

# **Analýza starých dialektů pomocí umělé inteligence ve vztahu k Voynichově rukopisu**

Analysis of Ancient Dialects Using Artificial Intelligence in Relation to Voynich's  
Manuscript

Bc. Dan Kunz

Diplomová práce

Vedoucí práce: prof. Ing. Ivan Zelinka, Ph.D.

Ostrava, 2022

# Zadání diplomové práce

Student:

**Bc. Dan Kunz**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Analýza starých dialektů pomocí umělé inteligence ve vztahu k  
Voynichově rukopisu  
Analysis of Ancient Dialects Using Artificial Intelligence in Relation to  
Voynich's Manuscript

Jazyk vypracování:

čeština

Zásady pro vypracování:

Diplomová práce se zabývá aplikací hlubokého učení pro zpracování, analýzu a vizualizaci starých dialektů vzhledem k abecedě Voynichova rukopisu. Předpokládá se využití metody hlubokého učení ANN.

Struktura práce je následující:

1. Zhodnoťte stav na poli neznámých textů a jejich analýzy.
2. Vyberte vhodné a zatím nepoužité metody pro analýzu a vizualizaci získaných informací analyzovaných dialektů s využitím hlubokého učení.
3. Vyberte vhodný reprezentativní text s preferencí Voynichova rukopisu.
4. Proveďte nutný preprocessing dat z tohoto textu za pomoci jazyka Python.
5. Zpracujte a vizualizujte tato data.
6. Vyhodnoťte a shrňte v závěru.

Seznam doporučené odborné literatury:

- [1] Cai, J., Li, J., Li, W. and Wang, J., 2018, December. Deeplearning model used in text classification. In 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (pp. 123-126). IEEE.
- [2] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D., 2019. Text classification algorithms: A survey. *Information*, 10(4), p.150.
- [3] Amancio, D.R., Altmann, E.G., Rybski, D., Oliveira Jr, O.N. and Costa, L.D.F., 2013. Probing the statistical properties of unknown texts: application to the Voynich manuscript. *PLoS One*, 8(7), p.e67310.
- [4] Bowerman, C.L. and Lindemann, L., 2020. The Linguistics of the Voynich Manuscript. *Annual Review of Linguistics*, 7.
- [5] Rugg, G. and Taylor, G., 2017. Hoaxing statistical features of the Voynich Manuscript. *Cryptologia*, 41(3), pp.247-268.
- [6] Daruka, I., 2021. On the Voynich manuscript. *Cryptologia*, 45(1), pp.44-80.
- [7] Zelinka, I., Zmeskal, O., Windsor, L. and Cai, Z., 2019, June. Unconventional methods in Voynich manuscript analysis. In *Mendel* (Vol. 25, No. 1, pp. 1-14).
- [8] Herrmann, J.M., 2017. The Voynich Manuscript is written in natural language: The Pahlavi hypothesis. arXiv preprint arXiv:1709.01634.
- [9] Prokopič, D., 2008. Historie šifrování od starověku do novověku.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Ivan Zelinka, Ph.D.**

Datum zadání: 01.09.2021

Datum odevzdání: 30.04.2022

---

doc. Ing. Petr Gajdoš, Ph.D.  
*vedoucí katedry*

---

prof. Ing. Jan Platoš, Ph.D.  
*děkan fakulty*

## **Abstrakt**

Diplomová práce je zaměřena na částečnou analýzu jednoho z nejzáhadnějších děl všech dob, nazývaného Voynichův rukopis. První část bude věnována samotnému popisu knihy a bude zde ukázáno několik pokusů různých badatelů a vědců o získání užitečné informace z textu nebo vlastností jazyka. Druhá část je věnována vlastní analýze. Bude zjišťována podobnost Voynichova rukopisu s různými jinými spisy pomocí centralit. Rovněž bude provedena analýza pomocí neuronových sítí a následná vizualizace podobnosti písmen. Analýzy budou prováděny v programovacím jazyce Python.

## **Klíčová slova**

Voynichův rukopis, centrality, neuronové sítě, autoenkodéry, Python, Google colab

## **Abstract**

The diploma thesis is focused on a partial analysis of one of the most mysterious works of all time, called Voynich's Manuscript. The first part will be devoted to the description of the book itself and will show several attempts by various researchers and scientists to obtain useful information from the text or properties of the language. The second part is devoted to the analysis itself. The similarity of Voynich's manuscript with various other writings will be determined with the help of centralities. Neural network analysis and subsequent visualization of letter similarity will also be performed. The analyzes will be performed in the programming language Python.

## **Keywords**

Voynich manuscript, centralities, neural networks, autoencoders, Python, Google colab



## **Poděkování**

Rád bych na tomto místě poděkoval vedoucímu této práce panu prof. Ing. Ivanu Zelinkovi, Ph.D. za vedení, pomoc a veškeré poskytnuté materiály. Dále bych rád poděkoval svojí manželce za lásku a podporu a taky rodině za umožnění studia.

# Obsah

Seznam použitých symbolů a zkratek	8
Seznam obrázků	9
Seznam tabulek	11
<b>1 Úvod</b>	<b>12</b>
<b>2 O Voynichově rukopisu</b>	<b>13</b>
2.1 Historie . . . . .	13
2.2 Popis knihy . . . . .	14
2.3 Potencionální autoři . . . . .	15
<b>3 State of Art</b>	<b>16</b>
3.1 Skrytý Markonův Model pro lingvistickou analýzu Voynichova rukopisu . . . . .	16
3.2 Analýza frekvence písmen ve Voynichově rukopisu . . . . .	17
3.3 Modelování ve Voynichově rukopisu . . . . .	17
3.4 Zkoumání statistických vlastností neznámých textů . . . . .	18
3.5 Mohou nám evoluční algoritmy pomoci rozšifrovat Voynichův rukopis? . . . . .	18
3.6 Pravděpodobnost slov . . . . .	19
3.7 Šifrovací metody ve středověku . . . . .	20
3.8 Důkaz o lingvistické struktuře ve Voynichově rukopisu za použití spektrální analýzy	21
3.9 Crowdsourcingem k vyřešení záhady . . . . .	21
<b>4 Použité technologie</b>	<b>24</b>
4.1 Python . . . . .	24
4.2 Gephi . . . . .	25
4.3 Google Colab . . . . .	26

4.4	Neuronové sítě . . . . .	27
<b>5</b>	<b>Síťová analýza</b>	<b>30</b>
5.1	Centrality . . . . .	30
5.2	Mapy centralit . . . . .	33
5.3	Porovnání map centralit . . . . .	34
5.4	Zhodnocení síťové analýzy . . . . .	35
<b>6</b>	<b>Analýza za pomoci neuronové sítě</b>	<b>38</b>
6.1	Vstup sítě . . . . .	38
6.2	Sestrojení neuronové sítě . . . . .	40
6.3	Zobrazení výsledku neuronové sítě . . . . .	46
<b>7</b>	<b>Závěr</b>	<b>52</b>
	<b>Literatura</b>	<b>53</b>
	<b>Přílohy</b>	<b>57</b>

# Seznam použitých zkratek a symbolů

VR	– Voynichův rukopis
ANN	– Artificial Neural networks
EVA	– European Voynich Alphabet
nx	– Pythonový balíček jménem NetworkX

# Seznam obrázků

2.1	Ukázka rozložitelné stránky a struktury textu [3]. . . . .	13
2.2	Úryvek z botanické části knihy [5]. . . . .	14
3.1	Ukázka ručně psaného písma ve Voynichově rukopisu [8]. . . . .	17
3.2	Vztah mezi pravděpodobností logaritmického slova a relativní poloha bodu jedinečnosti pro slova délky 4 - 8 písmen [12]. . . . .	20
3.3	Ukázka ze Zooniverse [18]. . . . .	22
3.4	Ukázka z aplikace [19]. . . . .	23
4.1	Nejvíce používané jazyky pro Data Science [22]. . . . .	25
4.2	Porovnání iPythonu, Jupyteru a Google Colabu [27]. . . . .	26
4.3	Ukázka struktury neuronové sítě [31]. . . . .	27
4.4	Ukázka struktury autoenkodéru [33]. . . . .	28
5.1	Síťové zobrazení okna o velikosti 300 začínající na pozici 16 558. . . . .	31
5.2	Ukázka z Google Colab sešitu. . . . .	32
5.3	Mapy centralit pro Bibli. . . . .	34
5.4	Mapy centralit pro Voynichův rukpis. . . . .	35
5.5	Mapy centralit pro dílo Stařec a moře. . . . .	36
5.6	Mapy centralit pro náhodně vybraná slova voynichova rukopisu. . . . .	36
5.7	Zvětšení vybraných map centralit . . . . .	37
6.1	Ukázka Image Corresponding Points na písmenu [38]. . . . .	39
6.2	Písmena abecedy namapovaná na písmena Voynichova rukopisu [39]. . . . .	40
6.3	Křivka učení. . . . .	42
6.4	Vizualizace funkce Sigmoid [43]. . . . .	43
6.5	Vizualizace funkce Relu [45]. . . . .	44
6.6	Porovnání optimalizačních algoritmů na datové sadě MNIST [46]. . . . .	46
6.7	Vizualizace výsledku autoenkodéru. . . . .	47
6.8	Vizualizace výsledku autoenkodéru z jiného pozorovacího úhlu. . . . .	48

6.9	Vizualizace výsledku autoenkodéru pro neomezená data. . . . .	49
6.10	Ukázka podobnosti písmen mezi po aplikování neuronové sítě [38]. . . . .	50
6.11	Podobnost písmen ve Voynichově abecedě. Písmena jsou seskupená do shluků v závislosti na vzájemné podobnosti [38]. . . . .	50

# Seznam tabulek

6.1	Primitivní autoenkodér . . . . .	41
6.2	Vícevrstvý autoenkodér . . . . .	41

# Kapitola 1

## Úvod

V celé historii se lze často setkat s texty, jenž byly zašifrovány a jejichž text byl nálezcům rukopisu neznámý. Mnoho takových textů pomohla rozluštit až výkonná výpočetní zařízení jako třeba superpočítače, které zvládnou vykonat až biliony operací za pouhou sekundu. V této diplomové práci je však pojednáváno o spisu, jenž se nikdy dekodovat nepodařilo. A právě toto stále přitahuje velké množství vědců a nedá jim v noci spát. Vždyť jak řekl Albert Einstein: „Nejkrásnější věc, kterou můžeme objevit, je záhada.“ [1].

Tato práce není zaměřena na dekodování samotného textu nýbrž na analýzu jeho částí a následné porovnání s různými jinými historickými spisy jako je Bible nebo dílo Stařec a moře od Ernesta Hemingwaye.

První část je věnována současnému stavu Voynichova rukopisu na vědeckém poli a následně popisu a vysvětlení technologií použitých při samotné analýze. Druhá část je zaměřena na popis praktické části. Rozbor analýz a porovnání výsledků pro různé texty. Jedná se o hledání centralit a shluků v textech a porovnávání podobnosti. Poté bude na Voynichovu abecedu aplikována neuronová síť a výsledky vizualizovány.



## Kapitola 2

# O Voynichově rukopisu

### 2.1 Historie

Je to již okolo 600 let, kdy byla napsána jedna z nejzáhadnějších knih všech dob. Tato kniha dostala své jméno po kupci Wilfridovi Michaelu Voynichovi, který ji začátkem 20. století získal v Anglii. Odtud tedy vznikl název Voynichův rukopis. Samotný autor nebo původní název však zůstává neznámý [2].

Počátkem minulého století cestoval po Evropě antikvář amerického původu Wilfried Michael Voynich a vykupoval staré vzácné tisky. V Itálii v jezuitské knihovně našel knihu, která se zcela odlišovala od všech ostatních. Nebyla nijak zdobená jako ostatní tamní knihy. Na první pohled však bylo poznat, že je velice stará. Nicméně to nejvíce pozoruhodné na celém nálezů bylo, že byla napsána zcela neznámým jazykem, který nebyl ničemu podobný (obrázek 2.1). Rukopis kdysi údajně koupil Rudolf II., ale o tom co se s ním pak dělo, víme jen velmi málo [2].



Obrázek 2.1: Ukázka rozložitelné stránky a struktury textu [3].

## 2.2 Popis knihy

Kniha obsahuje přes 230 stran, z nichž některé jsou rozložitelné. Text knihy je napsán na pergamen. Spis je napsán elegantním, ale jinak neznámým písmem a skoro všechny jeho strany obsahují ilustrace těžko identifikovatelných bylin (obrázek 2.2), souhvězdí či soustav trubic přepravujících zvláštní tekutiny a kresby drobných ženských postav. Rozměry knihy jsou 16x23 centimetrů. Nemá desky ani typickou úvodní stranu s informacemi o autorovi, místě vydání nebo názvu samotného díla. Obsahuje asi 35 tisíc slov různé délky [4].

Kniha obsahuje veliké množství ilustrací. Tyto ilustrace mohou být rozděleny do těchto základních šesti sekcí:

1. botanická
2. astronomická,
3. astrologická,
4. biologická,
5. kosmologická,
6. farmakologická [6].



Obrázek 2.2: Úryvek z botanické části knihy [5].

## 2.3 Potencionální autoři

Zjištění autora spisu by mohlo markantně přispět k vyřešení této záhadné knihy, a proto po této informaci mnoho výzkumníků báдалo.

Člověk, který rukopis prodal Rudolfovi II. tvrdil, že jej napsal anglický filozof Roger Bacon. Tento vědec žil od roku 1214 do 1292 a byl známý jako jeden z nejvzdělanějších lidí své doby. Avšak toto mohl prodejce tvrdit, jen aby knihu prodal s velkým ziskem.

