



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

WEB APPLICATION MODULE FOR THE DESIGN OF CYBERSECURITY CURRICULA

MODUL WEBOVÉ APLIKACE PRO TVORBU OSNOV KYBERNETICKÉ BEZPEČNOSTI

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Jiří Hrachovina

SUPERVISOR

VEDOUCÍ PRÁCE

M.Sc. Sara Ricci, Ph.D.

BRNO 2024

Bachelor's Thesis

Bachelor's study program **Information Security**

Department of Telecommunications

Student: Jiří Hrachovina

ID: 240623

**Year of
study:** 3

Academic year: 2023/24

TITLE OF THESIS:

Web application module for the design of cybersecurity curricula

INSTRUCTION:

This Bachelor thesis describes the design and implementation of an optimization program designed to find the best combination among a selection of courses to cover specific cybersecurity skills. The goal of this thesis is to get acquainted with the European Cybersecurity Skills Framework (ECSF) in the area of cybersecurity, optimization algorithms, and the REWIRE project. This thesis also includes web designing and working with a database. The program will be developed in Python, PHP, and Javascript programming languages. A critical requirement of the program is finding the best solution in a reasonable time.

RECOMMENDED LITERATURE:

[1] HAJNY, Jan, Marek SIKORA, Athanasios VASILEIOS GRAMMATOPOULOS a Fabio DI FRANCO, 2022. Adding European Cybersecurity Skills Framework into Curricula Designer. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22): Association for Computing Machinery. New York, NY, USA, 1–6.

[2] S. SKIENA, Steven, 2008. The Algorithm Design Manual. 2. London: Springer London. ISBN 978-1-84800-070-4.

**Date of project
specification:** 5.2.2024

**Deadline for
submission:** 28.5.2024

Supervisor: M.Sc. Sara Ricci, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
Chair of study program board

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

This thesis focuses on enhancing the web-based application REWIRE Cybersecurity Profiler by integrating a machine learning algorithm for course description analysis. This component enables the automatic analysis of course descriptions and links the resulting skill groups to the ENISA ECSF profiles. Additionally, several improvements to the existing application have been proposed, including the implementation of functional URLs, visualization of visitor data through graphs, generation of course descriptions using ChatGPT, and optimization of user interface elements. Finally, the thesis includes a demonstration of the application and proof of its utility in designing study programs.

KEYWORDS

Curricula design, Cybersecurity, Cybersecurity profiler, Machine learning, REWIRE Cybersecurity profiler, Software development, Web application

ABSTRAKT

Tato práce se zaměřuje na vylepšení webové aplikace REWIRE Cybersecurity Profiler integrací algoritmu strojového učení na analýzu popisů kurzů. Tento komponent umožňuje automatickou analýzu popisů kurzů a propojuje výstupní dovednostní skupiny s profily ENISA ECSF. Práce také navrhuje několik vylepšení existující aplikace a to implementaci funkčních URL, vizualizaci návštěvnických dat skrze grafy, generování popisů kurzů pomocí ChatGPT a optimalizaci prvků uživatelského rozhraní. Práce také obsahuje demonstraci aplikace a její užitečnosti v navrhování studijních programů.

KLÍČOVÁ SLOVA

Návrh osnov, Kybernetická bezpečnost, Cybersecurity profiler, Strojové učení, REWIRE Cybersecurity profiler, Vývoj softwaru, Webová aplikace

ROZŠÍŘENÝ ABSTRAKT

V posledních letech výrazně vzrostla poptávka po specialistech Informační bezpečnosti. I přes tento nárůst poptávky je na globálním pracovním trhu jejich značný nedostatek. Značným problémem je nesoulad mezi požadavky pracovního trhu a vzděláním nabízeným vzdělávacími institucemi. S řešením tohoto nesouladu pomáhá REWIRE Cybersecurity Profiler, tento nástroj pomáhá tvůrcům osnov navrhovat studijní programy tak aby splňovali požadavky pracovního trhu a jednotlivým uživatelům pomáhá doplňovat jejich kybernetické portfolio aby dosáhli jejich zvoleného profilu.

Tato práce se zaměřuje na vylepšení webové aplikace REWIRE Cybersecurity Profiler integrací algoritmu strojového učení na analýzu popisů kurzů, což výrazně zvyšuje časovou efektivitu a užitečnost aplikace. Kromě toho si práce kladla za cíl implementovat několik dalších vylepšení aplikace, jako je vizualizace uživatelských dat pomocí grafů, funkční URL, vylepšení uživatelského rozhraní a skript pro generování popisů kurzů pomocí ChatGPT.

Výstupem praktické části je především modul strojového učení pro automatickou analýzu popisů kurzů. Tento modul, stejně jako původní aplikace, se skládá ze tří částí: frontendové implementace v React.js, se kterou uživatel interaguje, Python algoritmu strojového učení, který provádí samotnou analýzu, a PHP backend skriptu, který zprostředkovává komunikaci mezi frontendovým uživatelským rozhraním a algoritmem strojového učení.

Frontend modulu je v aplikaci integrován do okna pro přidání nového kurzu ve formě tlačítka, které předává popis kurzu ve formě JSON PHP backendu ke zpracování. Backend nejprve zkontroluje existenci a nastavení oprávnění potřebných složek; pokud neexistují, jsou vytvořeny a nastavena uživatelská práva tak, aby je mohl algoritmus strojového učení používat. Z obdržných dat jsou odstraněny speciální znaky nekompatibilní s algoritmem strojového učení, který je nakonec spuštěn a je mu předán popis kurzu. Algoritmus strojového učení zpracuje předaný popis kurzu ve formě JSON a predikci skupin dovedností předá skrze backend frontendu, kde jsou zobrazeny.

Výstupem práce jsou také různá vylepšení aplikace, která zlepšují její funkčnost a uživatelský zážitek. Jedním z jednodušších vylepšení byla náhrada popisků z knihovny React-Bootstrap, které se nezobrazovaly správně, popisky z knihovny React-Tooltip. Dále byla implementována vizualizace uživatelských dat pomocí grafů v sekci pro správce. Byla vybrána data, jejichž informační hodnota je vizualizací navýšena a jsou zobrazena ve sloupcovém grafu, pokud jsou důležitá specifická čísla, nebo v koláčovém grafu, pokud jsou důležité podíly. Navíc byly implementovány funkční URL, které nejsou v React aplikacích standardně. Ty umožňují měnit obsah změnou URL a naopak se při změně obsahu URL automaticky aktualizuje.

Práce také obsahuje analýzu inženýrského studijního programu Informační bezpečnosti z Brněnské univerzity technologií. V této sekci je program popsán z hlediska obsahu, následně je popsán proces shromažďování potřebných dat a jejich zadání do aplikace. Výsledek analýzy byl manuálně opraven a ukázal, že použitý algoritmus nadkompenzuje, tedy vybírá spíše více skupin dovedností než méně, což je často způsobeno pouhými zmínkami technických témat.

V další části je již zmíněný studijní program využit k ukázce funkčnosti aplikace a její užitečnosti v navrhování studijních programů. V této části je kontrolována kompatibilita tohoto programu s vybraným ENISA profilem Chief Information Security Officer postupným volením vhodných předmětů pro naplnění požadavků zvoleného profilu a splnění podmínek studia. Konečná volba předmětů obsahuje všechny požadované skupiny dovedností kromě skupiny 'Policy Development', která není obsažena v žádném dostupném předmětu.

Na závěr je popsán skript pro generaci popisů kurzů pomocí ChatGPT. PHP skript požádá ChatGPT o vygenerování popisu kurzu pro uživatelem zvolenou skupinu dovedností a předá mu s touto skupinou asociované klíčové dovednosti a znalosti, které jsou při generaci brány v potaz. Vygenerovaný popis je následně vrácen uživateli. Tento skript je také využit k doplnění programu Informační bezpečnosti předmětem obsahujícím dovednostní skupinu 'Policy Development', čímž je dosaženo kompletního naplnění požadavků ENISA profilu.

HRACHOVINA, Jiří. *Modul webové aplikace pro tvorbu osnov kybernetické bezpečnosti*. Bachelor's Thesis. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Advised by M.Sc. Sara Ricci, Ph.D.

Author's Declaration

Author: Jiří Hrachovina
Author's ID: 240623
Paper type: Bachelor's Thesis
Academic year: 2023/24
Topic: Modul webové aplikace pro tvorbu osnov kybernetické bezpečnosti

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno

.....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to sincerely thank my thesis supervisor Dr. Sara Ricci, for her invaluable advice, unwavering support, and incredible patience. Her guidance helped me throughout the research and writing of this thesis. I would also like to thank Ing. Marek Sikora, whose expertise and technical assistance were crucial for the technical aspects of this thesis.

Contents

Introduction	12
Aim of the thesis	13
1 Background	14
1.1 REWIRE Cybersecurity Profiler	14
1.1.1 SPARTA Cybersecurity Curricula Designer	14
1.1.2 SPARTA Topics	15
1.1.3 NIST NICE Framework	15
1.1.4 ENISA ECSF Framework	16
1.1.5 Application Architecture Overview	17
1.2 Machine Learning Algorithm	18
1.2.1 What is Machine Learning	18
1.2.2 Machine Learning in This Project	18
1.2.3 Recurrent Neural Networks	19
1.2.4 Long Short-Term Memory	20
1.2.5 Bidirectional Encoder Representations from Transformers	20
1.2.6 Artificial Neural Networks	20
1.2.7 Natural Language Processing	21
1.3 Toolset Overview	22
1.3.1 Node.js	22
1.3.2 npm	22
1.3.3 Apache HTTP Server	22
1.3.4 pip	22
1.3.5 MySQL	23
1.3.6 phpMyAdmin	23
1.3.7 Git	23
1.4 Programming languages	23
1.4.1 JavaScript	23
1.4.2 PHP	24
1.4.3 Python	24
1.5 Essential libraries and Frameworks	25
1.5.1 JavaScript	25
1.5.2 Python	26

2	Thesis Results	29
2.1	Setting up the foundation	30
2.1.1	Understanding the Landscape	30
2.1.2	Configuring the Development Environment	31
2.2	Machine Learning Course Description Analysis	34
2.2.1	Objectives	34
2.2.2	Challenges in Manual Course Description Handling	34
2.2.3	Implementation	35
2.3	Tooltip Enhancement	38
2.4	Implementing User Statistics Graphs	39
2.5	Implementing User-Friendly URLs	40
2.6	Study Program Analysis	41
2.7	Cybersecurity Profiler Proof of Concept	41
2.8	Script for Course Description Generation	43
	Conclusion	44
	Bibliography	45
	List of appendices	50
A	User manual	51
B	Contents of the Electronic Attachment	52
C	JSON Files	53
C.1	MPC-IBE_all_available_courses	53
C.2	MPC-IBE_compulsory_only	53
C.3	MPC-IBE_compulsory&compulsory-optional	53
C.4	MPC-IBE_Final	53

List of Figures

1.1	European Cybersecurity Skills Framework	17
1.2	JSON representation of CSP user's data	24
2.1	CSP database as viewed in PhpMyAdmin	32
2.2	CSP - add course window	35
2.3	CSP - add course window after analysis	37
2.4	React-Bootstrap Tooltip Displayed Incorrectly	38
2.5	User Visit Duration in Web Application	39
2.6	Distribution of Visits Among Application Sections	40
2.7	Coverage of ENISA Profiles by the BUT FEKT Information Security Master's Program Compulsory and Compulsory-Optional courses . .	42
2.8	Coverage of ENISA Profiles by the BUT FEKT Information Security Master's Program Compulsory, Compulsory-Optional, and Optional Courses	43

Introduction

In recent years, the increasing interconnectivity of systems with the internet has highlighted the importance of cybersecurity. As more aspects of our digital lives are linked to online platforms, the risk of cyber threats has become more evident. This has led to a growing demand for cybersecurity specialists to protect digital environments from evolving security threats. Despite this increased demand, there is a global shortage of 3.9 million workers, with Europe facing a deficit of over 347,000 cybersecurity professionals[1]. A significant challenge has been ensuring that educational programs align with the rapidly changing demands of the job market.

