IT Education in Clouds and Clouds in IT Education

J. Henno*, H. Jaakkola** and J. Mäkelä***
* Tallinn University of Technology, Estonia
** Tampere University, Pori
*** University of Lapland, Rovaniemi, Finland
jaak@cc.ttu.ee

Abstract **– The rapid development of information and communication technology (ICT) and growing participation of students in work life has already in several decades moved ICT education into 'clouds', using sources of knowledge on Internet from all around the world. The COVID pandemic has increased this process, forcing universities to restrict classroom teaching and rapidly increased student's self-study.**

At the same time, increase of amounts of data to be processed is constantly introducing new high-level software technologies, layers and layers of packages and libraries, deeper and more complex. This has created a new 'top-down' programming style: a new project is started with importing mass of libraries which have been used in earlier projects and only then is considered how to use them in order to solve the programming task. The self-studying ICT students see only tips of modern software icebergs and it is difficult for them to understand their working without face-to-face classroom communication where details of the 'depths' are explained.

Keywords - ICT; self-study; software complexity; COVID

I. Introduction.

Humanity is obsessed with growth, but we have already for years overusing possibilities for quantitative growth [1],[2], thus we have intensively develop qualitative means for growth [3],[4]. According to UNESCO Science report 2021 [5], global research spending grows faster than other sectors of the World Economy. There are already 8.8 million scientists worldwide and the number is growing. Thus global research production of science and engineering publications is also rapidly growing, e.g. in USA the annual growth has been 4% during the last 10 years [6].

Scientific research together with statistical data collected from various fields of human activities are creating a world-wide overflow of data [7].

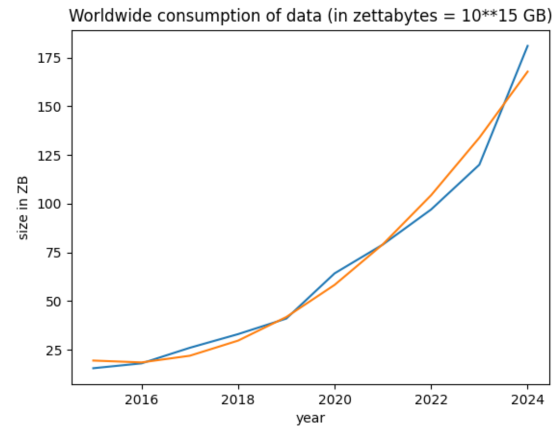

Figure 1. Worldwide use of data (blue line) and the quadratic model of the process (red).

The size of the Mipro conference [8] publications is also growing (as a part of this process).

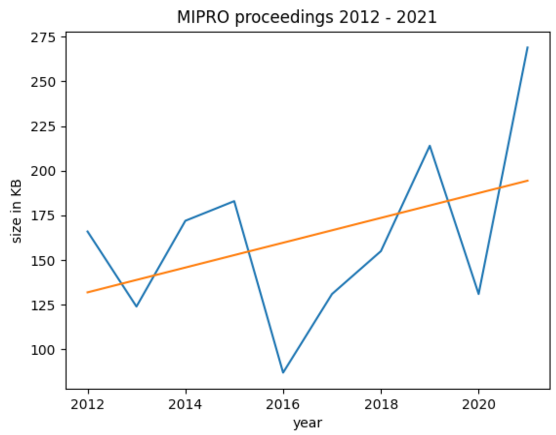

Figure 2. Growth of accepted publication to the Mipro conference coverence (by the size of proceedings [9]) and the linear model of the process

Most of this overflow of data is open and stored in various open-source repositories. The continuous growth of new knowledge is also reflected in education. Trying to comply with rising demand for remote learning makes colleges and universities to move cloud-based platforms [10]. Education is increasingly becoming virtual [11].

II. Computing in Clouds

The modern world-wide data processing infrastructure (computer + mobile networks, sometimes referred as MCC - Mobile Cloud Computing [12]) covers all continents and countries and is rapidly growing. According to recent forecast [13] the global cloud computing market size is expected to reach USD 1,251.09 billion by 2028, that the annual growth over the period being 19.1%. The growth of cloud computing in education is expected even bigger – 25.6% [14]. In 2021 nearly half (42%) of enterprises in EU used cloud computing [15].