Často spojované jména s touto knihou jsou matematik a astronom John Dee a alchymista jménem Edward Kelly. Je známo, že tyto dvě osobnosti spolu spolupracovaly. John Dee se snažil jakýmkoliv způsobem komunikovat s anděly a Kelly tvrdil, že právě to dokáže. Kelly je známý především pro pokusy proměnit olovo ve zlato. Kelly údajně přišel ke záhadné knize, která popisovala alchymistické pokusy. Rovněž John Dee měl údajně neobvyklou knihu popsanou zvláštními hieroglyfy.

Další významnou postavou je Leonardo DaVinci. Da Vinci se řadí mezi nejslavnější vynálezce všech dob a jeho život je obklopen mnoha tajemstvími. Někteří lidé jako například Edith Sherwoodová se domnívají, že právě on je autor Voynichova rukopisu.

Všechna tato jména se dlouho zvažovala a nikdo je nedokázal potvrdit ani vyvrátit. To vše do doby, než se provedla radiokarbonová metoda, která určila, že text byl napsán mezi roky 1404 až 1438. To je později než žil Roger Bacon, ale dříve než žili Leonardo da Vinci, Edward Kelly či John Dee. A tak se pátrání po autorovi dostalo do slepé uličky. Dodnes autor knihy není znám [2].

## Kapitola 3

# State of Art

Fakt, že tento dokument ještě nikdo nedokázal vyřešit, lidi přitahuje. A proto je spousta veřejně přístupných publikací, kde se můžeme dočíst, o co se autoři pokoušeli, co k tomu použili a k jakému závěru nakonec dospěli. V rámci této diplomové práce je prováděna rešerše o takových pracích. Na druhou stranu se jimi nebudeme podrobně zabývat a do hloubky probírat funkcionalitu pokusů uvedených v těchto publikacích. Na začátku nutno podotknout, že spousta z nich se zakládá na nějakém tvrzení, které není dostatečně vědecky podloženo, tudíž není vždy dobré přikládat tomu velkou váhu. Nicméně i přesto mohou posloužit jako inspirace nebo naopak ukáží, kudy cesta nevede.

### 3.1 Skrytý Markonův Model pro lingvistickou analýzu Voynichova rukopisu

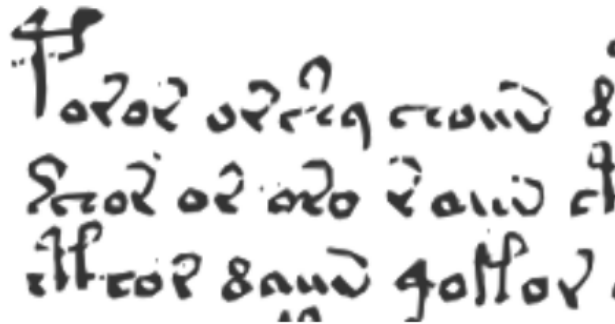
Ph.D. Luis Acedo se snažil o rozlišení samohlásek od souhlásek v textu VR. Ale jelikož nejčastějšími písmeny anglické abecedy je písmeno "e" a písmeno "t", není snadné najít jednoduchý vzor. Autora napadlo pro tento problém využít takzvaný "Hidden Markov Model" v češtině známý jako Skrytý Markonův model. Je to velice dobrý nástroj při modelování časových řad libovolné sekvence dat. Tento model byl již dříve úspěšně aplikován v oblasti matematické lingvistiky. Za pomoci určitého počtu skrytých stavů trénoval síť použitím algoritmů pro optimalizaci modelů. Tímto způsobem lze získat takzvané přechodové a pozorovací matice, které následně mohou být porovnány.

Na základě této analýzy autor dochází k závěru, že mezi dvěma stavy dochází k přechodům s podobnou frekvencí jako v jiných jazycích. Navíc identifikace souhlásek a samohlásek odpovídá některým dřívějším pokusům "zdola-nahoru" o rozluštění rukopisu. Autor uvádí, že hlavním výsledkem tohoto příspěvku je důkaz, že existují dva stavy spojené se symboly v rukopise, které by mohly odpovídat souhláskám a samohláskám [7].

## 3.2 Analýza frekvence písmen ve Voynichově rukopisu

Grzegorz Jaśkiewicz použil metodu zvanou frekvenční distribuce písmen. Tato metoda spočívá v tom, že každému písmenu přiřadí jeho frekvenci výskytu v celém textu. V této publikaci bylo srovnáno rozložení okolo 300 jazyků s textem Voynichova rukopisu. Na základě této přirozené vlastnosti jazyka byly vybrány ty nejpodobnější.

Následně autor udělal porovnání a dospěl k závěru, že písmena jsou podobná jak evropským, tak i asijským písmům (obrázek 3.1), což přispívá teorii, že autor knihy byl cestovatel. Dalším poznatkem bylo, že text obsahuje málo samohlásek [8].



Obrázek 3.1: Ukázka ručně psaného písma ve Voynichově rukopisu [8].

## 3.3 Modelování ve Voynichově rukopisu

V této publikaci se autoři z university Yale v Connecticutu zaměřili na zkoumání Voynichova rukopisu pomocí tématického modelování. Tématické modelování je sadou výpočetních metod, které se využívají k identifikaci shluků subjektů v textu. Využívají latentní dirichletovou alokaci, latentní sémantickou analýzu a nezápornou maticovou faktorizaci pro seskupení stránek Voynichova rukopisu do "témat". Následně porovnali tyto témata odvozená z výpočtových modelů se shluky odvozenými z Voynichových kreseb a z paleografické analýzy. Povedlo se jim zjistit, že výpočetně odvozené shluky úzce odpovídají spojení autora Voynichova rukopisu a ilustrací, což utvrzuje teorii, že Voynichův rukopis obsahuje smysluplný text a ne jen náhodné znaky.

Pozoruhodným poznatkem také je, že podle výsledků nezáporné maticové faktorizace je astrologická sekce vlastní téma a je vzdálená od všech ostatních. Je také zajímavé vidět úzké seskupení názorných rozetových diagramů, biologických a kosmologických témat napříč analýzami. Zdá se, že analýzy naznačují, že tyto části patří k širšímu souhrnnému tématu, ačkoli existují individuální rozdíly. To je obzvláště zajímavé, protože odstavce označené hvězdičkou a části rozetových sekcí

nenabízejí mnoho vizuálních vodítek k tomu, o čem by text mohl být, zatímco přístup modelování témat, dokonce s pevným počtem a náhodným výběrem slov, naznačuje, že mohou souviset s biologickými ilustracemi a textem [9].

### 3.4 Zkoumání statistických vlastností neznámých textů

Bylo již použito mnoho metod statistické fyziky, které dokáží odhalit vzory v textu, ale nikdo se nepokusil provést komplexní výzkum na rozdíly vzájemné závislosti mezi syntaktickými a sémantickými faktory. Autor Matjaz Perc ve své publikaci navrhuje rámec k rozhodnutí, zdali je text Voynichova rukopisu kompatibilní s přirozenými jazyky. K tomuto účelu Perc kombinuje tyto 3 typy metod:

- Založené na statistikách prvního řádu (průměr, odchylka), kde se používají údaje o slovech jako třeba frekvence,
- Založené na metrikách ze sítí reprezentující text,
- Založené na konceptu intermitentnosti a analýzu časových řad.

Zároveň navrhuje jak určit důležitost těchto měření při zkoumání neznámých textů bez ohledu na abecedu. Statistické vlastnosti slov a knih byly získány pro srovnávací studie zahrnující knihu (Nový zákon) v 15 jazycích a různé části textu napsané v angličtině a portugalštině. Účelem bylo identifikovat rysy schopné odlišit smysluplný text od jeho zamíchané verze (kde je pozice slov náhodná), a poté určit blízkost jednotlivých částí textu. Snahou autora není VR rozluštit, ale dokázat, že je kompatibilní s jinými přirozenými jazyky a také najít klíčová slova. A Právě to se dle jeho názoru podařilo [10].

### 3.5 Mohou nám evoluční algoritmy pomoci rozšifrovat Voynichův rukopis?

Jak již název napovídá autor této publikace Daniel Deavatman Hromada se zaměřuje na dešifrování za pomoci evolučních algoritmů. Základní složkou dekodovacího algoritmu je proces maximalizace fitness funkce, jehož cílem je najít optimální sadu substitučních pravidel, která umožní přepsat část rukopisu, které říkáme "kalendář" do seznamů ženských jmen. Autor se snaží využít obrázků jako klíče k textu. Tato metoda může být pravdivá, jedině pokud jsou splněny následující podmínky [11]:

1. Nápisy u žen jsou jejich jména.
2. Vlastní jména jsou méně náchylná k diachronní změně, než ostatní jazykové jednotky.

3. Místo aby obsahoval propracovanou esoterickou šifru, VR jednoduše přenáší text napsaný neznámým písmem.

Jádrem metody je optimalizační algoritmus, který hledá kandidáta transkripční abecedy aX, kterého když přiložíme ke kalendáři VR, dává výstup, jehož členy by se ideálně měly vyskytovat v jiném listu nazvaném „Crib“. Optimalizace se provádí evoluční strategií. Jednotlivé chromozomy kódují kandidátní transkripční abecedu a fitness funkce je dána jako součet délek všech tokenů, které byly úspěšně vybrány. Postup je inspirován strukturou genetického algoritmu (CGA, P+46). Počáteční populace je náhodně vygenerovaná a fitness funkce slouží jako hlavní výběrový operátor [11].

Autor jména porovnával se dvěma skupinami jazyků. Se slovanskými jazyky a s hebrejštinou.

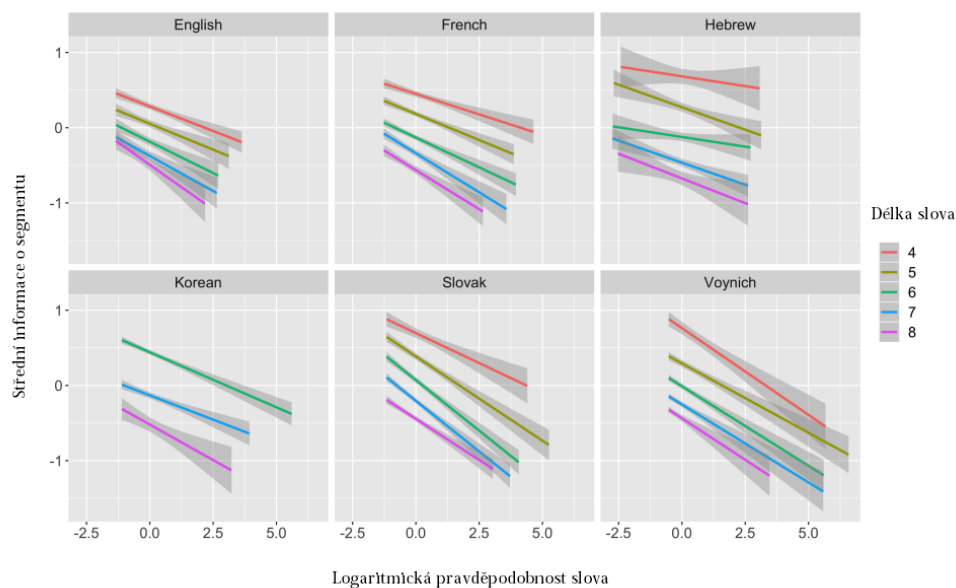
1. Slovanské jazyky - experiment vedl k identifikaci více substitučních abeced, které umožňují mapovat 240 písmen EVA, obsažených ve 40 různých slovech přítomných v kalendáři, na 35 sekvencí připomínajících ženská jména, vyjmenovaných mezi 13815 položkami.
2. Hebrejšтина - Výsledky druhého experimentu ukazují, že pokud kalendář obsahuje seznamy hebrejských jmen, pak by tato jména byla spíše ženského než mužského rodu.

Z obou experimentů vyplývá, že pokud jde o jména, byla psaná zprava doleva [11].

### 3.6 Pravděpodobnost slov

Layfield a kol. [12] se snaží poukázat na další důkazy pro tvrzení, že VR má několik charakteristik přirozeného jazyka. Hlavní novinkou je povaha metriky. Autor svou práci staví na publikaci Kinga a Wedela, kteří ve svém díle uvádí, že v sekvencích zvuků a jejich poloze v rámci slovních hranic existují jisté vzorce, které jsou sdíleny naskrz datovým souborem různých jazyků. Zejména demonstrují, že slova, která se vyskytují s menší pravděpodobností, obsahují více zvuků, ale hlavně obsahují také zvuky, které celkově sdělují více jednoznačných informací. Tento vzor bývá nejsilnější na začátcích slov, kde zvuky mohou poskytnout nejvíce informací. K této analýze je v této práci použita sada EVA.

Pro zjištění zda segmenty informace a pravděpodobnost slova obsahují stejný vzor, který našli King a Wedel, se vypočte bezkontextová pravděpodobnost slov pro všechna slova vydělením počtu slov celkovou sumou. Vybrali 5 jazyků, které se odlišují z pohledu jazykových rodin a morfológickou složitostí, které se zaměřují na indoevropskou rodinu jazyků kvůli jejich údajnému vztahu k Voynichově rukopisu. Výsledkem je, že VR opisuje stejný vzor jako těchto 5 předem zvolených jazyků v tom, že méně častá slova obsahují více informací (obrázek 3.2). VR sdílí charakteristiku přirozených jazyků, kterou prezentovali King a Wedel [12].



Obrázek 3.2: Vztah mezi pravděpodobností logaritmického slova a relativní poloha bodu jedinečnosti pro slova délky 4 - 8 písmen [12].