To address the critical shortage and bridge the gap between market demand and educational response, initiatives like SPARTA[2] have been developed. SPARTA created the Curricula Designer[3], a tool that helps educators evaluate study program content to ensure it aligns with current job market roles.

Expanding on this initiative, REWIRE[4] introduced the Cybersecurity Profiler¹, adding valuable features such as training and certification provisions. The tool also includes the capability to upload or create databases of existing curricula, trainings, and certifications. This expansion provides users with a powerful tool to craft their cybersecurity profiles according to market requirements.

This thesis explores contemporary Graphical User Interface (GUI) web application development and aims to familiarize the reader with the REWIRE Cybersecurity Profiler. It includes an in-depth examination of global cybersecurity frameworks, such as the National Institute of Standards and Technology's National Initiative for Cybersecurity Education (NIST NICE) and the European Union Agency for Cybersecurity's European Cybersecurity Skills Framework (ENISA ECSF).

The primary objective of this thesis was the implementation of a machine learning algorithm module for description analysis. This module allows for the automatic analysis of course descriptions, significantly increasing the time efficiency and utility of the application. Additionally, the thesis implemented several other improvements, such as visualization of user data through charts, functional URLs that both indicate the user's location on the page and redirect upon change, improvements to UI elements, and a script for course description generation using ChatGPT. The thesis also includes a demonstration of the application to show its utility in curricula design.

¹<http://cyberability.rewireproject.eu/>

Aim of the thesis

The aim of this thesis is to investigate and implement advancements in the REWIRE Cybersecurity Profiler, an existing web application. Specifically, the focus is on the integration of a machine learning algorithm designed for the automated analysis of REWIRE skill groups. Additionally, the thesis aims to introduce enhancements to the frontend of the application to improve overall user interaction. Through this exploration, the objective is to contribute to the field of modern GUI web application development, showcasing proficiency in both backend and frontend concepts. The thesis aims to provide a comprehensive understanding of the implemented machine learning module, the associated challenges, and the solutions devised during the integration process.

1 Background

1.1 REWIRE Cybersecurity Profiler

The REWIRE Cybersecurity profiler (CSP)[5] is a dynamic web application designed for mapping existing curricula, training programs, and certifications to specific cybersecurity roles. It assists the user in identifying suitable courses, training programs, or certifications. Additionally, it facilitates the creation of study programs and the analysis of their suitability for the previously mentioned cybersecurity roles.

The Cybersecurity Profiler web application also serves as a database of existing cybersecurity study programs, training programs and certifications. It integrates the master and bachelor study programs collected by the SPARTA project and the professional trainings gathered by the CONCORDIA project. Further courses and certifications were collected by REWIRE partners. Furthermore, users have the capability to create and assess their own training, for example and subsequently upload it to the database.

The REWIRE Cybersecurity Profiler expands on the SPARTA Cybersecurity Curricula Designer (CCD) by not only adding training programs and certifications, but also enabling users to create or upload new ones, as well as databases containing curricula, training programs, and certifications. These features provide the user with a powerful tool for designing their cybersecurity profile that complies with ENISA requirements.

The Cybersecurity Profiler application will introduce a bilateral flow, empowering users in two significant ways. Firstly, users can autonomously develop their curricula, training and certifications to meet ENISA standards. Conversely, leveraging selected ENISA profiles, the tool will identify requisite skills and knowledge, aligning them with existing resources from attached databases. This dual capability will assist users in strategic decision-making, guiding them toward fulfilling the specified ENISA profile efficiently.

As the user selects ENISA profiles, the tool will identify required skills and knowledge and match them with existing curricula, trainings and certifications labelled with ECSF skills and knowledge they cover.[5]

1.1.1 SPARTA Cybersecurity Curricula Designer

As noted earlier, the CSP is an evolution of the SPARTA CCD. The initial version of CCD also operated as a web application, providing users the capability to generate new study programs or upload existing ones. It facilitated the analysis of program content, ensuring their alignment with work role requirements.

The application was developed by Brno University of Technology within the SPARTA project. For the analysis the SPARTA Cybersecurity Skills Framework is used. Work roles and competencies mirror the requirements of the NIST NICE framework.[5]

1.1.2 SPARTA Topics

The SPARTA topics encompass the most relevant areas in cybersecurity. This hand-picked list was developed to align with established curricula guidelines and the list of competencies outlined in the NIST NICE framework, which is recognized for its comprehensive cybersecurity classification. The primary objective of the SPARTA topics is to provide a straightforward means to categorize subjects. This facilitates the comparison of cybersecurity study programs, and enables the mapping of curricula to specific work roles.

The SPARTA topics are elaborately linked to NICE Technical and Operational Competencies. This establishes a direct connection to specific work roles, facilitating a smooth transition between academia and the job market. The NICE framework encompasses Technical, Operational, Professional, and Leadership Competencies, with the mapping in SPARTA focusing exclusively on Technical and Operational aspects. This deliberate choice stems from the recognition that Professional and Leadership Competency groups, related more to teaching methods and additional modules, fall beyond the scope of the current SPARTA topics.

The SPARTA topics are categorized into three groups: Fundamental, Cyber Security, and New Trends. Fundamental subjects act as prerequisites for subsequent studies and may not be directly linked to the Framework, while Cyber Security topics have a direct association. The New Trends category includes topics that, being new, are not yet covered in the existing framework.[5]

1.1.3 NIST NICE Framework

The NIST Workforce Framework for Cybersecurity, commonly known as the NICE Framework, offers a structured framework to define the tasks, knowledge, and skills essential for individuals and teams engaged in cybersecurity work. By providing these foundational building blocks, the NICE Framework empowers organizations to cultivate their workforce for effective cybersecurity practices. Simultaneously, it aids learners in exploring the realm of cybersecurity work, guiding them towards relevant learning activities to enhance their knowledge and skills.

The NICE Framework offers a way to describe cybersecurity work by utilizing Task statements that outline the work to be performed, Knowledge, and Skill

statements that specify what learners—such as students, job seekers, and employees—need to know and be capable of doing to accomplish that work. These statements serve as the foundation for constructing Competency Areas and Work Roles, which prove valuable for both organizations and individuals seeking a comprehensive understanding of cybersecurity roles and responsibilities.[6]

1.1.4 ENISA ECSF Framework

The ENISA European Cybersecurity Skills Framework (ECSF) is a practical tool for identifying and defining the tasks, competences, skills, and knowledge of cybersecurity professionals in Europe. It serves as the European Union’s reference for defining and assessing these skills, as outlined in the Cybersecurity Skills Academy.

The ECSF aims to standardize the understanding of relevant roles, competencies, skills and knowledge, facilitating the recognition of cybersecurity skills and the development of related training programs. It categorizes all cybersecurity related roles into 12 profiles, each analyzed in terms of responsibilities, skills, synergies, and interdependencies.

With just 12 profiles, the ECSF is easily comprehensible. A downside of this is its insufficient precision, particularly concerning potential career paths, prerequisites for crafting more detailed training programs that will support not just initial training, but also continuous education and retraining.[5]

Figure 1.1 displays the 12 ENISA profiles. Each profile in ECSF is described through a table containing:

- *Alternative Title(s)*: other possible titles for the same profile,
- *Summary statement*: the purpose of the profile,
- *Mission*: the rationale of the profile,
- *Deliverable(s)*: the relevance with the perspective from a non-Cybersecurity/Information and Communication Technologies point of view,
- *Main task(s)*: the tasks performed by the profile,
- *Key skill(s)*: list of abilities for the profile,
- *Key knowledge*: list of essential knowledge for the profile,
- *e-Competences*: list of e-Competences Framework competencies covered by the profile.



Fig. 1.1: European Cybersecurity Skills Framework

1.1.5 Application Architecture Overview

The web application operates under a well-defined architectural framework, comprising of three main parts: a React JS frontend, a PHP backend and a MariaDB database.

React JS frontend

The frontend layer, responsible for delivering a dynamic and interactive user interface (UI), is built using React JS. React enables efficient handling of client-side operations, creating engaging interfaces. In development Node.js is utilized to manage and build the application. In production the application is hosted on a web server optimized for serving static files.

PHP Backend

Serving as the logic and processing hub, the PHP backend processes client request, executes business logic, and manages communication between the frontend and the database. The backend component is hosted on an Apache server, which is well equipped to run PHP scripts.

MariaDB Database

At the core of the system lies the MariaDB database, serving as the repository for data storage and retrieval. It stores data about certifications, courses, study programs, training programs, users, and other related information.

1.2 Machine Learning Algorithm

1.2.1 What is Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) and computer science focused on the use of data and algorithms that enable AI to mimic human learning, gradually improving its accuracy.[7]

In recent years ML has found use in a variety of fields, including natural language processing, speech recognition, computer vision, email filtering, agriculture, and medicine.[8][9] While they have been around for some time, what truly brought ML into the public consciousness was the introduction of large language models (LLM) like ChatGPT.[10]

1.2.2 Machine Learning in This Project

This project implements a repurposed version of the machine learning model previously used in the REWIRE Jobs ads analyzer. It uses a Recurrent Neural Network (RNN) model to automate the analysis of job advertisements. The model uses a pre-built word-piece tokenizer for Bidirectional Transformer (BERT). The concatenated output of the tokenizer is fed to an Artificial Neural Network (ANN) to classify security skills. Utilizing Long Short-Term Memory (LSTM) with a hidden layer size of 128 and ANN with a hidden layer of 256.

