung University	1.4	22
337	39.1	Polytechnic University of Catalonia	1.4	20
337	39.1	SUFE	1.4	7
337	39.1	Sharif University of Technology	1.4	10
337	39.1	Tampere University	1.4	16



337	39.1	Texas State University	1.4	13
337	39.1	Tongji University	1.4	10
337	39.1	UCCS	1.4	9
337	39.1	Universidad de Chile	1.4	10
337	39.1	University of Arkansas	1.4	10
337	39.1	University of Bayreuth	1.4	8
337	39.1	University of Duisburg-Essen	1.4	9
337	39.1	University of Hawaii at Manoa	1.4	11
337	39.1	University of Innsbruck	1.4	10
337	39.1	University of Kentucky	1.4	12
337	39.1	University of Kiel	1.4	8

Figure B.1. *Tampere University in the world ranking (2000–2022).*



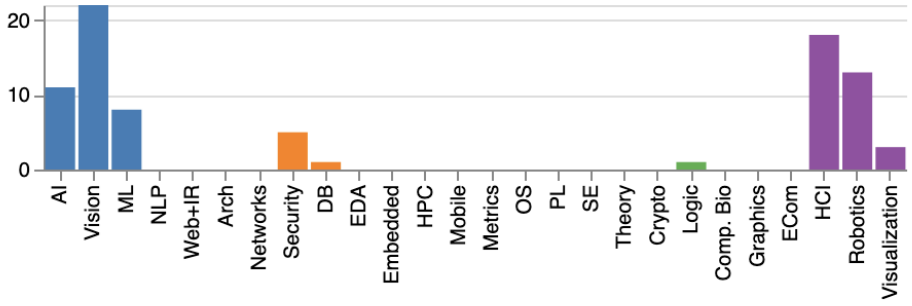
124	38.0	▶ Masaryk University	1.4	16
124	38.0	▶ Middlesex University	1.4	16
124	38.0	▶ Polytechnic University of Catalonia	1.4	20
124	38.0	▼ Tampere University	1.4	16

Faculty	*# Pubs*	*Adj. #*
Joni-Kristian Kämäräinen ROBOTICS,VISION	17	3.6
Esa Rahtu VISION	15	3.7
Jaakko Peltonen AI,ML,VISUALIZATION	12	3.9
Thomas Olsson 0002 HCI	5	1.5
Billy Bob Brumley SECURITY	5	1.3
Tomi Janhunen AI	5	1.4
Markku Turunen HCI	5	0.7
Tapio Elomaa AI	3	1.5
Veikko Surakka	3	0.8
Ari Visa ROBOTICS	3	0.8
Kaisa Väänänen	3	0.7
Juho Hamari HCI	2	0.5
Karen O. Egiazarian	1	0.5
Lauri Hella	1	0.5
Roope Raisamo	1	0.1
Kostas Stefanidis	1	0.3

124	38.0	▶ University of Bayreuth	1.4	8
124	38.0	▶ University of Duisburg-Essen	1.4	9
124	38.0	▶ University of Innsbruck	1.4	10

Figure B.2. *Tampere University in the Europe ranking (2000–2022).*

#	PR	Institution	Count	Faculty
1	83.3	▶ Aalto University	2.6	38
2	50.0	▶ University of Helsinki	1.8	23
3	16.7	▼ Tampere University	1.4	16



Faculty	*# Pubs*	*Adj. #*
Joni-Kristian Kämäräinen ROBOTICS,VISION	17	3.6
Esa Rahtu VISION	15	3.7
Jaakko Peltonen AI,ML,VISUALIZATION	12	3.9
Thomas Olsson 0002 HCI	5	1.5
Billy Bob Brumley SECURITY	5	1.3
Tomi Janhunen AI	5	1.4
Markku Turunen HCI	5	0.7
Tapio Elomaa AI	3	1.5
Veikko Surakka	3	0.8
Ari Visa ROBOTICS	3	0.8
Kaisa Väänänen	3	0.7
Juho Hamari HCI	2	0.5
Karen O. Egiazarian	1	0.5
Lauri Hella	1	0.5
Roope Raisamo	1	0.1
Kostas Stefanidis	1	0.3

Figure B.3. *Tampere University in the Finland ranking (2000–2022).*