### 3.7 Šifrovací metody ve středověku

Alise Gladyševa z Vilniuské univerzity se zaměřila na metody používané v tamní době. Ve středověku se používaly především metody jako transpozice. Tato metoda fungovala tak, že se jednoduše popřeházely písmena dle nějakých předem určených pravidel. Nebo také substituce, neboli nahrazení písmen. Tyto metody se pak různě vylepšovaly a kombinovaly. Dále se autorka zaměřila na okrajové popisky, které se v knize vyskytují. Například popisy u něčeho, o čemž se domnívá, že jsou měsíce v roce. V historii bylo mnoho pokusů tyto měsíce přečíst, a předpokládalo se, že je to španělsky. Avšak autorka se domnívá, že je to napsáno ve středověké galicijštině. Dospěla také k názoru, že šifrovací algoritmus Voynichova rukopisu byl ovlivněn substituční šifrou s použitím polyalfabetické šifry a taky transpoziční šifrou pro dvojité zašifrovaný alchymistický text. Autorka použila metody jako analýzu morfologie struktury slov, analýzu syntaxe větných struktur, sémantiku slov nebo lexikologii slov. Přichází také s důkazem, že znak 'o' jenž se v textu mnohokrát vyskytuje, ve skutečnosti není nula, ani zašifrované písmeno, ale určitý člen jednotného čísla mužského rodu ve středověké galicijštině. Takovýchto poznatků ve svém díle uvádí více. Všechny tyto poznatky se ovšem zakládají na předpokladu, že její úvaha o tom, že je VR napsán ve středověké galicijštině, je správná [13].



### 3.8 Důkaz o lingvistické struktuře ve Voynichově rukopisu za použití spektrální analýzy

Gabriel Landini ve své publikaci uvádí několik statistických charakteristik textů, které jsou společné pro přirozené jazyky a ukazuje, že Voynichův rukopis není jen sbírka náhodných slov. Pro tento účel využívá spektrální analýzu k charakterizaci korelace symbolů v rozličných spisech a odhaluje následující poznatky:

1. Long-range korelace ve velkých měřících.
2. Vrchol indikující délku modálního tokenu.
3. Pokles ve výkonu spektra v porovnání s kódovanými verzemi textu.
4. Vrchol, který korespondoval s periodickými strukturami v textu.

Tyto rysy Long-range korelace ve Voynichově rukopisu ( $r > 400$ ) mohou být způsobeny nerovnoměrnou hustotou znaků podél textu. Důvodů, proč tomu tak je, může být hned několik: změna jazyku, změna témat nebo vložení štítků do textu. Pokles síly ve spektrální analýze ve vztahu ke kódovaným textům naznačuje, že text není pouze náhodnou sekvencí znaků. Jistý Currier si již v roce 1976 všiml, že některé znaky a slova se vyskytovaly na konkrétních místech (třeba na koncích nebo začátcích řádků), zatímco jiné se na těchto pozicích nevyskytly nikdy. Navíc některé znaky byly běžné na začátcích odstavců, takže to vyvolávalo otázky, jestli nejsou tyto znaky nulové nebo jen pro okrasu, protože některé jiné se chovaly jako znaky přirozeného jazyka.

Známý kryptolog William Friedman věřil, že Voynichův rukopis byl napsán v nějakém syntetickém, či umělém jazyce. Currier byl toho názoru, že slova v textu nemusí nutně být "slova" a rovněž uvažoval o tom, že by Voynichův rukopis mohl být napsaný syntetickým jazykem. J. Stolfi přišel s teorií, že by rukopis mohl být napsán v nějakém tónovém jazyce jako je třeba čínština, ale psaný abecedním písmem, což by vysvětlovalo omezenou strukturu slov ve Voynichově rukopisu. Autor této publikace Landini se na základě zde uvedených tvrzení přiklání k tomu, že byl spis napsán v přirozeném jazyce, ale vzhledem k neobvyklým kuriozitám nelze vyloučit ani alternativy. To, jestli pozorování liniové struktury, zde potvrzené Landiniho korelační analýzou, je způsobeno metrickými omezeními, nebo přítomností nulových, nebo okrasných znaků, může být předmětem budoucích výzkumů [14].

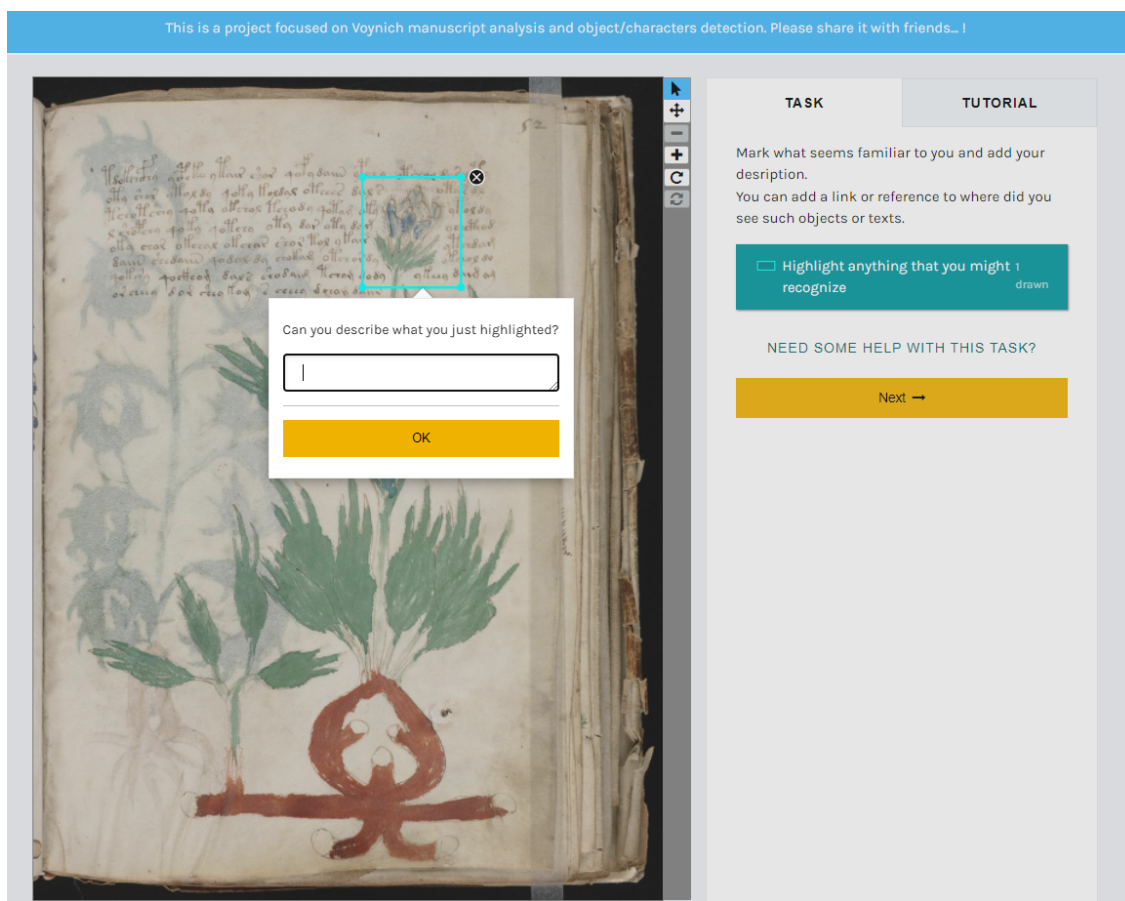
### 3.9 Crowdsourcingem k vyřešení záhady

Crowdsourcing je zapojení "davů" pro dosažení cíle. Může probíhat napříč různými odvětvími a skupinami. Lidé mohou zasílat své nápady prostřednictvím sociálních sítí, mobilních aplikací anebo

přímo skrze Crowdsourcingové platformy k tomu určené. Někdy může být tato činnost placená, záleží na povaze daného úkolu, ale většinou jde o ryze dobrovolnou činnost. Jedním z ukázkových příkladů mohou být třeba recenze na Google nebo kdekoliv jinde. Pokud jste někdy recenzovali např. nějakou restauraci, stali jste se, možná nevědomky, součástí Crowdsourcingu [15][16].

A právě této organizované činnosti chce pro bádání ohledně Voynichova rukopisu využít Bc. Aneta Opletalová pod vedením prof. Ing. Ivana Zelinky, Ph.D. Pro tento účel využívá platformu jménem Zooniverse.

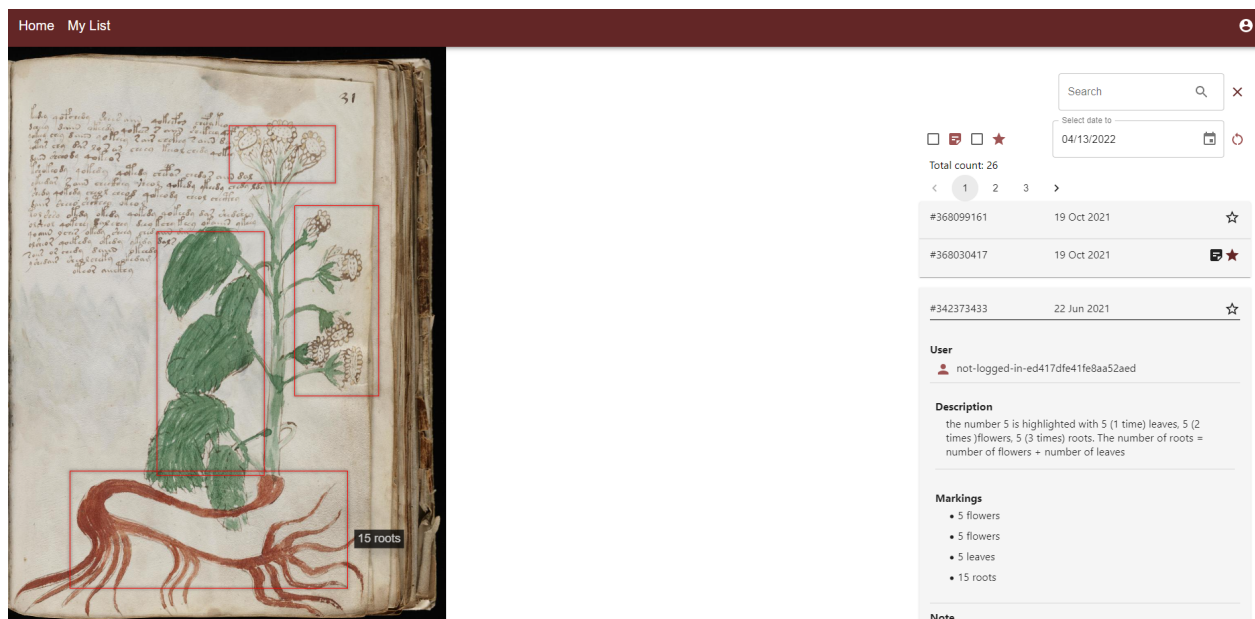
Zooniverse je největší a nejpoblárnější platforma pro výzkum založená na lidech. Toto je možné díky více než miliónu dobrovolníků po celém světě. Cílem této platformy je umožnit výzkum, který by jinak nebylo možné uskutečnit [17].



Obrázek 3.3: Ukázka ze Zooniverse [18].

Autorka na této platformě vytvořila projekt Voynichův rukopis, kde nahrála jeho stránky. Nyní má každý člověk možnost dát v potaz své postřehy (obrázek 3.3). Následně navrhla interaktivní rozhraní (obrázek 3.4), které umožňuje lepší práci s daty, které jsou vyexportovány ze Zooniverse a

jejich samotnou vizualizaci [19].



Obrázek 3.4: Ukázka z aplikace [19].

Tento projekt je stále na začátku, ale má velký potenciál, protože právě s větším počtem badatelů se zvyšuje šance, že někdo přijde na něco, co by mohlo být pro rozluštění klíčové. Důležitým krokem je nyní popularizace tohoto projektu, aby se mohlo zapojit co nejvíce lidí.

## Kapitola 4

# Použité technologie

V dnešní době je již mnoho nástrojů a knihoven, jež umožňují věci, které byly dříve náročné, ne-li neproveditelné. V této kapitole je popsáno jaké technologie byly použity a z jakého důvodu.

### 4.1 Python

Pro veškeré experimenty a výpočty bude využit vysokoúrovňový programovací jazyk jménem Python. Tento jazyk je univerzální a široce používaný (obrázek 4.1). Software v něm vyvíjí firmy jako Google, Youtube, Reddit a další [20].

V roce 1991 tento jazyk navrhl nizozemský programátor Guido van Rossum. Byl vyvinut zejména kvůli vylepšení čitelnosti kódu a jeho syntaxe vývojářům umožňuje psát kód na méně řádků. Aktuálně nejnovější verze pythonu je 3.7.3.

Tento programovací jazyk bude použit zejména pro velké množství užitečných knihoven, jež tento jazyk poskytuje [21].

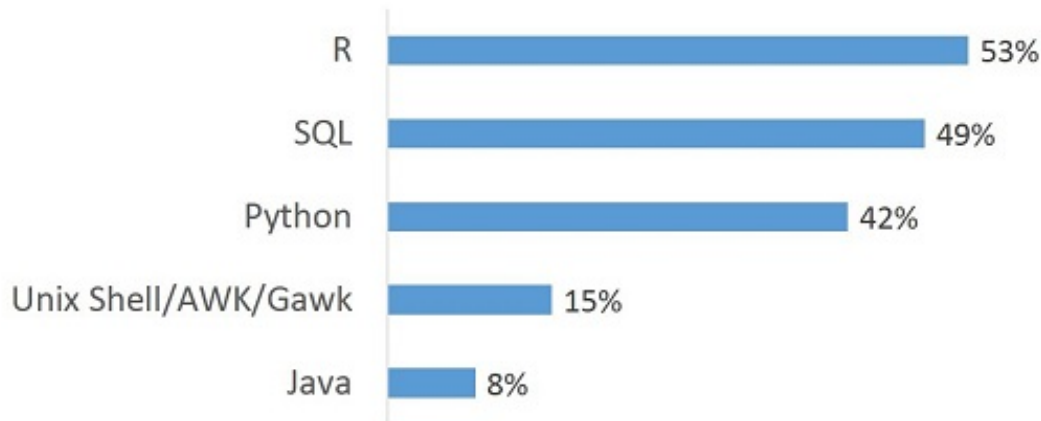
#### 4.1.1 Pandas

Pandas je open source balíček používající se nejčastěji pro analýzu a strojové učení. Je postaven na balíčku Numpy. Díky své popularitě je většinou součástí každé distribuce Pythonu. Ušnadňuje mnoho časově náročných opakujících se prací s daty. Jako třeba normalizace dat, přesun dat, kontrola dat atd. [23].