The training involved assessing the model's capacity to predict the necessary security skills utilizing a test dataset. The ground truth labels were manually assigned, and the labeled dataset was randomly split into 90% for training and 10% for testing. The LSTM model was only trained on the training dataset to avoid it from developing a bias by learning from the test dataset.[11]

The repurposed machine learning model is used to analyze course descriptions and identify the REWIRE skill groups present within them. However, as the model is not optimized for this specific use, it does not always produce accurate results. Improving its performance has been a continuous effort, and the results have been steadily improving. A major step in the right direction has been the introduction of a dictionary of words associated with each skill group. This change had a considerable effect on the model's precision.

1.2.3 Recurrent Neural Networks

A recurrent neural network is a type of artificial neural network which uses sequential data or time series data. They are commonly used for language translation, neural language processing (NLP), speech recognition and image captioning. You can find them in applications such as Siri or Google translate.

RNNs utilize training data for learning and are characterized by their inherent "memory". This memory attribute allows them to incorporate information from previous inputs, influencing both the current input and the resulting output, unlike traditional deep neural networks, which assume that inputs and outputs are independent of each other.

Using the idiom "feeling under the weather" as an example, we can explain how RNNs operate in processing sequential information. In this example, the idiom's significance relies on the specific order of its words. Similarly, RNNs take into account the order of words in a sentence. They utilize information from preceding words to predict the subsequent one, effectively capturing contextual relationships within the sequence.

To illustrate how RNNs function, consider the idiom "feeling under the weather," commonly used to describe someone who is ill. The idiom's meaning depends on the specific order of its words. Similarly, RNNs must account for the position of each word in a sequence to make sense of the data and predict the next word. RNNs differ from feedforward networks because they share parameters across each layer, maintaining consistent weight parameters. These shared weights are adjusted through backpropagation and gradient descent, enabling the network to learn and improve.

RNNs leverage the backpropagation through time (BPTT) algorithm to determine gradients, which is slightly different from traditional backpropagation as it is specifically tailored to sequence data. The foundational principles of BPTT are identical to those of traditional backpropagation, involving the model's self-training through error calculations from the output layer to the input layer. These calculations enable the adjustment and fitting of the model's parameters. BPTT differs from the traditional approach by summing errors at each time step. This is because feedforward networks, lacking shared parameters across layers, do not need to sum errors.

Because of this process, RNNs generally encounter two problems known as exploding and vanishing gradients. These issues are defined by the size of the gradient, which is the slope of the loss function along the error curve. In the case of vanishing gradients, the gradient becomes extremely small and continues diminishing, causing the weight parameters to approach insignificance, effectively halting the learning

process. Exploding gradients, on the other hand, arise when the gradient becomes excessively large, leading to an unstable model where weights grow uncontrollably and may eventually be represented as NaN, which stands for "not a number" and in the context programming and numerical computations, it represents an undefined or unrepresentable value. One approach to mitigate these issues is to reduce the number of hidden layers in the neural network, simplifying the RNN model.[12]

1.2.4 Long Short-Term Memory

Long short-term memory is a popular type of RNN architecture designed to address the vanishing gradient problem, which can occur in traditional RNNs.[12] LSTMs were introduced by Sepp Hochreiter and Jurgen Schmidhuber.[13]

The key advantage of LSTMs is their ability to capture long-term dependencies in sequential data. Unlike traditional RNNs, LSTMs have a more complex structure that includes memory cells and a set of gates (input gate, forget gate and output gate). These gates regulate the flow of information into and out of the memory cells, allowing LSTMs to selectively retain or discard information over extended sequences.[12]

1.2.5 Bidirectional Encoder Representations from Transformers

BERT is a neural network architecture employed in Natural Language Processing tasks, encompassing activities like language modeling, machine translation, sentiment analysis, and others.

The primary feature of BERT lies in its capacity to analyze input sequences bidirectionally, from both left to right and right to left. This dual processing capability allows it to capture contextual information from both past and future input tokens, proving crucial for a variety of NLP tasks.

Unlike other models, BERT does not consist of two separate Transformer models. Instead, it uses a single Transformer architecture where each token attends to every other token in the input sequence bidirectionally during the training phase. This enables BERT to effectively capture intricate dependencies among input tokens, facilitating accurate predictions in various NLP applications.[14]

1.2.6 Artificial Neural Networks

Artificial Neural Networks also known as just Neural Networks or Simulated Neural Networks, constitute a subset of machine learning and serve as a foundational element in deep learning algorithms. Named and designed in emulation of the human brain, they replicate the communication pattern of biological neurons.

ANNs consist of layers of nodes, encompassing an input layer, one or more hidden layers and an output layer. In this structure each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of an individual node surpasses the specified threshold value, the node is activated and passes data to the next layer of the network. Otherwise, no data is passed along.[15]

1.2.7 Natural Language Processing

Natural language processing is a branch of artificial intelligence, concerned with enabling computers to understand and process text and speech in much the same way humans do. NLP combines computational linguistics, which tries to understand language from a computational perspective, with statistical, machine learning and deep learning models. These technologies enable computers to understand human language in both written and spoken form and to understand its full meaning, even the speaker's or writer's intent and sentiment. NLP is at the core of programs that translate languages, respond to spoken commands and summarize large quantities of text rapidly.

Processing human language presents an exceptionally challenging task due to its inherent ambiguity. Sarcasm, metaphors, homonyms, grammar and its exceptions, variations of sentence structure. These are just a few examples of quirks in human language, that take humans years to learn, but NLP driven programs must be capable of understanding them from the start, if they're to be useful.[16]

NLP employs various methods to make sense of the linguistic input its 'ingesting':

- **Speech recognition**, also called speech-to-text is the task of converting spoken word into text data. It is required in any application that can follow voice commands.
- **Part of speech tagging**, also called grammatical tagging, is the process of marking up a word in text as corresponding to a particular part of speech, based on its definition and context.
- **Word sense disambiguation** involves selecting the appropriate meaning of a word with multiple interpretations through semantic analysis. This process determines the word that makes the most sense in a given context.
- **Named entity recognition** discerns and categorizes words or phrases as meaningful entities. For example 'Fred' as a name or 'Brno' as a place.
- **Co-reference resolution** is the task of identifying when two words are referring to the same entity. The most common case is determining the person or object to which a pronoun refers, but it can also involve identifying a metaphor or an idiom.

- **Sentiment analysis** analyzes the text to determine subjective qualities like: attitudes, emotion, confusion, sarcasm.
- **Natural language generation** is the task of putting structured information into human language.[16]

1.3 Toolset Overview

1.3.1 Node.js

Node.js is an open-source, cross-platform JavaScript (JS) runtime environment that allows developers to execute JS code server-side. It enables the development of scalable and high-performance web applications by using JS as both the server-side scripting language and the language for building the client-side logic.[17]

1.3.2 npm

npm (node run manager) is the largest software registry in the world, and the default package manager for Node.js, with which it is bundled. It provides a platform for developers to share and borrow code packages, making it an essential tool for JS development. The npm command line interface allows developers to install, update, and manage packages directly from their interface.[18]

1.3.3 Apache HTTP Server

Apache HTTP Server, often simply called "Apache" is an open-source web server widely used for serving web content on the internet. It is designed to handle HTTP requests and responses, manage web traffic, and ensure that web applications are available to the public.[19]

1.3.4 pip

pip (python installer for python) is the standard package management tool for Python. It is used to install and manage additional libraries and dependencies that are not part of the standard Python library. pip simplifies the process of managing Python packages by automating the installation, removal, and upgrading processes. It connects to the Python package index (PyPI) and other repositories to find and install packages.[20]

1.3.5 MySQL

MySQL is an open-source relational database management system (RDBMS) that enables users to store, manage, and retrieve data efficiently.[21] It is commonly used in web applications as part of the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python).[22]

1.3.6 phpMyAdmin

phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB databases. Users can create and delete databases, run SQL queries, manage user permissions, and import/export data in various formats, all through an elegant user interface.[23]

1.3.7 Git

Git is a free and open-source distributed version control system. It tracks changes in source code by maintaining a local copy of the entire repository, including its history and version tracking. Git is often used in collaborative projects, as it allows multiple developers to work on the same project simultaneously, integrating their changes smoothly.[24]

1.4 Programming languages

1.4.1 JavaScript

JavaScript is a high-level, dynamic programming language widely used in web development. It allows developers to add interactivity and dynamic behaviour to websites, making it an essential technology for creating modern, interactive web applications. Along with HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) it creates the core of the world wide web. It allows for client-side functionality such as form validation, interactive maps, and real-time content updates.[25]

JSON

JSON (JavaScript Object Notation) is a lightweight text-based format for representing structured data, that is easy for humans to read and write, and easy for machines to parse and generate. It is often used for transmitting data in web applications, which is its purpose in the CSP application. The data is structured in key-value pairs as can be seen in figure 1.2, arrays, and objects, enabling representation of complex data hierarchies.[26]

```
{
  "id": 1,
  "ip": "***.***.***.***",
  "organization": "Brno University of Technology",
  "city": "Brno",
  "country": "Czechia",
  "device": "Desktop",
  "browser": {
    "name": "Chrome",
    "version": "122.0.0.0"
  },
  "os": {
    "name": "Windows",
    "version": "10"
  },
  "sections": [
    "home"
  ],
  "user_id": 0,
  "start_time": "2024-03-01 15:09:16",
  "end_time": "2024-03-01 15:09:30"
}
```

Fig. 1.2: JSON representation of CSP user's data

1.4.2 PHP

PHP, which stands for "PHP:Hypertext Preprocessor", is a popular general-purpose scripting language that is especially well-suited to server-side web development. It is embedded within HTML code and executed on the server, generating HTML that is sent to the client, ensuring that the client never gets to see the underlying code. PHP is recognized for its simplicity, while also offering advanced features and security.[27] For these reasons it remains the most widely used server-side programming language.[28]

1.4.3 Python

Python is a high-level, interpreted programming language known for its readability. It supports procedural, object-oriented, and functional programming. Python emphasizes readability by using indentation instead of parentheses to define code blocks. Since its release, it has become one of the most widely used programming

languages, finding use in web development, artificial intelligence, data science, and many other fields.[29]

1.5 Essential libraries and Frameworks

1.5.1 JavaScript

React JS

React JS, also known as React, is an open-source JavaScript framework-like library used for building user interfaces, primarily maintained by Meta (formerly Facebook). React allows developers to create reusable UI components, from which complex UIs can be built. These components are rendered using a virtual DOM (Document Object Model).[30]

The DOM is a programming interface for web documents. It represents the document as nodes and objects, so that programming languages can interact with the page. As an object-oriented representation it can be modified by JS and other scripting languages.[31]

Axios

Axios is a promise-based HTTP client designed to make HTTP requests from the browser and Node.js environments. It simplifies the process of sending asynchronous HTTP requests to interact with web servers, retrieve data, and perform other operations that involve communication with a server.[32]

React-Bootstrap

Bootstrap is a popular open-source frontend framework. It includes a collection of pre-designed and responsive HTML, CSS, and JS components, such as buttons, forms, navigation bars, modals and more. Bootstrap simplifies the process of web development by offering consistent and visually appealing set of components that can be easily customized.[33][34]

React-Bootstrap is a library that provides Bootstrap components as React components, allowing developers to use the framework within React applications.[35][36]

ApexCharts

ApexCharts is an open-source JavaScript library designed for creating flexible and responsive charts. It offers a wide range of chart types, that are highly customizable

and easy to configure.[37] In this case a React compatible version of ApexCharts is used.