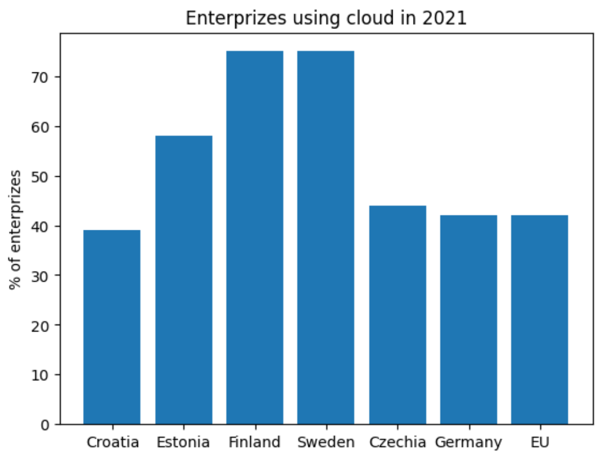

Figure 3. Percentage of enterprizes which used cloud in 2021 [16].

The worldwide cloud is besides quantitatively developing also qualitatively. Cloud is a service – you do not own anything, you pay for use. Besides the big cloud services (Microsoft Azure, Google Cloud, Amazon AWS) have appeared several new ones - Heroku, GearHost, Gamatera, Red Hat OpenShift, Oracle Cloud, Fly.io, Classroom.cloud etc, [17], which offer various prize plans and services; conditions vary depending on your country.

The first service in cloud was just provide connection. But soon appeared also applications (SaaS – Sofware as a Service, IaaS – Infrastructure as a Service), e.g. e-mail and messaging and then Microsoft presented all of Microsoft Office programs as an on-line service and many cloud providers followed. The next step was PaaS – Platform as a Service – a set of interconnected programs for some activity – database, text translation, text-to-speech/speech-to-text, chatbots, Machine Learning (ML) etc. For instance, the Microsoft Azure [18] proposes several services, programs and programming platforms: Bing, WindowsLive, OfficeLive, SQL, Microsoft Advertising, SharePoint, .Net etc, but in PaaS client is not driving, he can only select from presented by the platform opportunities.

For education in the clouds the quickest annual growth until 2027 is expected just for the PaaS sector – 22.4% [19] – education is expected to become more-and more a service like McDonald's – you just select from a menu available options – amount of fries and/or ketchup.

Cloud-based platforms for learning offer many features which give them edge over traditional education methodologies, first of all their integration with smartphones and other internet-connected devices which can facilitate learning from any location. A bit over-emphasized seems in some of them functionality for the social interaction what makes several of them to look like 'Facebooks for Education'.

There are also strictly technical platforms; which usually are intended for business, but can also be used for education. For instance, the PlayFab [20] (sub-platform of the Microsoft Azure) is designed for creating commercial online games (also for multiplayer games). The platform has functions for creating new players (following strict security rules) and collecting results of games - scoreboards. The features of scoreboards (e.g. ordering by results) allow to use them for managing student's results in tests/exams, thus in Tallinn University of Technology was tested a test system, where teacher(s) could upload from a browser a test (as a multiplayer game) and get sorted results as PlayFab scoreboards. Uploading a test and getting results happens from a www-browser and can be done anywhere - no Moodle managers needed. For this experimental system tests were created with the free Half-Backed Potatoes system [21], but a student created a similar system for Google's Firebase [22] describing tests with a spreadsheet (as a Kahoot [23] game). Both these systems allowed multiplayer (i.e. parallel) tests, where all participants started and ended the test at the same time, i.e. everyone was competing with everyone in a limited time. Game/test ended when some player announced 'Ready!' or when pre-set time (e.g. 5 min) run out, thus exluding a situation (common in traditional exams) when student is pleading "No, I'm not yet ready, I need some more time!" – but in the last five minutes has not added anything to his test paper. In our competition-based society timeliness is also an ability what we should teach to our students.

III. Growing complexity of computing

A. *'Top-down' computing*

In order to be useful from data should be extracted the 'essence' – knowledge. Tremendous growth of data has created research directions aimed at extracting new knowledge from data: Data Science (DS), Machine Learning (ML), Artificial Intelligence (AI). For handling massive amounts of data in clusters, grids and virtual networks have been developed many new, higher-level techniques and tools – new programming languages (R, Go, Julia, Swift etc), pre-compiled (for speed) modules (useful functions that eliminate the need for writing your own code), libraries (collections of modules) and frameworks (which already govern the whole process of creating a new software product).