#### 4.1.2 NetworkX

Tato open-source knihovna je zkonstruována k práci s grafy a jejich zkoumání. Tento projekt začal již v roce 2005. Zakládá si na tom, aby byla uživatelsky přátelská. Umí dobře pracovat se sítěmi a poskytuje velice užitečné algoritmy. Nabízí užitečné nástroje, ale má problém se škálovatelností, při zpracování milionových dat může docházet k nedostatku paměti [24].

### Top 5 of programming languages mostly used for data science's activities



Obrázek 4.1: Nejvíce používané jazyky pro Data Science [22].

#### 4.1.3 Matplotlib

Matplotlib je multiplatformní knihovna ke grafickému vykreslování a vizualizaci dat. Nabízí open source alternativu k programovacímu jazyku Matlab. Vývojáři mohou také využít rozhraní API matplotlib k vložení grafů do aplikací GUI. Skript této knihovny je vytvořen tak, aby bylo možné graf vykreslit již pomocí několika řádků. Využívá knihovny jako Numpy a Pandas [25].

#### 4.2 Gephi

Gephi je open-source software určený zejména pro analýzu a vizualizaci grafů a sítí. Využívá 3D renderovací engine, který mu v reálném čase umožňuje zobrazovat obsáhlá data. Tato metoda využívá grafiku počítače a procesor nechává volný pro ostatní výpočty. Dokáže se vypořádat i s velkými sítěmi (třeba i přes 20 tisíc vrcholů) a využívá výhodu vícejádrových procesorů. Dále poskytuje jednoduchý, ale rozsáhlý přístup k síťovým datům, prostorovou úpravu, filtrování, navigaci manipulování a shlukování. Rovněž představuje dynamické funkce a zdůrazňuje klíčové body dynamických sítí [26].

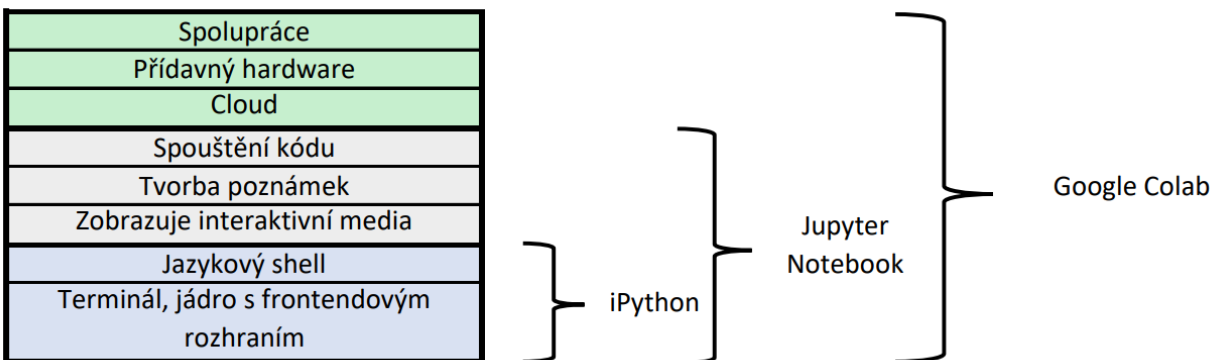
## 4.3 Google Colab

Google Colab je prostředí běžící na cloudu, které vývojáři umožňuje psát a spouštět kód napsaný v programovacím jazyce Python.

Proč použít Google Colab?

1. Obsahuje již předinstalované knihovny (ty nejvíce používané), což je velkou výhodou, jelikož uživatel je nebude muset ručně doinstalovávat.
2. Celý obsah je uložen na cloudu, takže se uživatel nemusí obávat, že by kvůli chybě na vlastním zařízení přišel o data.
3. Umožňuje velice dobrou kolaboraci, pokud je potřeba pracovat s více vývojáři (k tomu je potřeba mít založený Google účet).
4. Exportování, importování a následné spuštění daného sešitu je velice jednoduché. Jelikož je již vše uloženo na cloudu, lze to provést pouze pomocí linku.
5. Poskytuje zdarma GPU a TPU ačkoli nutno dodat, že u neplacené verze je tento výkon značně omezen[27].

Alternativou ke Google Colabu může být software jménem Jupyter (obrázek 4.2). Pracuje se s ním velice podobně, ale největším rozdílem je, že celý běží na našem zařízení. Takže je potřeba mít dostatečně výkonný hardware a také je potřeba doinstalovat si všechny potřebné knihovny [27].



Obrázek 4.2: Porovnání iPythonu, Jupyteru a Google Colabu [27].

### 4.3.1 TensorFlow a Keras

TensorFlow je end-to-end open source platforma pro strojové učení a využívání principů umělé inteligence. Obsahuje spoustu knihoven a užitečných algoritmů, které vývojáři velmi pomůžou s

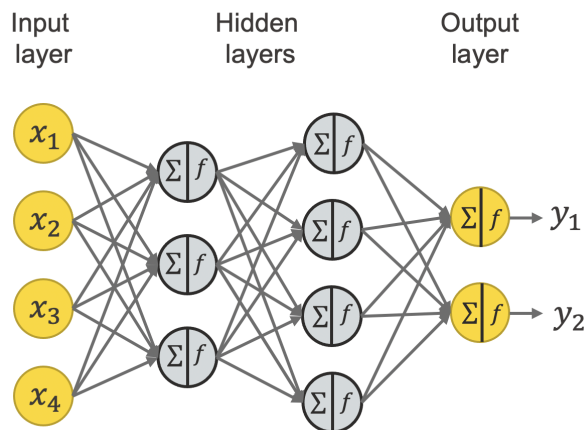
řešením problémů na něž klasické algoritmy nestačí. Tuto platformu využívají i tak velké firmy jako Google, Coca-Cola nebo Intel. Poskytuje také širokou kolekci nástrojů pro tvorbu modulů. Obsahuje data pre-processing, příjem dat, vizualizaci a další. Je taky vytvořena, aby mohla běžet na různých zařízeních a platformách. Tensorflow je řízen celosvětovou komunitou vývojářů. Zaměřuje se na potřeby lidí a podle toho probíhá další vývoj [28].

Keras je výkonná vysokoúrovňová neuronová síť napsaná v programovacím jazyce Python. Je to open-source knihovna, která díky své konstrukci umožňuje rychlé experimentování s hlubokými neuronovými sítěmi a běží na Tensorflow. Prioritou této knihovny je modulárnost, přátelský přístup k uživateli a snadná rozšiřitelnost. Začátek této knihovny sahá pouze pár let zpátky do roku 2015. Standartně k této knihovně přistupujeme přes `tf.keras` (tensorflow), ale modul dokáže fungovat i samostatně. Nestará se o nízkoúrovňové výpočty, ty přenechává jiné knihovně. Alternativou pro Keras může být knihovna PyTorch [29].

## 4.4 Neuronové síť

Počítač dokáže řešit nejrůznější rozsáhlé problémy dle instrukcí, které jsou mu zadány. Avšak pořád je to jen stroj, který nedokáže myslet a neumí zpracovat nic, na co jsme ho nepřipravili. Problémem jsou často také nekompletní, či nějakým způsobem porušená data. Tehdy přichází na scénu neuronové síť (anglicky Artificial Neural Networks).

Tyto síť jsou inspirovaný neurony a jejich propojeními v lidském mozku (obrázek 4.3). Pro zajímavost lidský mozek má 50 až 100 miliard neuronů a každý z nich dokáže vytvořit až 20 tisíc synapsí, což je název pro propojení mezi neurony. Nedávné průzkumy tvrdí, že denně člověk přijde o více než 80 tisíc neuronů, což ročně vychází na asi 31 milionů neuronů [30].



Obrázek 4.3: Ukázka struktury neuronové sítě [31].

S nápadem neuronové sítě přišel americký psycholog Frank Rosenblatt. Tato síť se jmenovala perceptron a je to ta nejjednodušší síť, jakou lze sestavit, jelikož má pouze jeden neuron. Jeho využití je však značně omezené jelikož, lze použít pouze na lineárně separovatelné množiny. Avšak tento objev poskytl různým badatelům základ a umožnil vznik komplexních sítí [32].

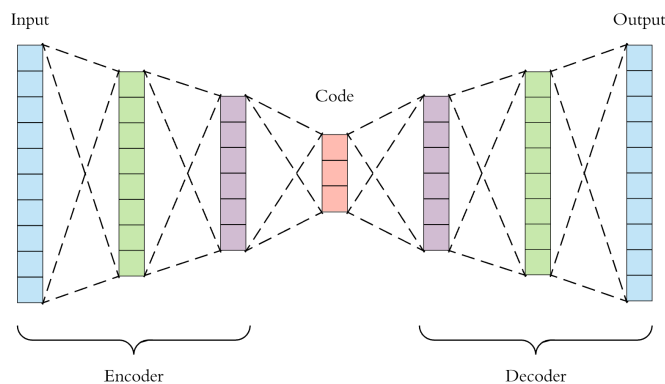
V dnešní době jsou neuronové sítě velice populární téma a vědci neustále přicházejí s novými vylepšeními již existujících sítí. Toto je seznam šesti typů neuronových sítí [33]:

- Feed-Forward Neuronová síť,
- Konvoluční Neuronová síť (CNN),
- Rekurentní Neuronová síť (LSTM/GRU/Attention),
- Transformers,
- Generative Adversarial Networks (GAN),
- Autoenkodéry.

A právě posledně zmíněné autoenkodéry byly použity v této práci o analýze starých dialektů.

#### 4.4.1 Autoenkodéry

Jednoduše řečeno je autoenkodér síť, která se má za úkol natrénovat tak, aby zkopírovala svůj vstup na výstup (obrázek 4.4). Pokud by se autoenkodér dokázal naučit dokonale, moc platné by nám to nebylo, protože by výstupem byla jen kopie vstupu, a proto jsou navrženy tak, aby to dokonale nedokázaly. Většinou jsou omezeny tak, že mohou kopírovat jen přibližně a pouze vstup podobný trénovacím datům. Tradičně se autoenkodéry používaly pro redukci dimenze dat. V poslední době se hojně využívají v generativním modelování [34].



Obrázek 4.4: Ukázka struktury autoenkodéru [33].



Autoenkodér se skládá ze tří hlavních částí:

1. Enkodér - má za úkol komprimovat data do zakódované reprezentace, většinou sníží dimenzi dat.
2. Bottleneck - obsahuje komprimované reprezentace, tudíž se dá říct, že jde o stěžejní vrstvu sítě.
3. Dekodér - pomáhá síti dekomprimovat data a zrekonstruovat je nazpět ze zakódované podoby. Následně se výsledek porovná se vstupem [35].

## Kapitola 5

# Síťová analýza

### 5.1 Centrality

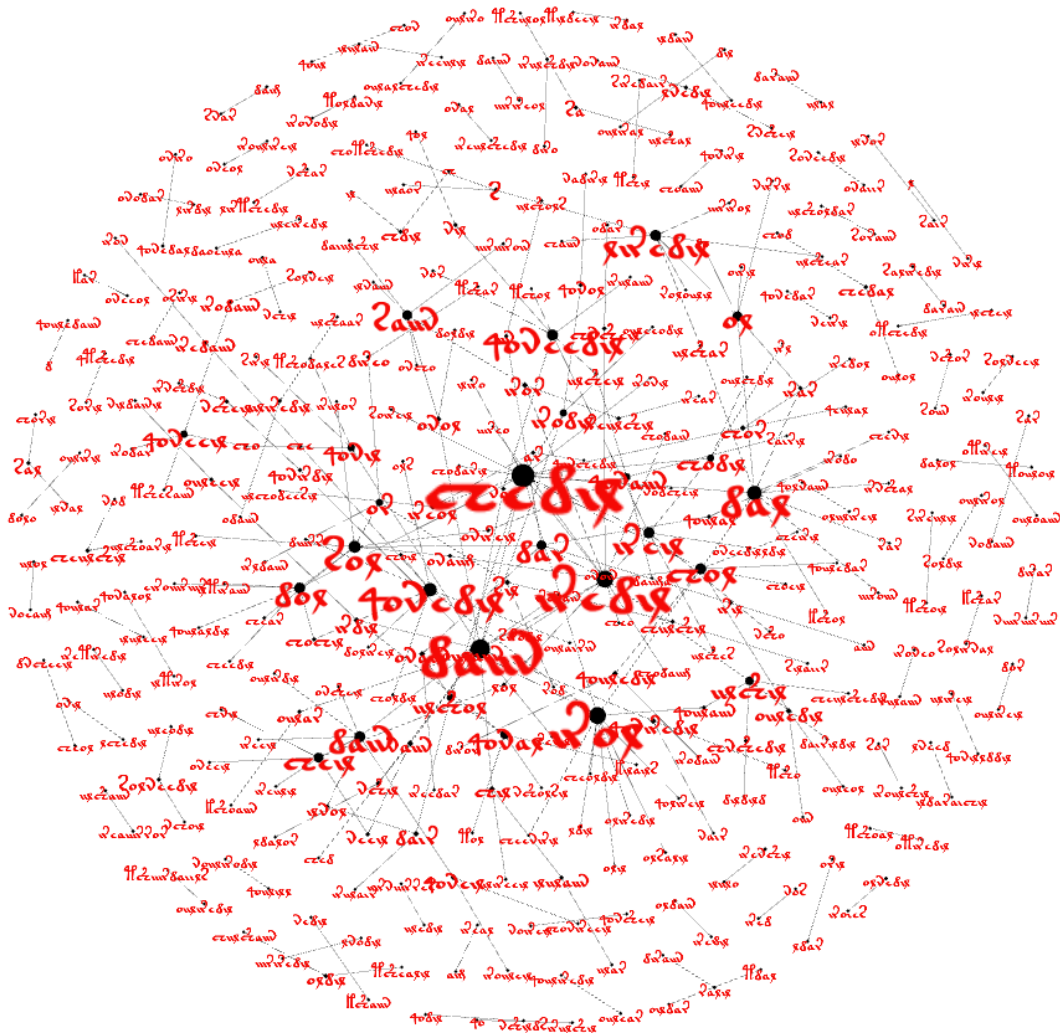
Základem této části je CSV soubor obsahující přepsaná slova Voynichova rukopisu do latinky. Tento soubor je v rámci preprocessingu nahrán a data upravena do formátu, se kterým pak bude pracováno. S daty je pracováno jako se sítí postupně na sebe navazujících slov. Pro každý takový vrchol v síti lze vypočítat tzv. centralitu. Tato hodnota udává, jak významný je tento vrchol v síti. Jsou rozlišeny čtyři hlavní druhy:

1. Degree centrality - nejjednodušší centralita, spočítá se na základě počtu hran jdoucích z daného vrcholu,
2. Betweenness centrality - udává kolikrát leží vrchol na nejkratší cestě mezi ostatními vrcholy v síti,
3. Closeness centrality - vypočítá nejkratší vzdálenost mezi všemi vrcholy a následně přiřadí hodnotu podle sumy všech nejkratších cest,
4. Eigenvectors centrality - podobně jako degree centralita, měří závislost vrcholů, podle počtu hran k ostatním vrcholům v síti. Navíc bere v potaz, jak dobře je tento vrchol připojen, kolik hran toto spojení má atd. [36].

Cílem bylo vytvořit nástroj umožňující spočítat jednotlivé centrality pro daný počet slov od dané pozici v textu. Pro počítání centralit a následné zobrazování je použita knihovna nx, která toto umožňuje. Nevýhodou této knihovny je, že nemá žádný layout, který by při velkém množství vrcholů a hran dokázal přehledně zobrazit data. Z tohoto důvodu vlastní skript obsahuje kód, jenž umožňuje export do csv nebo gexf souboru. Tento soubor si lze snadno importovat do programu Gephi, jenž umí velice dobře vizualizovat síťová data.

Aby bylo možné každému vrcholu přiřadit patřičný znak Voynichova textu, je potřeba si doinstalovat patřičný font. U tohoto kroku je potřeba dát si pozor a nainstalovat ho pro všechny uživatele.

Pokud by tak nebylo učiněno, font se nainstaluje do "APPDATA"daného uživatele. Tento font by se sice v adresáři s nainstalovanými fonty nacházel, nicméně v gephi by nebyl viditelný, takže by ho nebylo možné na síť aplikovat.



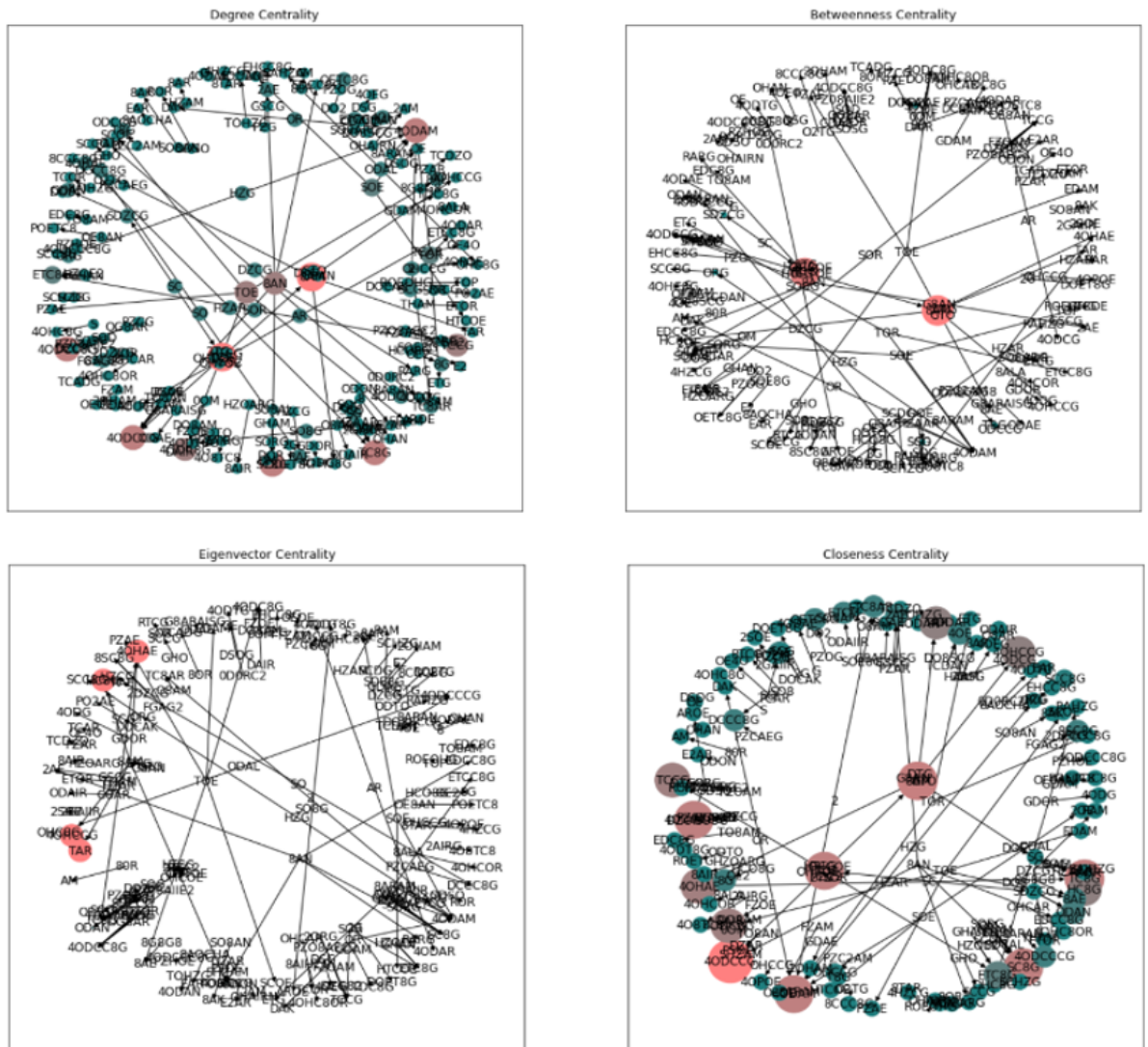
Obrázek 5.1: Síťové zobrazení okna o velikosti 300 začínající na pozici 16 558.

Na obrázku 5.1 je vidět vizualizace části sítě Voynichova rukopisu. Konkrétně tří set po sobě jdoucích slov začínající náhodně zhruba uprostřed rukopisu. Tato síť je vyobrazená ve vizualizačním softwaru Gephi, který kromě toho, že zvětší uzel na základě stupně, dokáže zvětšit i jeho popis, což značně zvýší přehlednost. U této vizualizace bylo použito rozložení jménem "fruchterman reingold".

Na obrázku 5.2 lze vidět výstřížek z vlastního Google Colab sešitu. Tento software umožňuje přidat posuvníky, jenž jsou zde využity. První posuvník slouží k zadání velikosti úseku o minimální délce 10 a maximální délce 500 slov. Druhý posuvník slouží k určení pozice v textu. Tyto úseky textu

```
[55] window_size = 150 #@param (type:"slider", min:10, max:500, step:1)
position = 16000 #@param (type:"slider", min:0, max:33244, step:1)
draw_networks(window_size, position, all_words_voynich)
```

window\_size:  150  
 position:  16000



Obrázek 5.2: Ukázka z Google Colab sešitu.

budou v této práci nazývány okny. Minimální pozice je 0 což znamená, že okno začne na úplném počátku textu a maximum je celkový počet slov - velikost okna. Na základě těchto parametrů se sestrojí a vykreslí síť (pro každou centralitu jedna síť). Knihovna nx, která je zde použita pro vykreslování sítí, obsahuje různé rozložení, ale žádné nedokázalo zobrazit větší síť přehledně.

## 5.2 Mapy centralit

Dalším pokusem je vypočtení tzv. "centrality map". Tato metoda využívá centralit, o kterých byla řeč v přechozí kapitole. Centralita se počítá z různých pozic a různě velikých úseků.

Na obrázku 5.3 je vidět grafické znázornění mapy. Osa X představuje pozice v celém textu (v našem případě 32 000 slov) a osa Y Velikost okna. Pro každý bod je třeba spočítat všechny 4 centrality. Má-li text délku 32 000 a okno o maximální velikosti 500, musí být tento výpočet prováděn pro 16 000 000 bodů. Jelikož je tato metoda časově náročnější, je zde využít pythonovský balíček "multiprocessing". Tento balíček umožňuje spouštět funkci paralelně. Každému procesu je třeba přiřadit, které části mapy má spočítat. Tento způsob velmi zkrátí celkový výpočetní čas. Takto to však není možné provést na Google Colabu, protože v neplacené verzi jsou k dispozici pouze dva procesy, takže by to nebylo příliš přínosné.

### 5.2.1 Spuštění skriptů

Finální adresář obsahuje script, který dokáže paralelně vypočíst mapu centralit pro všechny 4 již dříve zmíněné centrality ve vstupní síti. Tento skript se spouští následujícím způsobem:

---

```
py centrality_map_with_multiprocessing.py -i <inputFile> -o <outputDir> -m <maxWindowSize> -l <length>
```

---

1. -i (povinný) - cesta k vstupnímu souboru,
2. -o (povinný) - cesta k výstupnímu adresáři,
3. -m (volitelný) - velikost okna (výchozí hodnota je 100),
4. -l (volitelný) - pokud uživatel nechce mapu počítat pro celou síť může zadat jen prvních n slov (výchozí hodnota je délka celké sítě).

Pro spuštění tohoto skriptu je potřeba mít nainstalovaný Python a knihovny, jež jsou ve skriptu použity. Potřebné knihovny knihovny: numpy, nx, random, multiprocessing, time, dask.distributed, ast

Vstupní soubor musí být txt soubor a musí mít následující formát:

---

```
['FGAG2', 'GDAE', 'AR', 'GHAM', 'SOE', 'SORG',.....]
```

---

Výsledky se uloží do CSV souborů na již specifikovaném místě. Pro každou centralitu jeden soubor, tzn. vytvoří se 4 soubory. Tyto soubory pak lze načíst a zobrazit pomocí druhého skriptu. Vygenerované soubory nesmí být přejmenovány.

Skript pro zobrazení výsledků se spouští následovně:

---

```
py load_and_display_map.py -i <inputFolder> -c <colorMap>
```

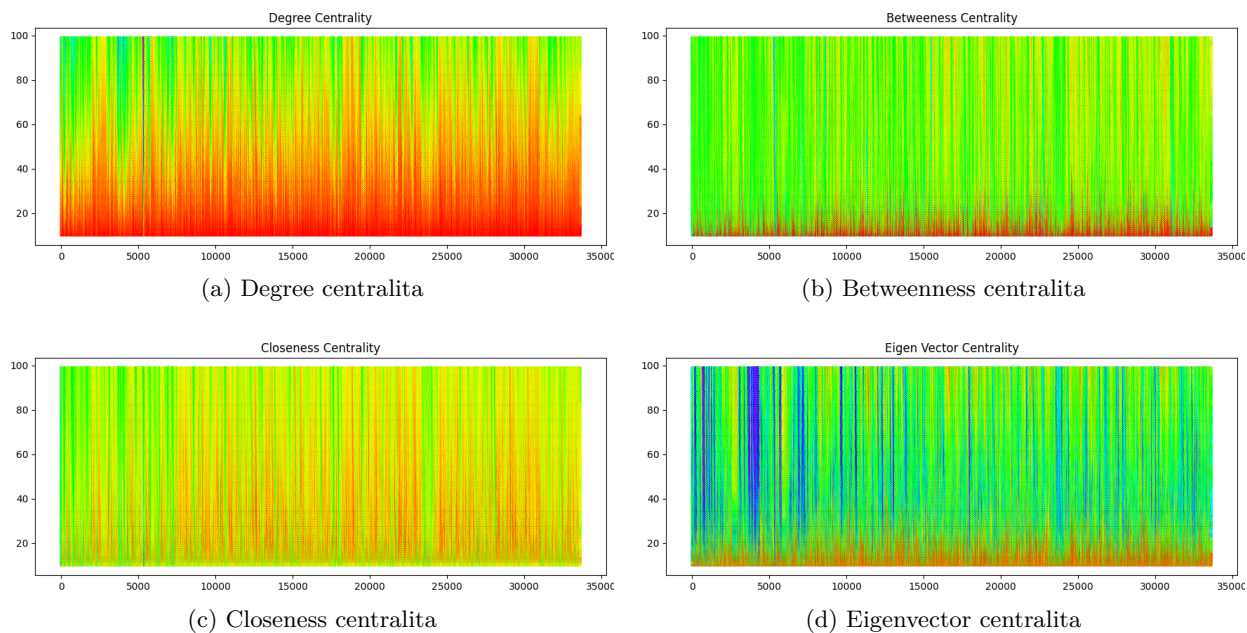
---

1. -i (povinný) - adresář obsahující soubory s výsledky předešlého skriptu,
2. -c (volitelný) - colormap (výchozí hsv, možnosti jsou vypsané na oficiálních webových stránkách knihovny matplotlib).