React Router

React Router is a React library that enables dynamic routing in web applications. It allows developers to create single-page applications with navigation that feels like a traditional multi-page application. React Router manages the routing by updating the user interface in response to URL changes and by modifying the URL based on interface changes.[38]

1.5.2 Python

Beautifulsoup4

Beautifulsoup4 is a Python library used for web scraping, essentially pulling data out of HTML and XML files. It allows for easy navigation, searching, and modification of document content by creating a parse tree for parsed pages.[39]

Chardet

Chardet is a Python library used for character encoding detection. Its primary purpose is to determine the character encoding of a given byte stream or file-like object. Character encoding is essential for correctly interpreting text data, especially when working with files or data that may use different encoding schemes.[40]

Emoji

Emoji is a Python library that allows developers to use and print emojis. It offers features such as converting between emoji shortcodes and Unicode characters, extracting emojis from text, and more.[41]

Matplotlib

Matplotlib is a popular data visualization library for Python, used for creating static, animated, and interactive visualizations. It provides a wide range of tools for constructing plots, charts, and other visual representations of data.[42]

NumPy

NumPy is a powerful numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It is considered a fundamental library

for scientific computing in Python and serves as the foundation for many other libraries and tools in the data science and machine learning ecosystem.[43]

Pandas

Pandas is a open-source data manipulation and analysis library for Python. It provides data structures for efficiently storing and manipulating large datasets, along with tools for cleaning, aggregating, and analyzing the data. Pandas is considered a fundamental library in the field of data science and is widely used for data preparation and exploratory data analysis.[44][45]

Scikit learn

Scikit_learn is an open-source machine learning library for Python that provides simple and efficient tools for data analysis and modeling. It is designed to be accessible to non-experts while offering a rich set of functionalities for experienced users. Scikit-learn is often used for tasks such as classification, regression, clustering, dimensionality reduction, and more.[46]

PyTorch

PyTorch, also known as "Torch" is an open-source machine learning framework for Python. It is used for building deep learning models, a type of ML typically used in image recognition and language processing. Torch supports complex model building.[47][48][49]

TorchText

TorchText, which is part of the PyTorch project, is a Python library that provides data processing utilities and datasets for natural language processing tasks. It handles text data, language modeling, and machine translation and supports operations such as tokenization, vocabulary management, and data pipeline building.[?]

tqdm

tqdm is a Python library used to display progress bars. It helps visualize the progress of loops of iterable objects, making it easier to monitor the execution of tasks.[52]

Transformers

Transformers is is an open-source library developed by Hugging Face. It provides easy access to a wide range of pre-trained transformer models for natural language

processing tasks. The library facilitates the use of state-of-the-art models like BERT, GPT, RoBERTa.[53]

Openpyxl

Openpyxl is a Python library for reading and writing Excel files. It allows users to create, modify, and extract data from Excel spreadsheets. The library supports various Excel features, including formatting, charts, formulas, and more.[54]

2 Thesis Results

This chapter presents the results of the enhancements made to the REWIRE Cybersecurity Profiler. It begins with my analysis of the application's structure and the three main components. Following the codebase analysis, I set up the development environment, installing Node, the LAMP stack for the CSP application, and a virtual environment for the Python machine learning algorithm. I also installed the necessary dependencies and documented the challenges encountered.

After completing the necessary setup, I moved on to implementing the machine learning algorithm for course description analysis. In the thesis, I first explain the rationale behind this implementation and how it is designed to help users save time and achieve better results.

To initiate the implementation, I modified the course addition window in the application to include an 'Analyze' button. This button triggers the ML analysis by sending the content of the description field to the backend using Axios.

In the PHP backend, I created a new script that processes the description, formats it into JSON, and sends it to the ML algorithm. The ML algorithm analyzes the input and returns a prediction of the relevant REWIRE skill groups. This prediction is then passed back through the PHP script to the CSP application, where it is displayed to the user.

Additionally, I made several other improvements. I fixed React-Bootstrap tooltips by replacing them with React-Tooltip. I implemented user statistics visualization using ApexCharts and enabled functional URLs with React Router, allowing URL updates as users navigate.

To demonstrate the application's functionality and utility, I showcase how it can be used to evaluate the coverage of ENISA profiles by study programs. Using the Information Security master's program at my university as an example, I focus on the Chief Information Security Officer as the desired ENISA profile. I demonstrate how the program's coverage of the profile can be assessed by incrementally adding compulsory, compulsory-optional, and optional courses. By utilizing the live feedback, I determine which courses to add based on the skill groups they include, thereby optimizing the program's alignment with the ENISA profile.

The improvements were implemented live and can be viewed in the production version available on the internet¹.

¹<http://cyberability.rewireproject.eu/>

2.1 Setting up the foundation

As this project expands upon an existing web application, a critical first step was establishing a strong foundation. This section explains how I set up the existing codebase and describes the steps taken to smoothly integrate the old system with the new features.

2.1.1 Understanding the Landscape

Codebase Analysis

Before making any changes or additions, it was essential to thoroughly understand the existing codebase. This involved examining the web application's architecture, dependencies, and overall structure. By gaining detailed insights into the current system, I was able to lay the groundwork for a well-informed integration process.

The system's codebase is organized into three main components: JavaScript, PHP, and the database. This modular architecture was intentionally designed to enhance maintainability, scalability, and flexibility. The machine learning algorithm is separate from this structure. This separation is reflected in the application's file structure, with each part contained in its designated space.

The JavaScript component handles the frontend logic and user interface elements. It functions as a standalone module responsible for user interactions and dynamic content presentation. This modularization allows for independent development and updates, promoting a more agile and responsive user experience.

The PHP component handles the backend logic, managing server-side processing and communication between the frontend and the database. By isolating server-side functionalities, this modular approach enhances code maintainability and simplifies debugging. Each PHP module is dedicated to specific tasks, contributing to a clear and organized backend structure. The backend is hosted on an Apache server.

The database component stores and manages the application's data using MySQL. Its modular design allows for the efficient organization of data tables and relationships. This separation ensures that the database is not only scalable but also adaptable to future changes in data requirements. The database is hosted on a dedicated MySQL server.

While each component operates independently, the system's strength lies in the seamless interaction between these modules. JavaScript communicates with PHP to facilitate dynamic content updates, and PHP, in turn, interacts with the MySQL database for data retrieval and storage. This interconnected yet modular structure fosters a robust and cohesive system.

2.1.2 Configuring the Development Environment

To start working on the application, I first set up my development environment on my Linux machine. This initial setup was crucial for enabling further development.

Node.js and npm Configuration

To set up the development environment, I started by installing Node.js and npm. After researching to find compatible versions for the project, I installed them and confirmed they were working properly.

The development environment uses Node.js to host the React JS frontend. Managing the Node.js version is important to ensure it works well with React and its packages. I chose Node.js version 14.21.3 for this project because it is compatible with the versions of React modules used.

The management of JavaScript packages for the React frontend is facilitated by npm, which is automatically installed in conjunction with Node.js. For this project, I opted for npm version 6.14.18.

Apache server, MariaDB and PHP configuration

Next, I installed the components of the LAMP stack, which include Apache, MariaDB (a MySQL-compatible database), and PHP.

To set up Apache, I installed the latest stable UNIX version available at the time, which was 2.4.58. I chose the newest version as there were no dependencies requiring a legacy version. To ensure proper functionality of the Apache server, I edited firewall settings to allow HTTP and HTTPS traffic to and from the server. This adjustment was necessary to facilitate communication with the server over these protocols. With the Apache server successfully installed and operational, I proceeded to install my chosen database management system, MariaDB, specifically one of the newer short-term stable versions, 11.1.3. Following the standard installation procedures I created a new user specifically for the backend to establish communication with the database. This user was configured to facilitate secure and controlled access to the database for the application's backend functionality.

Afterwards, I installed PHP, opting for the latest version 8.2.12. Following installation, I configured the default location for server files to the application server file directory.

In the final steps, I installed the newest version of PhpMyAdmin to facilitate easy administration of the database. After ensuring correct function between Apache and PhpMyAdmin, I imported the application’s database through the phpMyAdmin interface. This step allowed me to manage and interact with the database efficiently using the user-friendly interface provided by PhpMyAdmin, which can be seen in Figure 2.1.

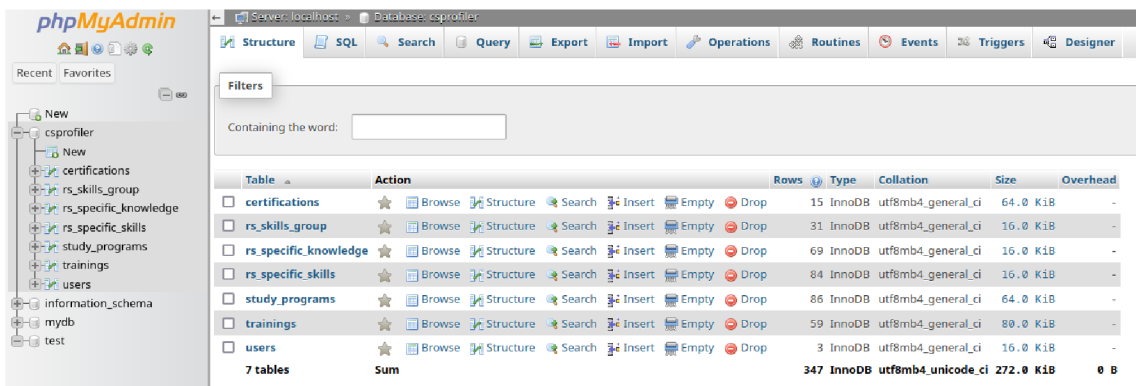


Fig. 2.1: CSP database as viewed in PhpMyAdmin

Python Configuration

Python is crucial for this project, supporting the functionality of the machine learning algorithm. I chose version 3.8.18 for its compatibility with the necessary libraries.