The 'Lingua Franca', the common of all areas of Data Science is Python [24] (sometimes called 'everyone's second language' - Python has been among the 10 most popular programming languages every year since 2004 [25], in February 2022 Python was on the top, growing 15.33%). Its 'rising above others' has several reasons (e.g. from 'Zen of Python" [26]: *'Beautiful is better than ugly', 'Simple is better than complex', 'Readability counts'* etc.), but one of the main reasons have been libraries – you do not have to 'invent a bicycle' –somebody has already done this and stored the code as a (open-source) library.

Currently are in the Python Package Index [27] (open-source repository for Python code) stored over 5.6 million files.

The number of libraries is rapidly growing also for other languages, e.g. during the first week of February 2022, Feb 1st - Feb 7th were every day for the `node.js` [28] platform uploaded in average 819 modules [29].

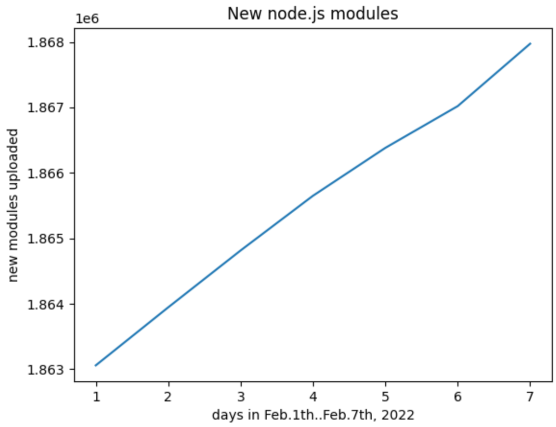

Figure 4. Number of uploaded new node.js modules during Feb.1st..Feb.7th, 2022 [29]

B. *A Light-weight Example*

Use of pre-built modules and libraries make creation of new pieces of software fabulously easy. For instance, the graph in Fig.2 was created with few lines of Python 3 code (the lines starting with # are commentaries for humans; for ease of referencing lines are numbered – the numbers are not part of the code) and two Python 3 libraries: `numpy` (numeric Python) and `matplotlib` (library for plotting mathematical functions; for convenience are packages given shorter names):

```
import numpy as np
import matplotlib.pyplot as plt
# x-axis values
x = range(2012,2022)  # integers from
2012 to 2021
# corresponding y-axis values
y =
[166,124,172,183,87,131,155,214,131,269]
# plotting the graph
plt.plot(x, y)
# naming the axes
plt.xlabel('year')
plt.ylabel('size in KB')
# giving a title to the graph
plt.title('MIPRO proceedings 2012 -
2021')
# computing coefficients for the linear
model, m = slope, b=intercept
m, b = np.polyfit(x, y, 1)
#draw the line
plt.plot(x, m*x + b)
# show the plot
plt.show()
```

Nowadays this text should be understandable also to non-programmers.

C. *What happens under the hood*

But programmers studying the above code will wonder – no loops ??? The 'classical' program, i.e. using only commands from a programming language without any libraries would have to use several loops, e.g. projecting values to axes. Calculating approximation (regression) model involves (in theory) differentiating the minimal squares model and then solving the system of linear equations – here the regression parameters `m,b` appear in one line. The AI (Artificial Intelligence) – calculating the size and visually pleasing placement of axes - is totally hidden.

The main task of Data Science is reducing the size of data, extracting knowledge by creating models. In the above code a single line produces coefficients `m,b` of the linear regression function. For making prediction for future years this line replaces need to know ten y-values with only two numbers. It is easy to create a more complex models, e.g. the second-order model in Fig.1 is obtained modifying only two lines in the above program:

```
n,m,b = np.polyfit(x,y,2)
plt.plot(x,n*x*x+m*x+b)
```

All the functionality of the above program is hidden in the two explicitly called libraries `numpy, matplotlib` and many others called by these two. The `pyinstaller` [30] tool reveals, that for drawing of this graph are imported altogether 162 modules (and 28 items still appeared as `MissingModule`):

```
bootlocale • _collections_abc • _weakrefset …
types • warnings • weakref
```

The added modules increased the size (on disk) of the program 31882 times – from 2 KB to 63764 KB. It is impossible to know what all these 162 modules actually do and why they are needed.

Such 'blow-up' is a common feature of modern software. The developed by Facebook JavaScript library `React.js` advertises that "*React makes it painless to create interactive User Interfaces*" [31]. The very first example, a web page with text "Hello World" [32] loads for creating this functionality three JavaScript libraries with 73571+3358+26275 = 103204 lines of JavaScript code (but the resulting page can be created without any JavaScript). For a student this is not "painless", but (very) painful. `React.js` was created by Facebook (now Meta) programmers to create and handle many www-pages with different structure (task for 72000 programmers in 2022 [33]). But something with 103204 lines of code is not a tutorial for students, they start with a one page.