Potřebné knihovny pro spuštění skriptu: pandas, matplotlib.

### 5.3 Porovnání map centralit

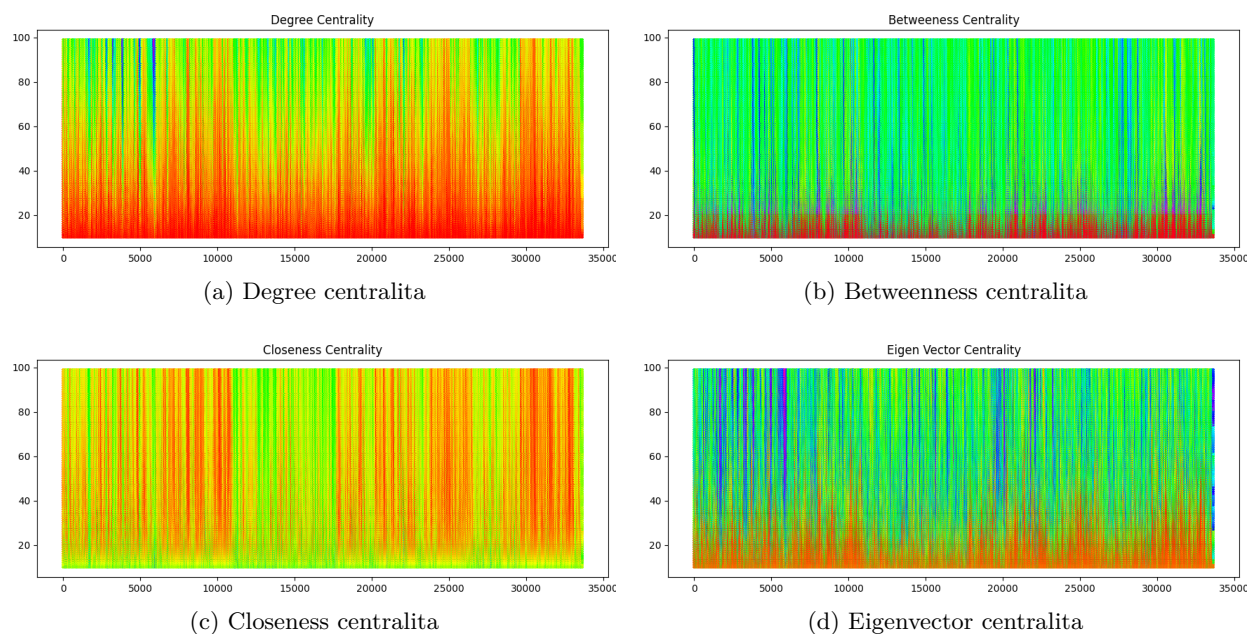
Na obrázku 5.3 lze vidět výsledky centralit pro nejprodávanější knihu světa a sice Bibli. Konkrétně tento překlad obsahuje přes 700 tisíc slov. To je příliš mnoho, aby byla tato analýza prováděna v celém rozsahu.



Obrázek 5.3: Mapy centralit pro Bibli.



Výpočetně by to trvalo poměrně dlouho, ale hlavní problém by byla nečitelnost grafu, způsobená velkým množstvím hodnot. Z toho důvodu bude z Bible vzato jen prvních  $N$  slov, přičemž  $N$  je rovno délce Voynichova rukopisu.



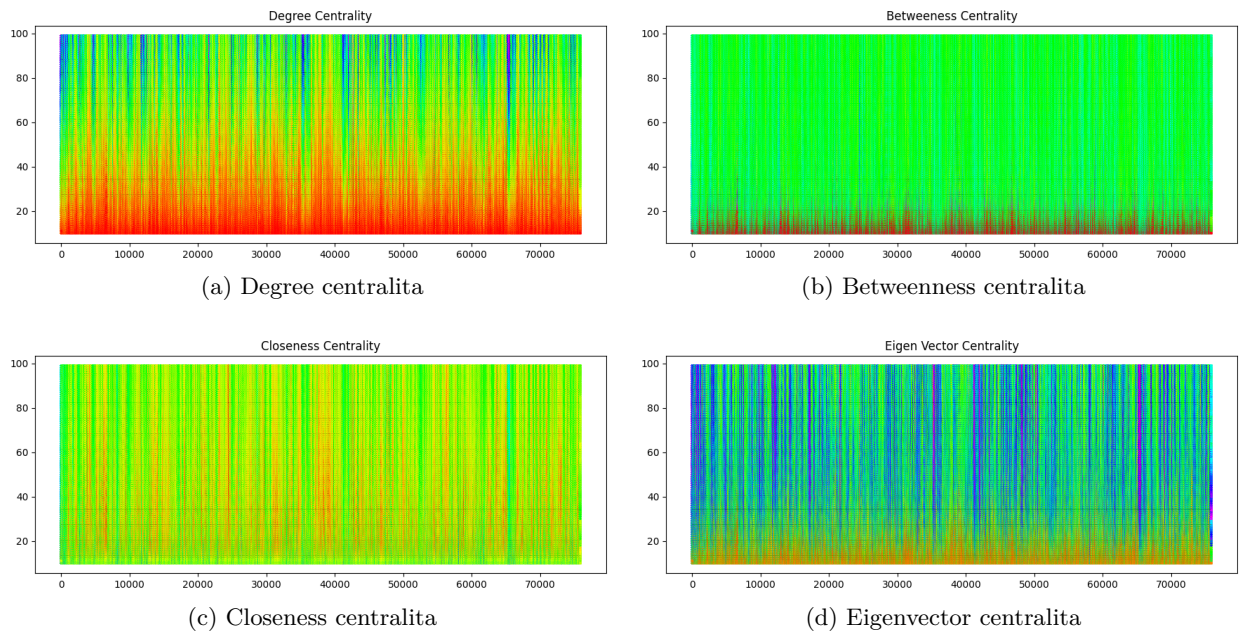
Obrázek 5.4: Mapy centralit pro Voynichův rukpis.

Na obrázku 5.4 jsou vidět výstupy pro literární dílo Ernesta Hemingwaye *Stařec a moře*. Tato Nobelovou cenou oceněná novela obsahuje něco přes 70 tisíc slov a algoritmus byl aplikován na celý její rozsah.

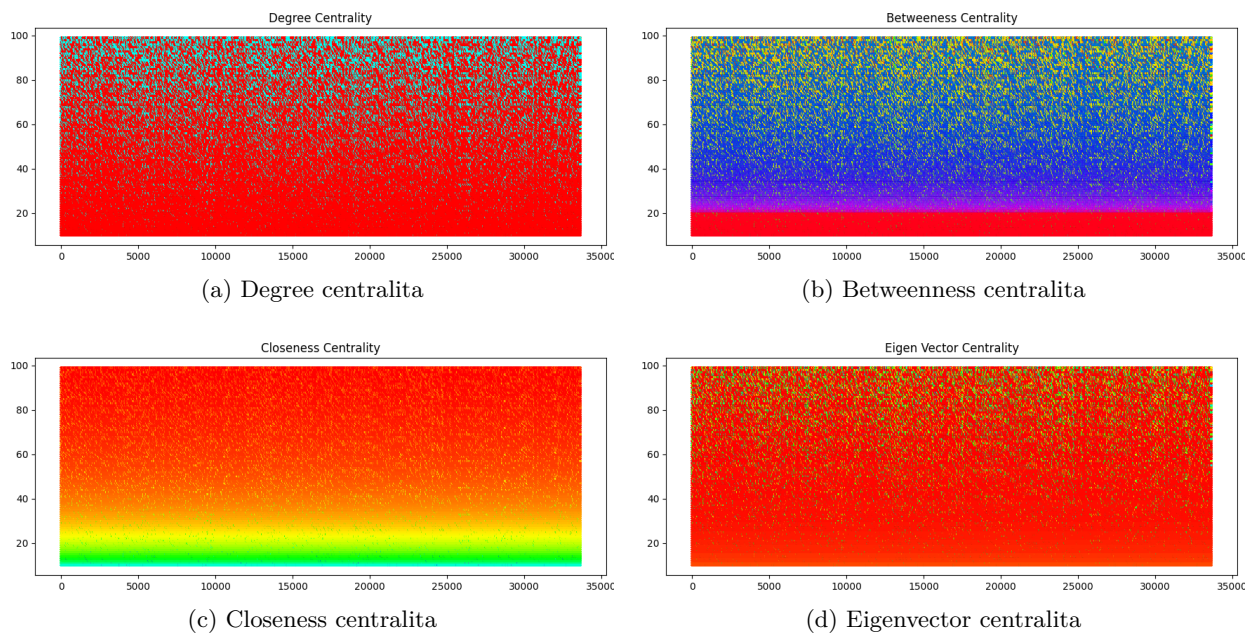
Pro lepší porovnání byl vygenerován zcela náhodný text (obrázek 5.6). Byl vytvořen tak, že byla vybrána veškerá unikátní slova Voynichova rukopisu. Všechna tato slova měla stejnou pravděpodobnost na zvolení. Tato slova byla do pole přidávána, dokud nebylo dosaženo délky celkového počtu slov Voynichova rukopisu. Čili text byl zcela náhodný, ale skládal se ze slov samotného rukopisu. Na takto vygenerovaný text se pak aplikovaly stejné metody, jako na originální text Voynichova rukopisu.

## 5.4 Zhodnocení síťové analýzy

Na obrázku 5.7 lze vidět přibližnou mapu centralit. Napravo pro Voynichův rukopis a nalevo pro náhodný text složený ze slov Voynichova rukopisu. Ačkoliv to vypadá, že se barevné schéma liší, byla pro oba tyto vzorky použita stejná colormap. Avšak v náhodném textu není taková barevná diverzita, jelikož se tam nenachází takové shluky jako v originálním textu a hodnoty nejsou tak

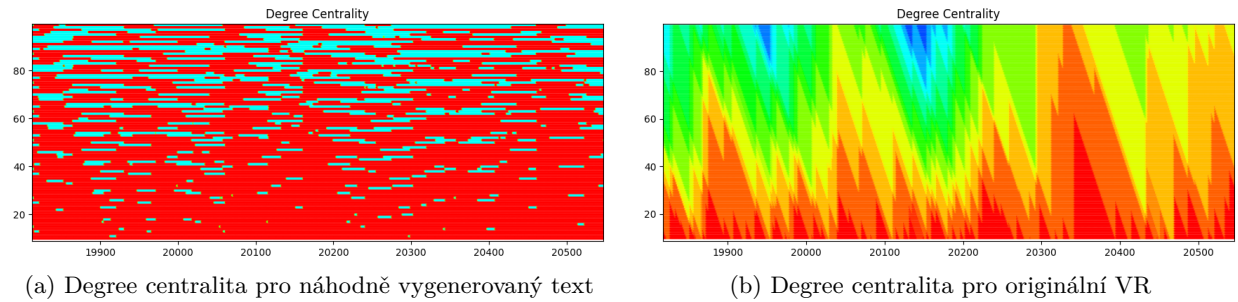


Obrázek 5.5: Mapy centralit pro dílo Stařec a moře.



Obrázek 5.6: Mapy centralit pro náhodně vybraná slova voynichova rukopisu.





Obrázek 5.7: Zvětšení vybraných map centralit

rozmanité. V přiblíženém tvaru jde hezky vidět vzor, který Voynichův rukopis opisuje a je na první pohled jasné, jak se od sebe navzájem tyto 2 vzorky liší.

Na obrázcích 5.3, 5.4 a 5.5 lze vidět, že jsou si velice podobné a sdílejí spolu velice podobné vzory, jenž jsou typické pro každé psané umělecké dílo. Naproti tomu náhodný text (obrázek 5.6) obsahuje pouze šum, který sice má jakýsi vzor, ten však není shodný se vzory, jaké tvoří mapy smysluplných textů. Z tohoto pozorování vyplývá, že text Voynichova rukopisu není náhodným pokusem a sdílí vlastnosti přirozeného jazyka.

## Kapitola 6

# Analýza za pomoci neuronové síť

### 6.1 Vstup síť

Druhá část analýzy spočívá v aplikování neuronových sítí na vektorových datech reprezentujících abecedu Voynichova rukopisu. K tomuto účelu bude využita Pythonová knihovna Keras a pro vizualizaci opět knihovna matplotlib.

Importování fontů do Google colabu může být kvůli omezeným právům nesnadná úloha, ale toto je jeden ze způsobů, jak toho docílit.

---

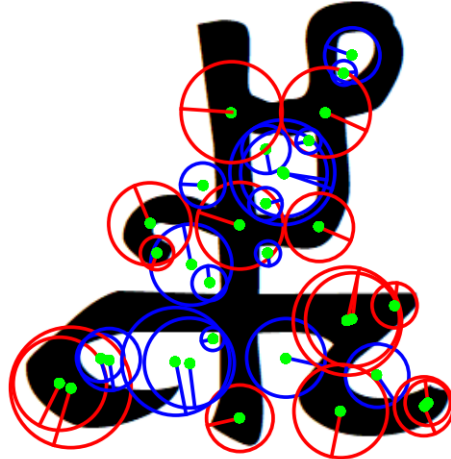
```
import matplotlib.font_manager as fm
!wget "https://fontlibrary.org/assets/downloads/voynich/302
      bb5408e0dc1801365b27bed99357d/voynich.zip"
!unzip "/content/voynich"
!mv voynich-1.23-webfont.ttf /usr/share/fonts/truetype/
font_path = '/usr/share/fonts/truetype/voynich-1.23-webfont.ttf'
voynich_font = fm.FontProperties(fname=font_path)
```

---

Když je na první pozici v řádku vykřičník, znamená to, že se jedná o linuxový příkaz. Nejprve je potřeba stáhnout patřičný font. Jelikož je ve formátu zip, tak je ho třeba nejprve odzipovat. Následně je font přemístěn do patřičného adresáře. Tuto cestu, kde byl font přemístěn, je třeba si uložit. Dále je importován "font manager" od matplotlibu a pomocí jeho funkce "FontProperties" je font uložen do proměnné, která je později přiřazena na potřebné místo. Pokud by tento krok s Matplotlib managerem nebyl proveden, font by byl viditelný v linuxovém systému, ale nebylo by možné ho využít při plotování grafu a právě tam je potřeba.