In the development environment, the Python machine learning algorithm operates within a dedicated Python virtual environment. This self-contained directory encapsulates a specific Python interpreter along with its associated libraries and dependencies. By using an isolated environment for the Python project, it is ensured that the dependencies and versions of packages used in one project do not interfere with those used in another.

In the production environment, at the time of writing, a virtual environment was not used, occasionally leading to severe issues. The absence of a virtual environment can result in compatibility conflicts, version mismatches, and other complications that may significantly impact the reliability and stability of the deployed Python machine learning algorithm.

Dependency Management

Both the preexisting Cyber Security Profiler and the machine learning algorithm rely on specific dependencies. In React applications, dependencies and their versions are conventionally documented in the 'package.json' file, while Python projects use the 'requirements.txt' file for the same purpose. After configuring the React and Python environments, I systematically installed the required dependencies for each project. In both cases, the required dependencies had to be installed one by one do to collisions.

Setup Finalization

With all the necessary components configured, and dependencies installed, I reached the point where I could successfully run the Cybersecurity Profiler application. This marked the beginning of the development phase, allowing me to initiate planned changes and additions to the application. The groundwork laid in the setup phase now positioned me to actively engage in the enhancement and expansion of the application's functionality.

Version Control Systems

Version control is a crucial aspect of maintaining an organized development environment. In this project, I utilized the Git version control system to track changes in the codebase.

Challenges and Solutions

Throughout the configuration process various, unexpected challenges emerged. Common pitfalls, such as compatibility issues and configuration errors, appeared during the setup of both the CSP application and the ML algorithm. These issues were systematically addressed through a combination of online resources, trial and error, and collaboration with project REWIRE members.

While installing the Python dependencies, I encountered a challenge due to the outdated versions of libraries. Many of them were deprecated, meaning that developers no longer recommend their use. As a result, it became difficult to install these libraries as they were not widely available on Python repositories. To overcome this obstacle, I diligently searched through available repositories until I located the desired libraries with the correct versions. Thus solving the problem.

Later on, it turned out that the extensive search was unnecessary, as the Python ML algorithm functioned well with the latest versions of the required libraries. This occurrence is quite uncommon in Python development, as updating to newer library

versions often carries the risk of breaking code due to changes introduced between versions.

2.2 Machine Learning Course Description Analysis

Adding numerous courses through the application can be a time-consuming endeavour, highlighting the need for efficiency and accuracy in the analysis of course descriptions. In response to this problem, we introduce a transformative feature: automatic course description analysis powered by machine learning. Manually assessing and comprehending the nuances of each description can be a daunting and time consuming-task. This new automated analysis feature aims to significantly streamline this process, making it faster and much easier.

2.2.1 Objectives

The core objective of this feature is to automate the analysis of course descriptions using machine learning. By leveraging advanced algorithms, we seek to minimize manual effort and expedite the evaluation process. This not only enhances efficiency but also introduces a level of consistency and standardization in the analysis framework, mitigating subjectivity issues associated with manual evaluations.

A key benefit of this automated analysis is the provision of timely and useful insights. Allowing educators and administrators, to access fast evaluations, we aim to facilitate quicker and more informative decision making in the context of educational planning. The feature is designed to present information clearly, making it easier for users to understand and use the results.

2.2.2 Challenges in Manual Course Description Handling

This section explains why there is a need for an automated process of description analysis.

Time Constraints

In the manual process of handling course descriptions, administrators and educators are confronted with a substantial time burden. Manually evaluating numerous course descriptions can be a time-consuming task, often taking resources away from more strategic aspects of educational planning.

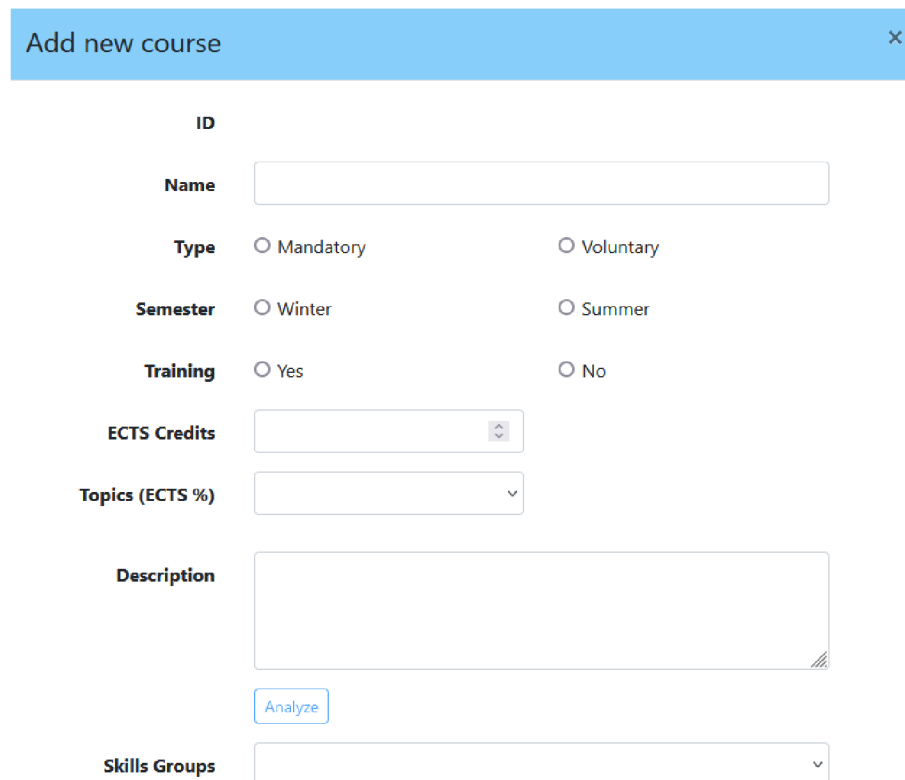
Subjectivity issues

Subjectivity is a challenge in the manual evaluation of course descriptions. Different people can have varying interpretations and assessment criteria, leading to inconsistencies. Administrators and educators may unintentionally bring their own perspectives and biases to the manual evaluation, leading to differences in understanding and categorization of course content. These issues can also affect the identification of key skills and the assessment of course relevance.

2.2.3 Implementation

User Interface

The 'Add New Course' window serves as the entry point for users adding new courses. It guides the user through entering essential course information, with each input field clearly labeled for easy understanding.



Add new course ×

ID

Name

Type Mandatory Voluntary

Semester Winter Summer

Training Yes No

ECTS Credits

Topics (ECTS %)

Description

Skills Groups

Fig. 2.2: CSP - add course window

As we navigate through the interface shown in Figure 2.2, we focus on the input fields for essential course information. Special attention is given to the "Description" field, which captures the course details. Notably, the 'Analyze' button below the

'Description' field is a key element, initiating the machine learning analysis of the provided course description.

Equally important is the 'Skills Groups' field. While it can be used for manually selecting REWIRE skills groups, it more importantly displays the prediction created by the machine learning algorithm.

When the 'Analyze' button is pressed and the "Description" field contains text, the content is sent to the application backend using the Axios post function in the form of a JSON file. In the backend, the data undergoes necessary preprocessing before its analysis by the ML algorithm.

Backend Handling

Upon receiving the data, several crucial tasks must be completed before it is ready for analysis by the machine learning algorithm. First, the script verifies the existence of the necessary input and output folders. If these folders are present, the script clears any previous inputs and outputs to ensure a clean slate for upcoming operations. If the folders are missing, the script creates them. Additionally, it modifies user permissions to ensure the server has access to the data for processing.

Next, a new folder named with the current date and time is created within the input and output directory. This folder serves as a designated repository for the machine learning algorithm's input. The received JSON data undergoes processing before being saved in this folder. Specifically, characters incompatible with the machine learning algorithm are removed to ensure proper function of the algorithm.

Following these preparatory steps, the machine learning script is initiated by a command executed in the PHP file. This command triggers the execution of the machine learning algorithm, which analyses the prepared input data.

Machine Learning Analysis

The machine learning algorithm processes JSON data from the input directory, generating a prediction of relevant REWIRE skill groups. This prediction is then stored in the output directory in a folder named with the date and time. The output is formatted as a JSON file, which needs to be parsed before utilization. The file is edited to convert it into an array of strings. Using a lookup table, these strings are replaced with corresponding IDs. These IDs are then passed back to the frontend in JSON format.

After receiving the machine learning algorithm's predictions for REWIRE skill groups in the form of IDs from the application's backend, the associated names are selected and displayed in their respective fields, as illustrated in Figure 2.3.

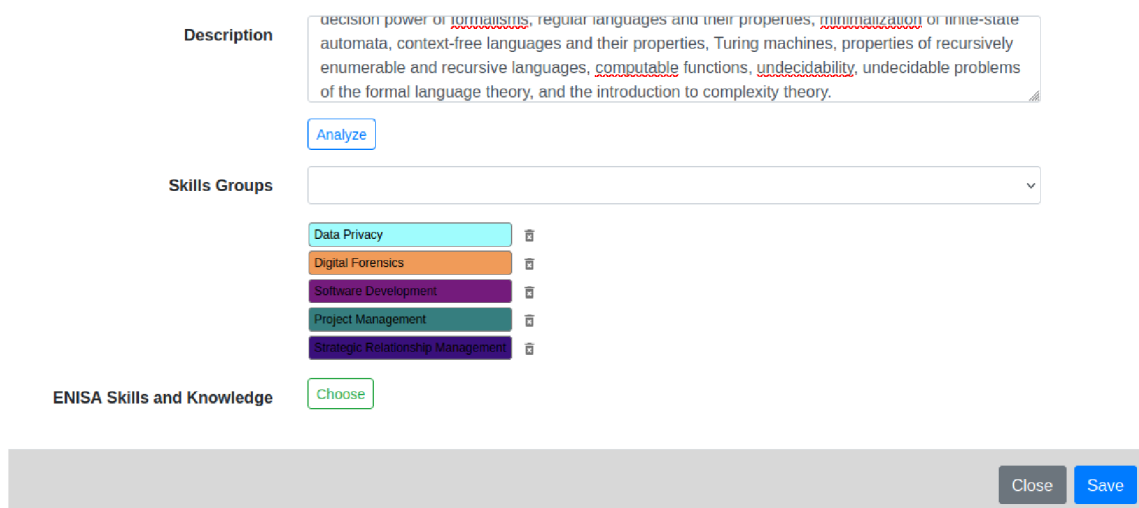


Fig. 2.3: CSP - add course window after analysis

Issues During Development

During the integration of the machine learning algorithm, I encountered several major issues that temporarily halted development. However, these challenges were successfully addressed through a combination of online resources and collaborative problem solving with project REWIRE members.