But 40.14% of web developers use `React.js` [34]. The main burden – 73571 lines of code (after un-minifying) comes from the Babel library for interpreting Ecmascript ES6 (the official name for JavaScript). All major browsers can handle ES6 already for last five years [35], thus Babel is not needed [36], it only increases page's size and slows it down, but with 'copy-paste' programming style it is difficult to understand that something is obsolete ('All previous examples used that!').

Software technology is constantly and rapidly developing and changing, thus software packages also grow old, become obsolete or their APIs change. Use of outdated libraries makes difficult (sometimes impossible) to perform the necessary modifications, software maintenance becomes time-consuming, labor intensive and error-prone. When the `jQuery` library appeared back in 2006 it was needed to reduce developer's problems with the Microsoft's Internet Explorer 6 browser, which did not follow standards and was dangerous to use (the IE 6 has been included in the list of the 25 Worst Tech Products of all time [37]). The `jQuery` parses the HTML document, but all modern browsers can do this and do not need any help, thus currently its main function has become obsolete [38] [39] [40], it just duplicates browser's functionality. Use of `jQuery` introduces several excessive operations (request to download `jQuery`, downloading `jQuery`, passing link of page's text to `jQuery`, processing in `jQuery`, passing results back to browser), thus slows down the page speed by 30-50% [41]. But 'developers', whose main technology is 'copy-paste' from some existing website often do not even understand, why they are copying something, thus the `jQuery` is still used in 78% of the top first million websites [38].

D. *Problems for IT Students*

The current pandemic has greatly increased student's self-study. Already in 2017 the major text-books publisher McGraw Hill found that 94% of students say that digital learning helps them to retain new material [42]. Internet contains many high-level sources of IT knowledge. But it contains also many 'fake news', libraries, created with the only desire of creator to advertise himself, some directly dangerous downloads [43] [44] or deliberately corrupted libraries [45].

Modern software is like icebergs in Antarctica – the main part is hidden. Self-studying students see only the top. It is quite difficult for students to 'dig deep', understand the hidden functionality, understand cryptic error messages which come from somewhere (what means "*can't multiply sequence by non-int of type 'numpy.float64'*" ?) and cannot assess the real value of proposed on Internet novelties, which are advertised with meaningless statements "*It just works*". A student claimed (when presenting his diploma project) that he does not use css-stylesheets – he uses Bulma [46], which advertises itself as "*the modern CSS framework that just works. No CSS knowledge required*". Student was a quite proper target to use something what does not require knowledge – his css-file contained 15301 css-rules for application having less than 100 html-elements to use these rules. Student had got this css-monstrum by copy-paste and did not know anything about its content – 182 rules were addressing object '*hero*', what his application did not have. For comparison – css-file of one of the world's biggest newspapers 'The new York Times" weekend edition has 455 css-rules for tens of thousands of html-elements.

The enormous hidden depth is present in all modern software. Tree lines of code for neural net model in a Python 3 program creates a model with nearly 4 million parameters (output from the Python command `model.summary()`):

```
vocab_size:  18
(64, 100, 18) # (batch_size,
sequence_length, vocab_size)
Model: "sequential"
___________________________________________________
_______________________
Layer (type)                 Output Shape
Param #
===================================================
=======================
embedding (Embedding)      (64, None, 256)
4608
___________________________________________________
_______________________
gru (GRU)                  (64, None, 1024)
3938304
___________________________________________________
_______________________
dense (Dense)                (64, None, 18)
18450
===================================================
=======================
Total params: 3,961,362
Trainable params: 3,961,362
Non-trainable params: 0
```

With these nearly 4 million parameters the neural net inferred from the texts of presentations from the Mipro "Computers in Education" sub-conference that in Croatia teachers are teaching computers and here are used quite complex computers (a brief excerpt from the program's output to seed "*students*") [47]:

Computers teachers and the actor model is that every course controls, the comparison of the most popular programming courses could be included in the class and all of them with some topics of the course teachers in the process of developing and procedures and other reasons for the analyzing the control group (communication sciences in the classroom. Computers in Croatia are complex and addressed by the authors of the students themselves in the process of learning and education to the students who have to do by developing student their results.