### 6.1.1 Image Corresponding Points

Image Corresponding Points je metoda poskytnutá softwarem Mathematica, která na vstupním obrázku najde hodnoty, které uzná za důležité. Dívá se na vlastnosti, jako barva, kontrast nebo sklon [37]. Body vygenerované z abecedy EVA pomocí této metody budou sloužit jako vstup sítě. každé písmeno však vypadá jinak a má jiný počet těchto bodů. Na obrázku 6.1 lze vidět body jenž tato metoda vygenerovala pro jedno písmeno.



Obrázek 6.1: Ukázka Image Corresponding Points na písmenu [38].

### 6.1.2 Vstupní množina

Jako vstupní data používáme písmena Voynichovy abecedy ve vektorové formě v následujícím tvaru:

---

```
{{{161.35333251953125, 145.65086364746094}, 5.317816734313965,  
-3.0451502799987793, -1}, {{189.20054626464844, 146.8954620361328},  
5.636270523071289, 0.5661166349994105, -1},  
{{177.7019805908203, 173.17710876464844}, 4.002931118011475,  
0.509308163319723, -1}, {{176.69248962402344, 151.82997131347656}, ...
```

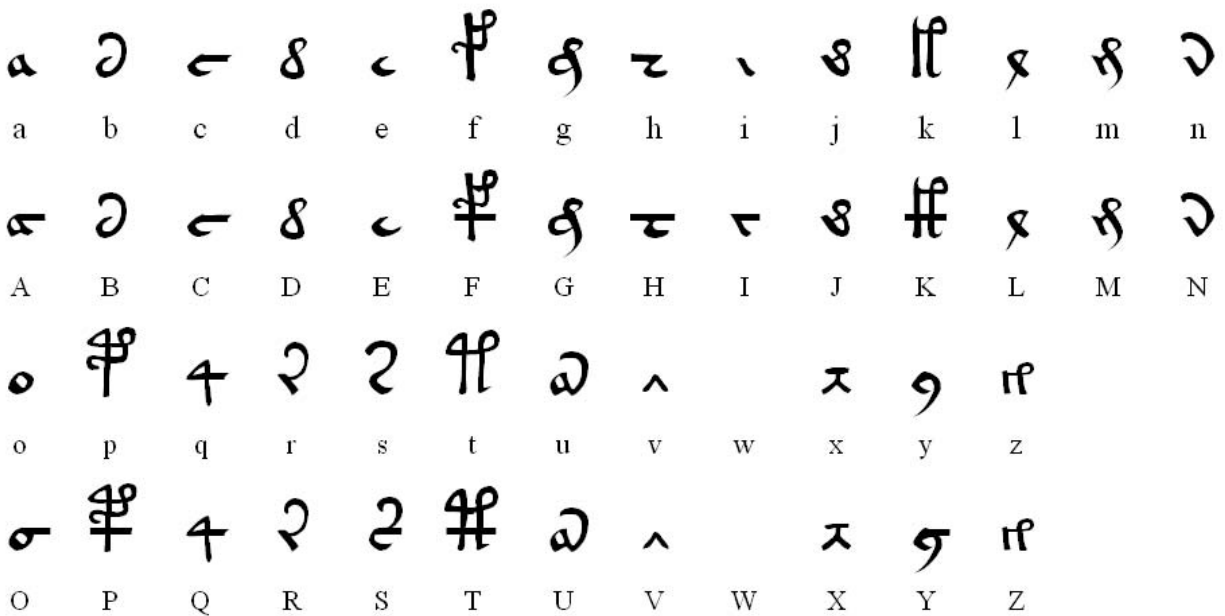
---

Tato data jsou v rámci preprocessingu upravena a převedena do tvaru 256 x 1. Je nutné data uložit do "ndarray", což je speciální pole od Numpy, jelikož to knihovna Keras vyžaduje. Data jsou vygenerována s maximální délkou 256, avšak ne všechna písmena obsahují tolik bodů, proto jsou

chybějící místa do 256 doplněna nulami. Pokud není řečeno jinak, výsledky, popis a vizualizace v této práci se vztahují k těmto vstupním datům.

Jak bylo řečeno v kapitole 6.1.1, každé písmeno může mít různý počet Corresponding Points. Písmeno z Voynichovy abecedy s nejmenším počtem těchto bodů jich má pouze okolo 200 a písmeno s největším počtem jich má přes 1030. V druhém experimentu bude pracováno s těmito daty, která nejsou omezena. Pole bude doplněno nulami až do nejnižší mocniny dvou, která je vyšší než 1030, což je 2048.

Nakonec jsou data normalizována.



Obrázek 6.2: Písmena abecedy namapovaná na písmena Voynichova rukopisu [39].

Na obrázku 6.2 lze vidět písmena abecedy EVA, která je zde využita pro mapování a zobrazení písmen, při vizualizaci.

## 6.2 Sestrojení neuronové sítě

Jako první je sestaven nejprimitivnější autoenkodér (tabulka 6.1). Obsahuje pouze vstupní vrstvu, zploštění, pak vrstvu se 3 neurony (bottleneck), další vrstvu tentokrát s 256 neurony a následně reshape, který vrátí hodnoty do původního tvaru. Validační chyba se po 10 000 epochách pohybuje okolo 0.03. Tudíž je jasné, že bude potřeba neuronové sítě přidat více vrstev.

V druhém pokusu jsme síť obohatili o více vrstev (tabulka 6.2). Tato sestavená síť již dokázala původní data zrekonstruovat s validační chybou okolo 0.008. To je již dostačující hodnota pro náš

Tabulka 6.1: Primitivní autoenkodér

Vrstva	Tvar	funkce
Vstupní vrstva	(256,1)	
Zploštění	(256)	
Dense vrstva	(3)	sigmoid
Dense vrstva	(256)	relu
Reshape	(256, 1)	

experiment, nicméně byl proveden ještě jeden pokus, ve kterém bylo využito konvolučních vrstev. V této síti bylo dosaženo podobné úspěšnosti po menším počtu epoch. A právě tato konvoluční síť byla v této práci pro analýzu zvolena a bude více popsána v následující kapitole.

Tabulka 6.2: Vícevrstvý autoenkodér

Vrstva	Tvar	funkce
Vstupní vrstva	(256,1)	
Dense vrstva	(256, 64)	relu
LeakyRelu	(256, 64)	
Dense vrstva	(256, 8)	relu
Dropout	(256, 8)	
Zploštění	(256)	
Dense vrstva	(3)	sigmoid
Dense vrstva	(2048)	relu
Reshape	(256, 8)	
Dense vrstva	(256, 64)	relu
Dropout	(256, 64)	
LeakyRelu	(256, 64)	
Dense vrstva	(256, 1)	relu

### 6.2.1 Popis sestrojené neuronové sítě

Vlastní neuronová síť se skládá z 15 skrytých vrstev a vstupní a výstupní vrstvy. Tudíž tato síť patří do kategorie hlubokých neuronových sítí. Níže je vidět část neuronové sítě použité pro analýzu vektorových dat.

---

```

input_layer = Input(shape=(256,1))
encoder_input = Conv1D(64, 2, activation='relu', padding='same', name="1.
    Conv1d_encoder")(input_layer)
encoder_input = MaxPooling1D(2)(encoder_input)
...
decoder_input = Dense(3, activation='sigmoid', name="bottleneck")(encoder_input)

```

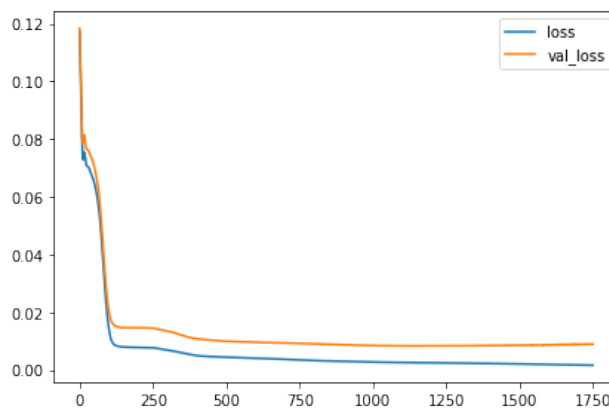
```

...
decoder_input = UpSampling1D(2)(decoder_input)
decoder_output = Conv1D(1, 2, activation='relu', padding='same', , name="3.
    Conv1d_decoder")(decoder_input)
model = keras.models.Model(input_layer, decoder_output)
model.compile(loss='mse', optimizer='adamax')

```

---

Na obrázku 6.3 lze vidět křivku učení sestrojené konvoluční sítě. Jak je u většiny ANN typické, na počátku dochází během několika epoch k velkému zlepšení, ale postupně se stabilizuje. Pokud by nebyl výpočet zastaven, postupně by docházelo k mírnému zvyšování hodnoty validační chyby, tudíž ke zhoršení kvality výsledku. A proto je využito tzv. "callbacku" jménem "EarlyStopping", který učení zastaví, jakmile k tomu začne docházet.



Obrázek 6.3: Křivka učení.

### 6.2.1.1 Dense

Dense vrstva je nejobvyklejší a nejčastěji používaná vrstva. Tato vrstva provádí následující operaci:

---

```
výstup = aktivace(dot(vstup, jádro) + bias)
```

---

1. Vstup - reprezentuje vstupní data,
2. Jádro - reprezentuje váhu dat,
3. Dot - reprezentuje skalární součin všech vstupů a jejich odpovídajících vah,
4. Bias - představuje hodnotu používanou ve strojovém učení k optimalizaci modelu,

5. Aktivace - představuje aktivační funkci.

Výstupní tvar bude ovlivněn počtem neuronů přidělených této vrstvě [40].

### 6.2.1.2 Conv1D

Tato vrstva umožňuje vytvořit konvoluční jádro, které konvoluje přes jedinou dimenzi a vytvoří tenzor výstupů [41].

### 6.2.1.3 MaxPooling1D

MaxPooling1D je vrstva, která vezme maximální hodnotu v daném prostorovém okně, čímž převzorkuje vstupní data. Výstup je ve tvaru: (Tvar vstupu - velikost poolu + 1) / počet kroků [42].

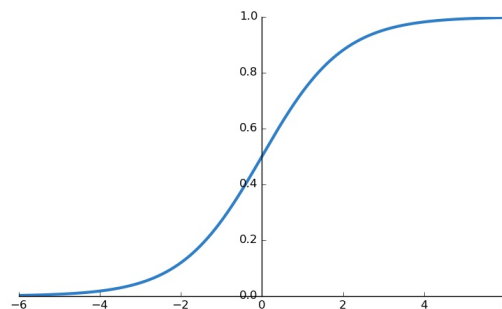
### 6.2.1.4 Sigmoid

Funkce Sigmoid (někdy též Logistická funkce) patří k těm nejvíce používaným aktivačním funkcím. Na obrázku 6.4 lze vidět její vyobrazená v grafu. Rovnice této funkce vypadá následovně:

$$\frac{1}{1 + e^{-x}} \quad (6.1)$$

Vlastnosti funkce Sigmoid:

- funkce má tvar písmene S,
- výstup funkce je číslo v rozmezí 0 až 1 se středem 0.5,
- funkce je monotónní, avšak její derivace monotónní není,
- funkce je diferencovatelná, a proto dokážeme zjistit sklon křivky v jakýchkoliv dvou bodech.



Obrázek 6.4: Vizualizace funkce Sigmoid [43].

Tato funkce byla představena neuronovým sítím v devadesátých letech minulého století a nahradila funkci jménem Step. Byla to klíčová změna pro ANN architekturu. Hlavním důvodem bylo, že Step funkce nemá žádný gradient pro práci s Gradient Descentem, zatímco Sigmoid funkce má definovanou nenulovou derivaci na celé ploše. Toto umožňuje udělat nějaký pokrok při každém kroku tréninku.

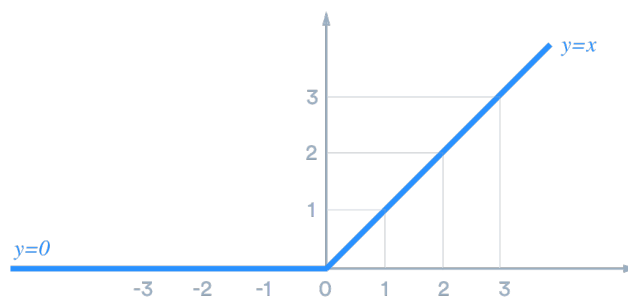
Hlavní problémy:

1. Mizící gradient - Jakmile se vstupy stanou malými nebo naopak velkými, funkce saturuje na 1 nebo 0 s derivací velice blízko 0. Z toho důvodu nemá skoro žádný gradient, který by šel sítí propagovat nazpět, takže na ostatní vrstvy takřka nic nezbude.
2. Výpočetně náročné (exponenciální složitost).
3. Výstup není vycentrovaný na 0 [44].