One of the challenges I faced was the erratic behavior of the script responsible for initiating the machine learning algorithm. After multiple attempts and a thorough analysis of error logs, I realized that this issue was not a straightforward, single-layer problem. The complexity of the issue required a more in-depth investigation and troubleshooting to identify and address the root causes.

Firstly, I encountered an issue with the algorithm's ability to open the necessary files. While the algorithm worked after adjusting permissions when run standalone, the problem arose when executed as part of a terminal script in PHP. This script ran under the Apache server's user context, named 'httpd' in this case. To resolve this, I granted the 'httpd' user the necessary permissions, which solved the file access problem within the PHP execution environment.

Another challenge arose with executing the script within the virtual environment. The problem was that when a virtual environment is initiated, it is limited to that terminal session. Since the script was executed in a new terminal session each time, this caused complications, as the script was being executed outside of the virtual environment. To fix this, I modified the process to initiate the virtual environment in the same command that executed the script, ensuring consistent and correct execution within the designated environment.

The final hurdle was the script abruptly shutting down due to an error. The root

cause was the machine learning algorithm’s inability to locate the required libraries, even though they were installed in the virtual environment. I discovered that the issue stemmed from using the default Python interpreter instead of the one specific to the virtual environment. To fix this, I ensured that the script was executed with the Python interpreter within the virtual environment, preventing library-related issues and allowing for successful execution.

By fixing these issues, I managed to get the machine learning algorithm to run smoothly. My efforts, along with valuable outside assistance in troubleshooting and problem solving, led to the successful execution of the algorithm, marking a significant milestone in the development process.

2.3 Tooltip Enhancement

In our efforts to improve the application, another minor, but not insignificant improvement was replacing the React-Bootstrap tooltips with those from React-Tooltip. The decision to transition was prompted by the unreliability of the original React-Bootstrap tooltips, which sometimes flickered or flashed across the screen and occasionally displayed incorrectly. This change aimed not only to address these reliability issues, but also to provide a better user experience.



Fig. 2.4: React-Bootstrap Tooltip Displayed Incorrectly

In Figure 2.4 the tooltip is wrongly displayed in the top-left corner of the screen instead of above the gray-colored square as intended. Despite efforts to identify the root cause of this issue, no solution was found. Opting for a pragmatic approach,

we decided to replace the tooltips entirely. This straightforward solution resolved the misplacement problem. Unfortunately, because each tooltip was implemented differently, I had to replace them one by one.

2.4 Implementing User Statistics Graphs

An already implemented feature of the application is the collection of user statistics, such as which parts of the application were utilized and how long visitors remained on the web application. Since manually examining this data is quite time-consuming, we decided to implement graphical visualization. We identified which data would be more comprehensible when visualized, specifically choosing the device type (desktop or mobile), operating system, country, visits by registered users (i.e., the number of visits by registered users), and the aforementioned length of visit and parts visited.

My task was to implement visualization using a version of the existing JavaScript library, ApexCharts, which has been adapted for use in React applications. In this implementation, two types of charts are used: pie charts and bar charts, as illustrated in Figures 2.5 and 2.6 . Pie charts, which focus on showcasing ratios, are used for data where specific numbers are less important. Conversely, bar charts are utilized when specific numbers are crucial.

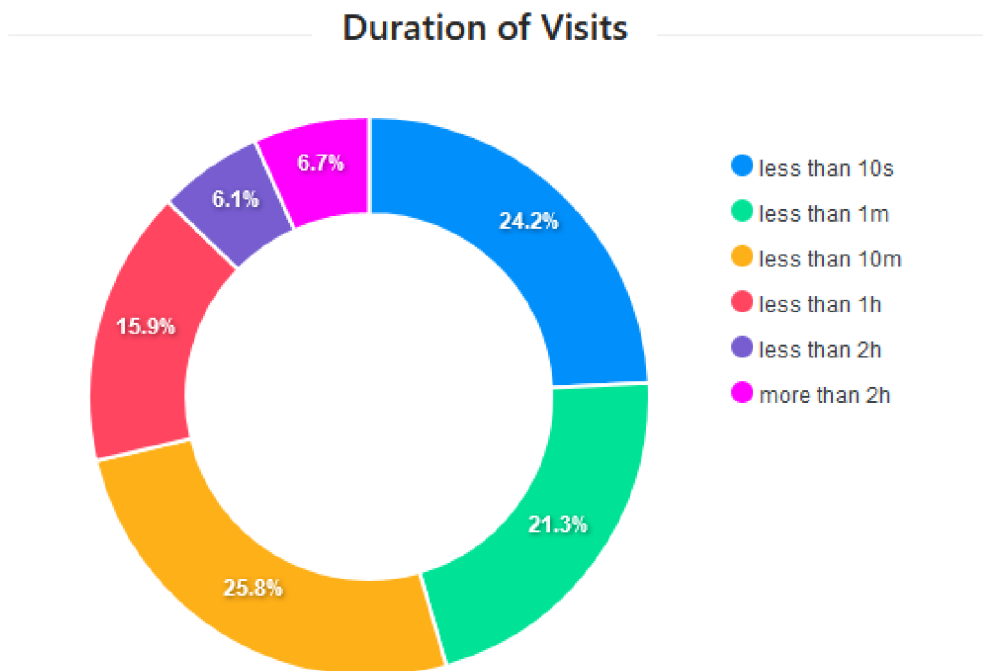


Fig. 2.5: User Visit Duration in Web Application

Visited Sections

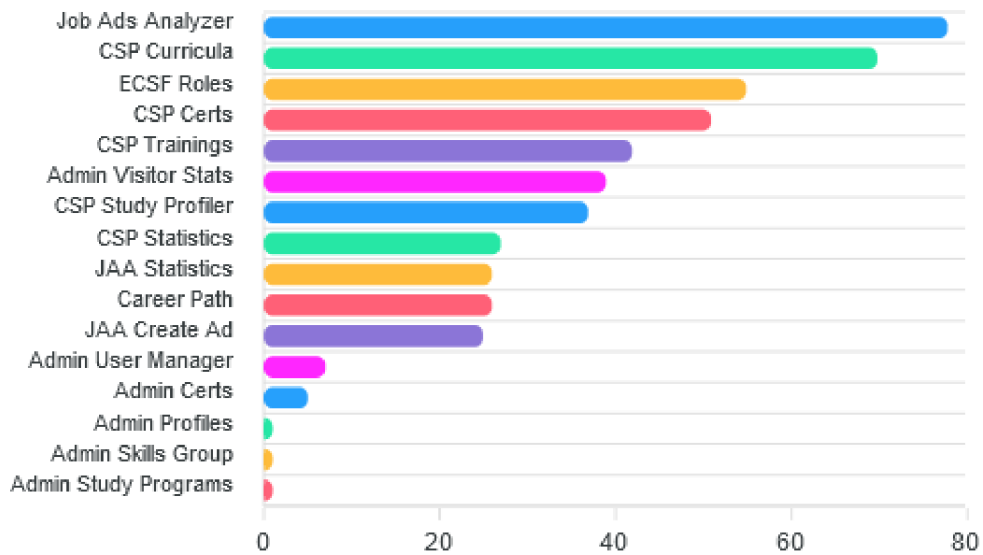


Fig. 2.6: Distribution of Visits Among Application Sections

2.5 Implementing User-Friendly URLs

Traditional web applications were built as multi-page applications (MPAs), where each page is a separate HTML (Hypertext Markup Language) document. When a user navigates to a different page, the browser requests it from the server, resulting in a full page reload. This change is reflected in the URL (Uniform Resource Locator), which is different for each page. In contrast, React applications are typically single-page applications (SPAs), where the entire application runs on a single page, and navigation does not involve full page reloads. In a React application, the URL does not change by default; however, this feature can be added using React Router.

We decided to implement functional URLs in the application to improve accessibility, navigation, and overall user experience. The application now allows users to access different parts by changing the URL, and it also updates the URL when users navigate to different sections using internal links.

The URLs are organized in a hierarchical structure, meaning that, for example, all the sub-sections of the admin area follow a pattern like `’/admin/specific-page.’` This structure helps maintain a clear and logical navigation system.

2.6 Study Program Analysis

To test the accuracy of the adapted ML algorithm, we applied it to the "Information Security" master's program (MPC-IBE)[55] offered by Brno University of Technology (BUT). This two-year program comprises a total of 60 courses: 15 compulsory, 17 compulsory-optional, and 28 optional. The courses collectively amount to 271 ECTS credits, with students required to earn a minimum of 120 credits. The study program encompasses not only technical fields such as cryptography, mathematics, and informatics, but also humanistic disciplines, with a special emphasis on law. This approach provides students with a comprehensive understanding of cybersecurity, addressing both technical and legal aspects.

To analyze the course data of the program, it first had to be collected. I reviewed each course and extracted the relevant data required by the CSP to add a new course. Specifically, this included the course name, type (whether compulsory or optional, with compulsory-optional courses classified as compulsory), semester (winter or summer), training (indicating if the course requires laboratory exercises), and the number of ECTS credits. For the course descriptions used in the ML analysis, I included the available course descriptions, aims, and syllabi of both the lectures and the laboratories of the course. This combination was then analyzed, and the output was later manually corrected by my supervisor. After adding all the courses, I used the export feature of the application to download the data of all the courses as a JSON file, which has been included as an annex in the thesis.

From the manual adjustment process, we can conclude that the ML algorithm tends to overcompensate by selecting more skill groups rather than fewer, often requiring the manual deletion of skill groups. This issue was particularly pronounced in humanistic disciplines such as law and languages, where technical topics are briefly mentioned in the descriptions but are not deeply covered in the course.

2.7 Cybersecurity Profiler Proof of Concept

Throughout this work, I have described the functionality of the CSP application and detailed the improvements I implemented. In this section, I will demonstrate how the application can be used to evaluate the coverage of ENISA profiles by university programs and how it can assist in designing these programs.

As an example study program, I used the previously mentioned "Information Security" master's program at the Brno University of Technology. The desired ENISA profile I selected was the Chief Information Security Officer (CISO).

First, I added all the compulsory courses from the MPC-IBE program into the CSP curricula designer and then selected courses from the available compulsory-

optional courses to fulfill the program requirements. As shown in Figure 2.7, most of the desired skill groups are already included in the added courses. Since there is no difference in the included skill groups between just the compulsory courses and the combined compulsory and compulsory-optional courses, I have not included an additional image of the output.