In another test with the same input – a text corpus created from "Computers in Education" presentations computer stated bluntly:

Students are not included in the process of e- learning in the context of the process of teaching and learning.

Such results make many features of modern computers to seem as non-deterministic, random [48]. For instance, if somebody wants to get a high-order model for data about Mipro presentations (Fig.2) using a neural net, then most probably the result will look like this:

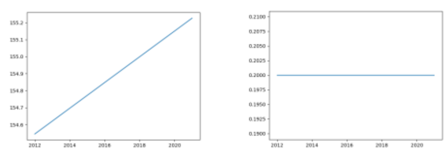

Figure 5. Possible results when trying to model data about Mipro presentations with Tensorflow – there are too few points

Use of modules, libraries and platforms is inevitable in modern software production, they enhance modularity and essentially reduce development time. But their 'copy-paste' use encompasses for students many problems.

Many IT students work as professional programmers already on their first-second year of studies. Tasks what commercial firms present to them are in several orders more complex than the above graph-plotting program and for creating a solution they are instructed to start with importing tens of libraries. Thus they practice and learn 'top-down' programming – starting with the upper-level general structure of their programming project, importing popular libraries (whose functionality they do not totally understand) and hoping, that these libraries miraculously solve all the bottom-level problems. For example, a student could program a test (as a multiplayer www and mobile game) in the Google-s www and mobile framework Firebase [22]. Program presented to users multiple-select questions and assessed their answers, but repeated questions – often for the next question appeared the same what user just got and author was not able to correct this (trivial) problem – he could not find such functionality from Firebase functions, thus could not eliminate repetitions.

IV. Conclusions

The world-wide pandemic has (and still is) essentially disrupted education. According to UNESCO "*More than 1.5 billion students and youth across the planet are or have been affected by school and university closures due to the COVID-19 pandemic.*"[49] This has greatly increased student's self-studies from Internet resources.

This 'moving online' is forcing the currently most faster-developing subarea of IT – the Data Science [50], [51]. While it is possibly not the most asked skill in US [52], it is still one of the most sought ability worldwide and demand is much greater than supply [53].

Self-learning IT students do not see the general picture, the hidden under multiple layers of libraries body of software icebergs, they do not understand contexts and connections and often do not apply their own judgment – "*Everybody does so!*" or "*This simply works!*". Thus in classroom one of the most important ones is 'moving back to roots' explaining students the reasons why some IT technology, protocol or library is just like it is and how it become such. The Internet is a marvelous source of many superb materials for self-study of IT technologies, but most of WWW-tutorials are totally functional recipes: import these libraries, then write those lines of code – with minimal explanations, why just these libraries and just those lines of code or using the totally meaningless explanation "*It Just Works*!" The university classroom should provide all missing explanations and reasons.

References

[1] D.H. Meadows, J.Randers, D.L. Meadows. Limits to Growth: The 30-Year Update. ISBN-10 : 193149858X, 338 pp

[2] G.Herringthon. Update to limits to growth: Comparing the World3 model with empirical data. https://onlinelibrary.wiley.com/doi/10.1111/jiec.13084

[3] T. Jackson. Post Growth. 2021, ISBN-13 978-1-5095-4251-2, 240pp

[4] S. Wagenknecht. Prosperity Without Greed. 2016, ISBN 978-3-593-50758-3, 267 pp

[5] UNESCO Science Report: The race against time for smarter development. https://www.unesco.org/reports/science/2021/en/download-the-report

[6] Publications Output: U.S. Trends and International Comparisons. https://ncses.nsf.gov/pubs/nsb20206/

[7] Total data volume worldwide 2010-2025. https://www.statista.com/statistics/871513/worldwide-data-created/

[8] Mipro. http://www.mipro.hr/

[9] MIPRO proceedings (2012 - 2021). http://www.mipro.hr/Dokumentim/Zbornici/tabid/196/language/en-US/Default.aspx

[10] https://www.infoworld.com/article/3608716/higher-education-embraces-cloud-computing-just-in-time.html

[11] Henno, J., Jaakkola, H., Mäkelä, J. Teaching for virtual work. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings 8756778, pp. 818-826

[12] Mobile Cloud Computing. https://www.eescorporation.com/mobile-cloud-computing/

[13] Cloud Computing Market Size, Share & Trends Analysis Report by Service (SaaS, IaaS), by Enterprise Size (Large Enterprises, SMEs), by End Use (BFSI, Manufacturing), by Deployment, and Segment Forecasts, 2021-2028. https://www.researchandmarkets.com/reports/5397840/cloud-computing-market-size-share-and-trends