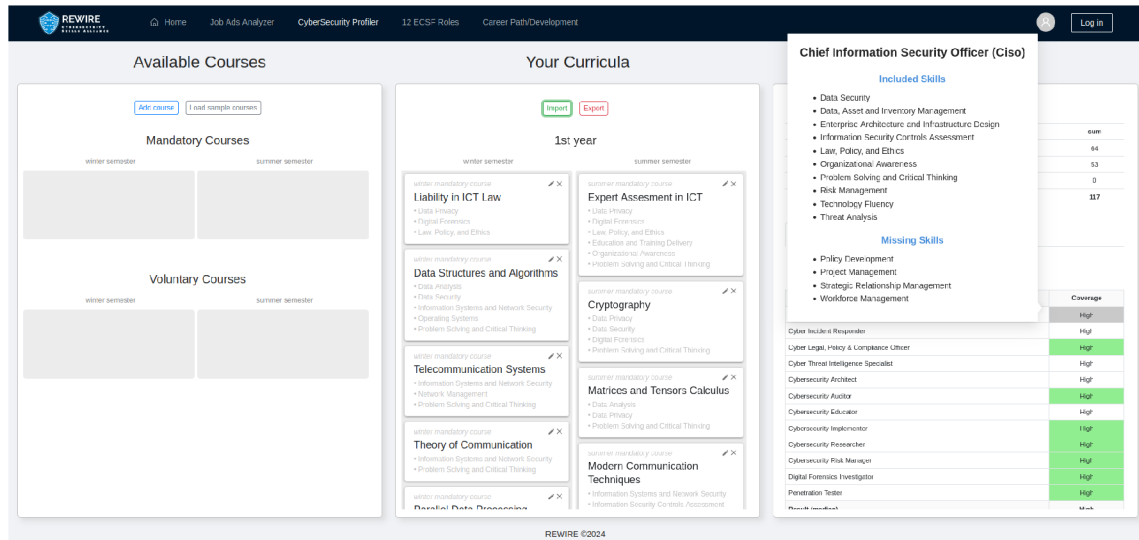


Fig. 2.7: Coverage of ENISA Profiles by the BUT FEKT Information Security Master's Program Compulsory and Compulsory-Optional courses

To supplement the selected profile and fulfill the European Credit Transfer and Accumulation System (ECTS) credit requirements of the program, I added an optional course named Business English. As shown in Figure 2.8, this course included all the missing skill groups except Policy Development.

After searching for a suitable course that includes Policy Development as a skill group, I discovered that no such course is available to MPC-IBE students. From this, we can conclude that a new course incorporating this skill group should be added to the program to ensure it covers all the desired skill groups from the CISO profile.

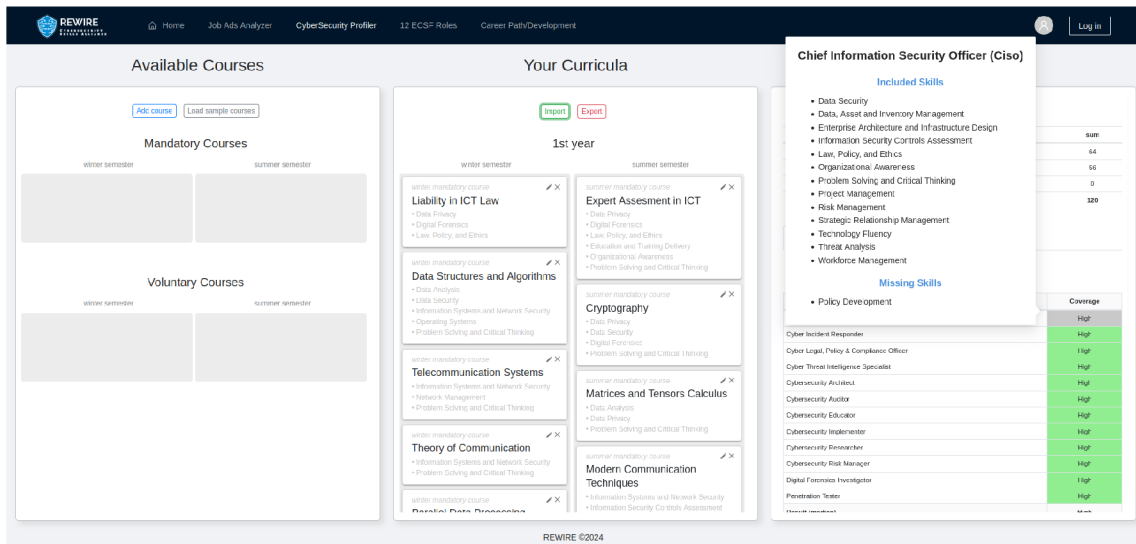


Fig. 2.8: Coverage of ENISA Profiles by the BUT FEKT Information Security Master's Program Compulsory, Compulsory-Optional, and Optional Courses

2.8 Script for Course Description Generation

As a solution to the previously mentioned problem, proposed by my supervisor, I created a script utilizing ChatGPT to generate a course description. The script sends the desired skill group name along with its associated key skills and key knowledge to ChatGPT, requesting it to generate a course description that includes said skill group and incorporates the key skills and key knowledge in the text. This PHP script uses an API (Application Programming Interface) provided by OpenAI, which allows it to interact with ChatGPT and query it. The generated result is then passed to the user. In the future, this script is planned to be integrated into the web application.

In this context, I used the script to generate a course description that includes the skill group "Policy Development." ChatGPT was provided with a request in the form of a sentence, asking it to generate a course description, without headers or any special characters, that includes the skill group "Policy Development" and the associated key skills and knowledge. I then used this generated description to supplement the MPC-IBE program, thereby fulfilling all the requirements of the CISO ENISA profile.

Conclusion

The goal of this bachelor thesis was to learn about modern GUI web application development and global cybersecurity frameworks like NIST NICE and ENISA ECSF and to implement a machine learning module for analyzing course descriptions and several other improvements to the Cybersecurity Profiler application.

For the theoretical part, I studied cybersecurity frameworks like NIST NICE and ENISA ECSF and familiarized myself with project REWIRE, their Cybersecurity Profiler, and the history and preconditions for its development. Since the main part of the practical work was the implementation of a machine learning algorithm, I dedicated a large part of the thesis to explaining machine learning with a special emphasis on natural language processing. Lastly, I described the tools and programming languages used during development, along with the used libraries.

In the practical part, I first describe the process of setting up the preexisting system and the many pitfalls encountered along the way. I then explain the integration of the machine learning algorithm into the application, followed by the various improvements I worked on, including responsive URLs and graphical visualization of user data in the admin section. Finally, I showcase the usefulness of the Cybersecurity Profiler using a study program from our university.

My experience with implementing and testing the machine learning algorithm in the CSP application makes me to believe that machine learning will play a significant role in the future of web applications. It is already gaining popularity, as seen with AI assistants, and its potential for enhancing user experience and functionality is immense. I have also come to understand the importance of role standardization, especially in a dynamic field like cybersecurity.

The improvements made to the application were implemented live and can be viewed in the production version available on the internet². The journey of working on this thesis has been a valuable learning experience. I have gained extensive knowledge about the professional world of cybersecurity and machine learning, and I have improved both my coding skills and writing ability. I am grateful for having had the opportunity to work on a project that will be used in the real world.

²<http://cyberability.rewireproject.eu/>

Bibliography

- [1] ICS2. 2023. Cybersecurity Workforce Study. ICS2. Available at: <https://www.isc2.org/Research>. [cit. 2024-03-15]
- [2] SPARTA. D9.2: Curricula Descriptions. August 2020. Available at: <https://www.sparta.eu/deliverables/>. [cit. 2024-05-17].
- [3] HAJNY, J; RICCI, S; PIESERSKAS, E; LEVILLAIN, O; GALLETTA, L; DE NICOLA, R. Framework, tools and good practices for cybersecurity curricula. IEEE Access, vol. 9, pp. 94723-94747, Jul. 2021, doi: 10.1109/. [cit. 2024-05-17].
- [4] REWIRE. 2020. REWIRE: Cybersecurity Skills Alliance - A new Vision for Europe. Available at: <https://rewireproject.eu/>. [cit. 2024-05-17].
- [5] DZURENDA, Petr; RICCI, Sara; SIKORA, Marek; NOHAVA, Martin; CHATZPOULOU, Argyro et al., DZURENDA, Petr a RICCI, Sara (ed.). Mapping the framework to existing courses and schemes. Online. REWIRE - Cybersecurity Skills Alliance A New Vision for Europe. 2023, roč. 2023, č. R3.4.1, s. 10-63. Available from: https://rewireproject.eu/wp-content/uploads/2023/03/REWIRE_R3.4.1_Deliverable-v8-Final-EC-Check.pdf. [cit. 2023-12-13].
- [6] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY [NIST]. Getting Started with the NICE Framework | NIST. Online. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY [NIST]. National Institute of Standards and Technology. 2020, 2023. Available from: <https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center/getting-started>. [cit. 2023-12-13].
- [7] INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What is machine learning? Online. IBM. What is machine learning? | IBM. B.r. Available from: <https://www.ibm.com/topics/machine-learning>. [cit. 2023-12-13].
- [8] J. Hu, H. Niu, J. Carrasco, B. Lennox and F. Arvin, "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14413-14423, Dec. 2020, doi: 10.1109/TVT.2020.3034800.
- [9] Yoosefzadeh-Najafabadi M, Earl HJ, Tulpan D, Sulik J, Eskandari M. Application of Machine Learning Algorithms in Plant Breeding: Predicting Yield From

- Hyperspectral Reflectance in Soybean. *Front Plant Sci.* 2021 Jan 12;11:624273. doi: 10.3389/fpls.2020.624273. PMID: 33510761; PMCID: PMC7835636.
- [10] INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM], What are large language models (LLMs)? Online. Available at: <https://www.ibm.com/topics/large-language-models>. [cit. 2024-05-17].
- [11] RICCI, Sara; SIKORA, Marek; PARKER, Simon; LENDAK, Imre; DANIDOU, Yianna et al. Job Adverts Analyzer for Cybersecurity Skills Needs Evaluation. Online. In: *Proceedings of the 17th International Conference on Availability, Reliability and Security*. New York, NY, USA: ACM, 2022, s. 1-10. ISBN 9781450396707. Available at: <https://doi.org/10.1145/3538969.3543821>. [cit. 2023-12-13].
- [12] INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What are recurrent neural networks? Online. INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What are Recurrent Neural Networks? | IBM. B.r. Available from: <https://www.ibm.com/topics/recurrent-neural-networks>. [cit. 2023-12-13].
- [13] HOCHREITER, Sepp and SCHMIDHUBER, Jürgen, 1997. Long Short-Term Memory. Online. *Neural Computation*. 1997-11-01, vol. 9, no. 8, pp. 1735-1780. ISSN 0899-7667. Available at: <https://doi.org/10.1162/neco.1997.9.8.1735>. [cit. 2024-05-18].
- [14] SATURNCLOUD. Bidirectional Transformers. Online. Bidirectional Transformers | Saturn Cloud. C2023. Available from: <https://saturncloud.io/glossary/bidirectional-transformers/>. [cit. 2023-12-13].
- [15] INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What is a neural network? Online. INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What is a neural network? | IBM. B.r. Available at: <https://www.ibm.com/topics/neural-networks>. [cit. 2023-12-13].
- [16] INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What is natural language processing? Online. INTERNATIONAL BUSINESS MACHINES CORPORATION [IBM]. What is natural language processing? | IBM. B.r. Available from: <https://www.ibm.com/topics/natural-language-processing>. [cit. 2023-12-13].
- [17] Node.js. About Node.js. Online. Available at: <https://nodejs.org/en/about>. [cit. 2024-05-18].

- [18] npm. About npm. Online. npm. Available at: <https://docs.npmjs.com/about-npm>. [cit. 2024-05-18].
- [19] Apache Software Foundation, About the Apache HTTP Server Project. Online. Apache Software Foundation. Available at: <https://httpd.apache.org/>. [cit. 2024-05-18].
- [20] pip. pip documentation. Online. Python Packaging Authority. Available at: <https://pip.pypa.io/en/stable/>. [cit. 2024-05-18].
- [21] Oracle. c2024. Getting Started with MySQL. Online. Oracle. Available at: <https://dev.mysql.com/doc/mysql-getting-started/en/>. [cit. 2024-05-18].
- [22] Amazon Web Services. c2024. What is LAMP Stack?. Online. Amazon. Available at: <https://aws.amazon.com/what-is/lamp-stack/>. [cit. 2024-05-18].
- [23] Oracle. c2012-2021. Introduction — phpMyAdmin 5.1.4 documentation. Online. phpMyAdmin. Available at: <https://docs.phpmyadmin.net/en/latest/intro.html>. [cit. 2024-05-18].
- [24] git. About Version Control. Git SCM. Available at: <https://git-scm.com/about>. [cit. 2024-05-19].
- [25] MDN Web Docs. What is JavaScript?. Online. Mozilla. Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [cit. 2024-05-18].
- [26] MDN Web Docs. Working with JSON. Online. MDN Web Docs. Available at: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>. [viewed 18 May 2024].
- [27] PHP Group. c2004-2024. PHP: What is PHP?. Online. PHP.net. Available at: <https://www.php.net/manual/en/intro-what-is.php>. [cit. 2024-05-18].
- [28] LINGO, London. 10 Reasons Why PHP is Still Relevant in 2023. Online. Medium, 9 May 2023. Available at: <https://medium.com/@london.lingo.01/10-reasons-why-php-is-still-relevant-in-2023-d533ed6dc687>. [cit. 2024-05-18].
- [29] Python Software Foundation. About Python. Online. Python Software Foundation. Available at: <https://www.python.org/about/>. [cit. 2024-05-18].
- [30] React. c2024. A JavaScript library for building user interfaces. Online. Meta Platforms, Inc.. Available at: <https://react.dev/>. [cit. 2024-05-18].

- [31] MDN Web Docs. c1998-2024. Introduction to the DOM. Online. Mozilla. Available at: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction. [cit. 2024-05-18].
- [32] Axios. Getting Started. Online. Axios. Available at: <https://axios-http.com/docs/intro>. [cit. 2024-05-18].
- [33] Bootstrap. Introduction. Online. Bootstrap. Available at: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>. [cit. 2024-05-18].
- [34] Bootstrap. Get started. Online]. Bootstrap. Available at: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. [cit. 2024-05-18].
- [35] React Bootstrap. c2024. Introduction. Online. React Bootstrap. Available at: <https://react-bootstrap.netlify.app/docs/getting-started/introduction/>. [cit. 2024-05-18].
- [36] React Bootstrap. c2024. About. Online. React Bootstrap. Available at: <https://react-bootstrap.netlify.app/docs/about/>. [cit. 2024-05-18].
- [37] ApexCharts. c2024. APEXCHARTS.JS. Online. ApexCharts. Available at: <https://apexcharts.com/>. [cit. 2024-05-18].
- [38] React Router. Feature Overview. Online. Remix Software Inc.. Available at: <https://reactrouter.com/en/main/start/overview>. [cit. 2024-05-18]
- [39] Beautiful Soup. Beautiful Soup Documentation. Online. Crummy. Available at: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [cit. 2024-05-18].
- [40] PILGRIM, Mark; BLANCHARD, Dan and CORDASCO, Ian. c2015. Chardet Documentation. Online. Chardet. C2015. Available at: <https://chardet.readthedocs.io/en/latest/>.
- [41] KIM, Taehoon; WURSTER, Kevin and JALILOV, Tahir. emoji. Online. emoji. Available at: <https://emoji-python.readthedocs.io/>. [cit. 2024-05-18]
- [42] Matplotlib Development Team. c2012-2023. Matplotlib: Visualization with Python. Online. Matplotlib. Available at: <https://matplotlib.org/>. [cit. 2024-05-18].
- [43] NumPy Developers. c2008-2022. NumPy Documentation. Online. NumPy. Available at: <https://numpy.org/doc/stable/>. [cit. 2024-05-18].

- [44] pandas. c2024. Python Data Analysis Library. Online. pandas. Available at: <https://pandas.pydata.org/>. [cit. 2024-05-18].
- [45] pandas. c2024. Package overview. Online. pandas. Available at: https://pandas.pydata.org/docs/getting_started/overview.html. [cit. 2024-05-18]
- [46] Scikit-learn Developers. scikit-learn: Machine Learning in Python. Online. Scikit-learn, Available at: <https://scikit-learn.org/stable/>. [cit. 2024-05-18].
- [47] NVIDIA. c2024. PyTorch. Online. NVIDIA. Available at: <https://www.nvidia.com/en-us/glossary/pytorch/>. [cit. 2024-05-18].
- [48] YEGULALP, Serdar. 2017 January 9. Facebook brings GPU-powered machine learning to Python. InfoWorld. Available at: <https://www.infoworld.com/article/3159120/facebook-brings-gpu-powered-machine-learning-to-python.html>. [cit. 2024-05-18].
- [49] ALI, Moez. June 2023. NLP with PyTorch: A comprehensive guide. DataCamp. Available at: <https://www.datacamp.com/tutorial/nlp-with-pytorch-a-comprehensive-guide>. [cit. 2024-05-18].
- [50] Torchtext Contributors. c2024. torchtext. PyTorch. Available at: <https://pytorch.org/text/stable/index.html>. [cit. 2024-05-18].
- [51] MOLTZAU, A. August 19 2020. What is torchtext?. Medium. Available at: <https://alexmoltzau.medium.com/what-is-torchtext-6e3528d15200>. [cit. 2024-05-18].
- [52] RAHUL, Shah. 240 April 2024. How to use progress bars in Python. Online. Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/how-to-use-progress-bars-in-python/>. [cit. 2024-05-18].
- [53] Hugging Face. Transformers. Online. Hugging Face. Available at: <https://huggingface.co/docs/transformers>. [cit. 2024-05-18].
- [54] GAZONI, Eric; CLARK, Charlie. openpyxl documentation. Online. Read the Docs, Available at: <https://openpyxl.readthedocs.io/en/latest/>. [cit. 2024-05-18].
- [55] Brno University of Thechnology. 2023-2024. curriculum "Information Security". <https://www.vut.cz/en/students/programmes/programme/8369>. [cit. 2024-05-19].

List of appendices

A	User manual	51
B	Contents of the Electronic Attachment	52
C	JSON Files	53
C.1	MPC-IBE_all_available_courses	53
C.2	MPC-IBE_compulsory_only	53
C.3	MPC-IBE_compulsory&compulsory-optional	53
C.4	MPC-IBE_Final	53

A User manual

Describes my contributions to the application, indicating where they can be found within the app and how to use them.

B Contents of the Electronic Attachment

Displays the directory structure of the application code, including only the files that I authored or modified.

```
/ ..... root directory of the attached file
├── client ..... contains frontend files
│   ├── package.json ..... added new libraries
│   ├── package-lock.json ..... added new libraries
│   └── src ..... contains React.js frontend
│       ├── index.js ..... added React Router wrapper
│       └── components ..... contains files separated into components
│           ├── admin
│           │   ├── Admin.scss ..... changed CSS
│           │   └── VisitorsStatistics.js ..... implemented ApexCharts charts
│           ├── app
│           │   └── App.js ..... Implemented React Router
│           └── csprofiler
│               ├── trainings
│               │   ├── Training.js ..... changed tooltips
│               │   ├── TrainingForm.js ..... changed tooltips
│               │   └── Trainings.js ..... changed tooltips
│               ├── study-profiler
│               │   └── Course.js ..... changed tooltips
│               ├── certifications
│               │   ├── Certification.js ..... changed tooltips
│               │   ├── CertificationForm.js ..... changed tooltips
│               │   └── Certifications.js ..... changed tooltips
│               └── curricula
│                   ├── Course.js ..... changed tooltips
│                   ├── CourseForm.js ..... frontend implementation of ML algorithm
│                   └── Statistics.js ..... changed tooltips
└── server ..... contains PHP backend files and Python files
    └── csprofiler
        ├── analyzeCoursedescription.php .... backend implementation of ML algo.
        ├── htaces ..... added new PHP files
        ├── generateCourseDescription.php ..... course generating script
        └── skill_groups.php ..... contains skill group definitions
```


C JSON Files

These JSON files are used in chapter 2.6 Cybersecurity Profiler Proof of Concept to showcase use and utility of the CSP application. They can be imported into the CSP curricula designer.

C.1 MPC-IBE_all_available_courses

This JSON file includes all the courses from the MPC-IBE study program, encompassing all compulsory, compulsory-optional, and optional courses available to students enrolled in the program.

C.2 MPC-IBE_compulsory_only

This JSON file contains only compulsory courses from the MPC-IBE study program.

C.3 MPC-IBE_compulsory&compulsory-optional

This JSON file contains compulsory and the selected compulsory-optional courses from the MPC-IBE study program.

C.4 MPC-IBE_Final

This JSON files contains compulsory, selected compulsory-optional, and selected optional courses from the MPC-IBE study program.