[14] Cloud Computing in Education Market by Service Model (SaaS, PaaS, and IaaS), Deployment Model (Private Cloud, Public Cloud, Hybrid Cloud, and Community Cloud), User Type (K-12 and Higher Education) and Region - Global Forecast to 2021. https://www.marketsandmarkets.com/Market-Reports/cloud-computing-education-market-17863862.html

[15] Eurostat. Cloud computing used by 42% of enterprises. https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20211209-2

[16] Eurostat. https://ec.europa.eu/eurostat/statistics-explained/index.php

[17] 7 Best Completely FREE Cloud Hosting Services [2022]. https://www.websiteplanet.com/blog/best-free-cloud-hosting-services/

[18] Azure. https://azure.microsoft.com/

[19] Cloud Computing in Education - Global Market Trajectory & Analytics. https://www.businesswire.com/news/home/20210504006048/en/Cloud-Computing-in-Education-Market---Global-Trajectory-Analytics-to-2027---ResearchAndMarkets.com

[20] Playfab. https://playfab.com/

[21] Hot Potatoes. http://hotpot.uvic.ca/

[22] Firebase. https://firebase.google.com/

[23] Kahoot!. https://kahoot.com/

[24] Python. https://www.python.org/

[25] Tiobe. https://www.tiobe.com/tiobe-index/

[26] Zen of Python. https://docs.python-guide.org/writing/style/#zen-of-python

[27] PyPi. https://pypi.org/

[28] node.js. https://nodejs.org/en/

[29] New Modules. http://www.modulecounts.com/

[30] pyinstaller. https://pypi.org/project/pyinstaller/

[31] React. https://reactjs.org/

[32] React Getting Started. https://www.w3schools.com/react/react_getstarted.asp

[33] Number of full-time Meta employees from 2004 to 2021. https://www.statista.com/statistics/273563/number-of-facebook-employees/

[34] https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

[35] Browser Support for ECMAScript 2016. https://www.w3schools.com/js/js_versions.asp

[36] Do you need Babel for ES6? https://www.sidmartinbio.org/do-you-need-babel-for-es6/

[37] The 25 Worst Tech Products of All Time. https://www.pcworld.com/article/535838/worst_products_ever.html

[38] https://thenewstack.io/why-outdated-jquery-is-still-the-dominant-javascript-library/

[39] You Might not Need jQuery. https://youmightnotneedjquery.com/

[40] Active jQuery Usage and Its Problem. https://love2dev.com/blog/jquery-obsolete/

[41] https://www.measurethat.net/Benchmarks/Show/6633/1/jquery-vs-getelementbyid-vs-queryselector

[42] McGraw Hill. https://www.mheducation.com/highered/explore/studytrends.html

[43] Malware discovered in JavaScript Library accessed by millions each week. https://www.itpro.co.uk/security/361347/malware-in-javascript-library-accessed-by-millions-each-week

[44] Twelve malicious Python libraries found and removed from PyPI. https://www.zdnet.com/article/twelve-malicious-python-libraries-found-and-removed-from-pypi/

[45] When open-source developers go bad. https://www.zdnet.com/article/when-open-source-developers-go-bad/

[46] Bulma. https://bulma.io/

[47] J. Henno, H. Jaakkola, J. Mäkelä. Machine Learning vs Human Learning. 44th International Convention on Information, Communication and Electronic Technology, MIPRO 2021 - Proceedings, 2021, pp. 542–548, ISSN 1847-3946

[48] Henno, J. Jaakkola, H. Mäkelä, J. Non-determinism in nowadays computing and IT education2020 43nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2020, pp 794-801 .

[49] UNESCO. Global Education Coalition. https://en.unesco.org/covid19/educationresponse/globalcoalition

[50] Five of the fastest-growing tech jobs in 2022. https://www.mygreatlearning.com/blog/fastest-growing-tech-jobs/

[51] Employers are desperate for data scientists. https://www.zdnet.com/article/employers-are-desperate-for-data-scientists-as-demand-booms/

[52] Is Data Science Still a Rising Career in 2021. https://towardsdatascience.com/is-data-science-still-a-rising-career-in-2021-722281f7074c

[53] Demand for Data Scientists Far Exceeds Supply. https://dataconomy.com/2019/12/demand-for-data-scientists-far-exceeds-supply-heres-how-your-company-can-cope/