### 6.2.1.5 Relu

Relu (Rectified Linear Unit) se v rámci hloubkového učení používá nejčastěji. Tato funkce vrací 0 pokud je vstup záporný, ale pro jakýkoliv pozitivní vstup tuto hodnotu vrátí nazpět (obrázek 6.5). Funkce je definovaná následovně:

$$\begin{cases} 0 & \text{if } x \leq 0, \\ x & \text{if } x > 0 \end{cases} \quad (6.2)$$



Obrázek 6.5: Vizualizace funkce Relu [45].

1. Funkce je graficky složena ze 2 lineárních částí, které berou v potaz nelinearity. Relu je nelineární, pokud sklon není konstantní. Takže je nelineární okolo 0, ale sklon je vždy buď 0 (pokud je vstup záporný) nebo 1 (pokud je vstup kladný).
2. Funkce Relu je spojitá, avšak není diferencovatelná, jelikož její derivace je 0 pro jakýkoliv záporný vstup.



3. Výstup funkce nemá maximální hodnotu, což pomáhá Gradientu Descentu.
4. Oproti Sigmoid je mnohem rychlejší.

Tato funkce je velmi jednoduchá, ale v neuronových sítích funguje velice dobře. I tato funkce má však jednu chybu. Tento problém se nazývá "Dying Relu". Během tréninku, některé neurony tzv. zemřou, což znamená, že jejich výstup bude pouze 0. Může se dokonce stát, že takhle zemře polovina neuronů celé sítě. Zejména pokud je použita vysoká rychlost učení. Neuron zemře, když se jeho váhy změní tak, že vážený součet všech jeho vstupů je záporný pro veškeré instance v trénovacích datech. Jakmile se to stane, na výstupu budou pouze nuly a neovlivní již sestup gradientu, jelikož gradient funkce je 0, pokud je vstup záporný [44].

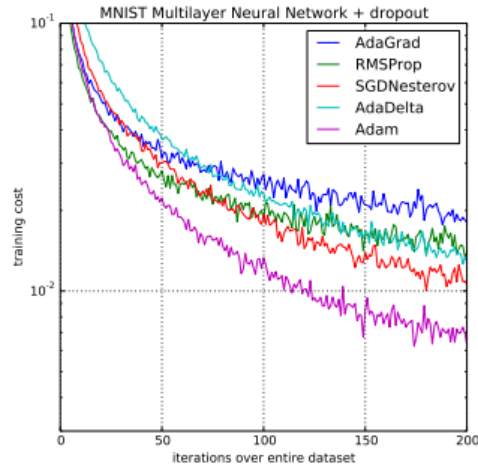
#### 6.2.1.6 Adam

Adam je optimalizační algoritmus, jehož jméno je odvozeno z anglického "Adaptive moment estimation", což můžeme přeložit jako odhad adaptivního momentu. Byl představen Diederikem Kingmanem a Jimmym Baem jako metoda pro stochastickou optimalizaci.

Výhody tohoto algoritmu:

1. Přímá implementace.
2. Malé nároky na paměť.
3. Invariantní k diagonálnímu přeškálování měřítka gradientů.
4. Efektivní při výpočtech.
5. Efektivní pro nestacionární cíle.
6. Vhodný při problému s velkými daty, či při vysokém počtu parametrů.
7. Ideální pro problémy s velmi nebo naopak málo zkreslenými gradienty.
8. Hyperparametry mají intuitivní interpretaci a standardně nevyžadují dlouhé ladění.

Tato metoda vypočítává individuální adaptivní rychlosti učení pro odlišné parametry z odhadů prvního a druhého momentu gradientu. Na obrázku 6.6 lze vidět porovnání optimalizačních algoritmů [46].



Obrázek 6.6: Porovnání optimalizačních algoritmů na datové sadě MNIST [46].

### 6.2.1.7 MSE

MSE neboli mean square error je nejčastěji používaná ztrátová funkce pro regresi. Tato funkce je popsána následujícím vzorcem:

$$L(y, \hat{y}) = \sum_{i=0}^N (y - \hat{y})^2 \quad (6.3)$$

Jednou z vlastností této funkce je, že velké chyby jsou kvadraticky více penalizované než chyby malé [47].

## 6.3 Zobrazení výsledku neuronové sítě

Pro zobrazení výsledku je potřeba dostat hodnoty ze 3 neuronů uprostřed sítě, což je již dříve zmiňovaný bottleneck. Výstup těchto neuronů vypadá takto:

---

```
[0.99984825 0.7195845 0.99998426]
[0.99978524 0.5532664 0.99998236]
[0.9998654 0.7703911 0.99998486]
[0.9998734 0.6558289 0.9999895 ]
[0.9998317 0.73798263 0.99998 ]
[0.99982446 0.690854 0.99998164]
[0.9998753 0.6637878 0.99998903]
[0.9997975 0.6787171 0.9999795 ]
[0.99839956 0.00761168 0.99997985]...
```

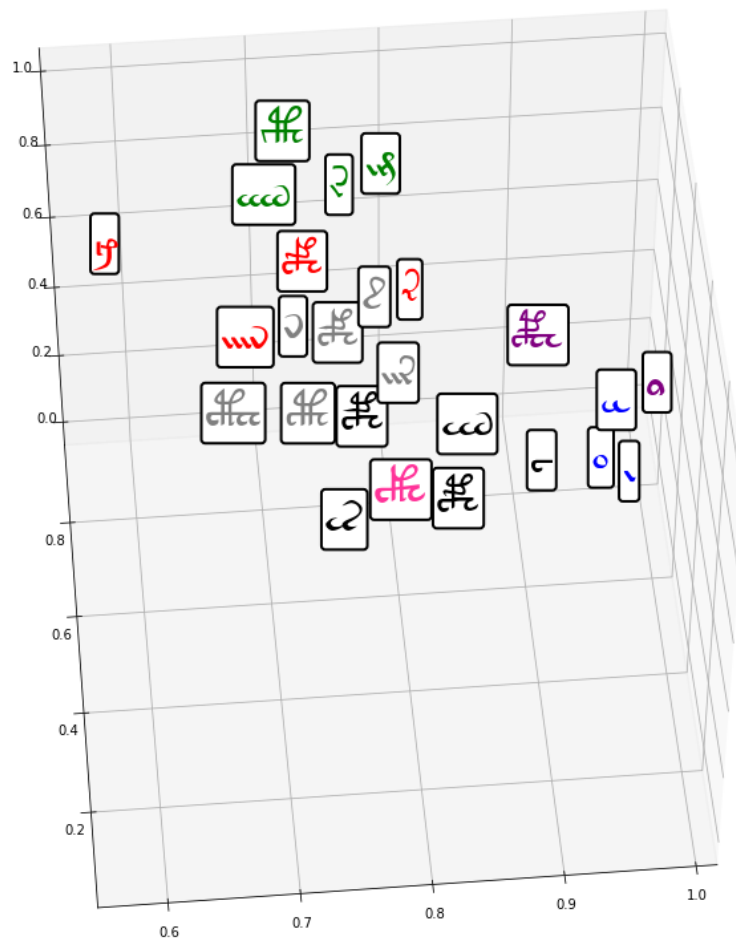
---

Každý řádek tohoto výstupu odpovídá jednomu písmenu abecedy Voynichova rukopisu. Tyto hodnoty jsou použity jako souřadnice pro vizualizaci výsledku. Pro vyobrazení výsledku je opět využita knihovna `matplotlib`, tentokrát pro zobrazení písmen ve 3D prostoru. Jelikož se písmena v grafu překrývala, bylo navíc využito knihovny `adjustText`. Jak už bylo výše zmíněno, Google colab obsahuje pouze některé knihovny a tuto knihovnu neobsahuje. Nicméně i na to Google colab myslí a následujícím jednoduchým příkazem ji lze doinstalovat.

---

```
!pip install adjustText
```

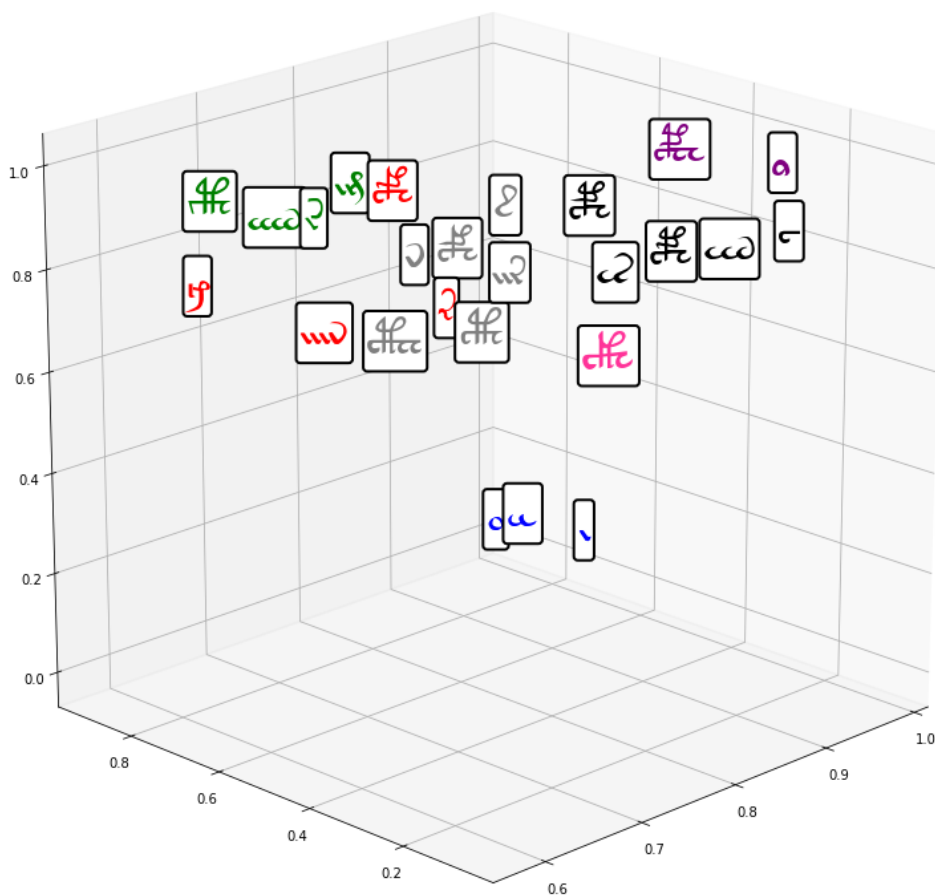
---



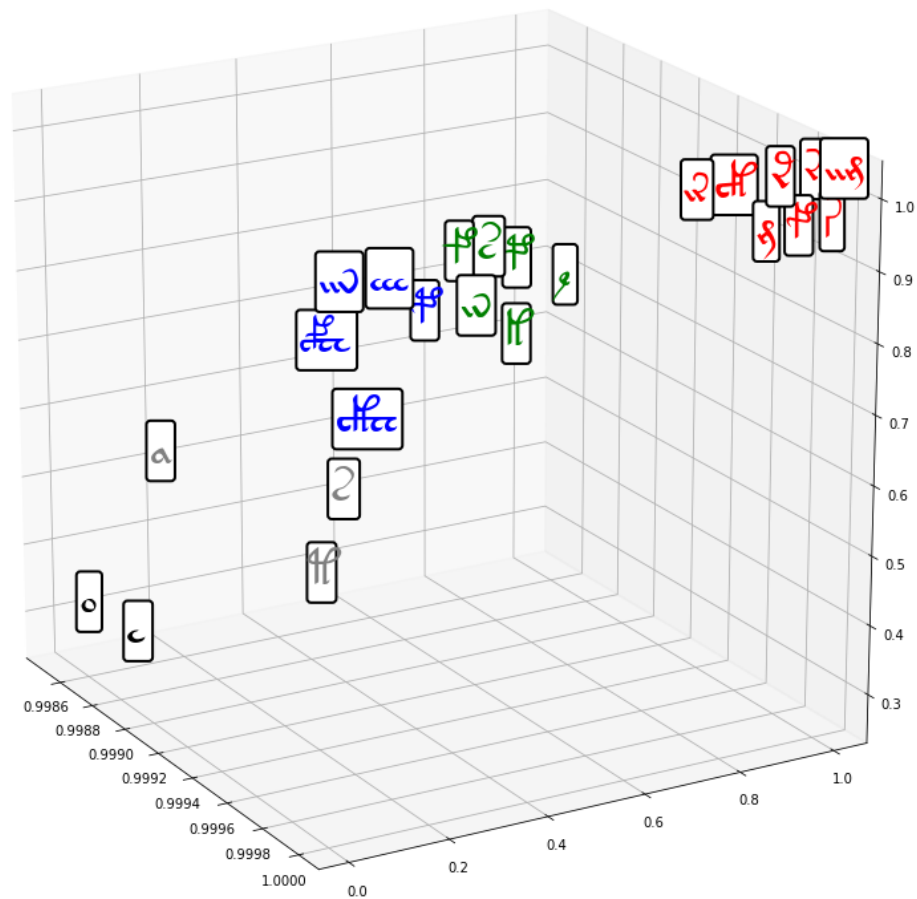
Obrázek 6.7: Vizualizace výsledku autoenkodéru.

Pro vypočtení shluků je využito algoritmu "KMeans". Tento předpřipravený algoritmus obsahuje knihovna "sklearn". Stačí pouze zadat počet shluků a seznam bodů a funkce vrátí pole výstupů. Pro každý vstup jedno číslo určující jeho shluk.

Na obrázku 6.9. je vidět výsledek sestrojené neuronové sítě pro rozšířený neomezený vstup zmíněný v kapitole 6.1. Na tato data by aplikována lehce upravená konvoluční síť jako v předešlém experimentu. V tomto případě se validační chyba pohybovala dokonce kolem 0.006. Kvůli lepší vizualizaci bylo z grafu odebráno písmeno, které odpovídalo písmenu "i" na obrázku 6.2. Toto písmeno bylo velmi vzdálené od ostatních a tvořilo samostatný shluk.



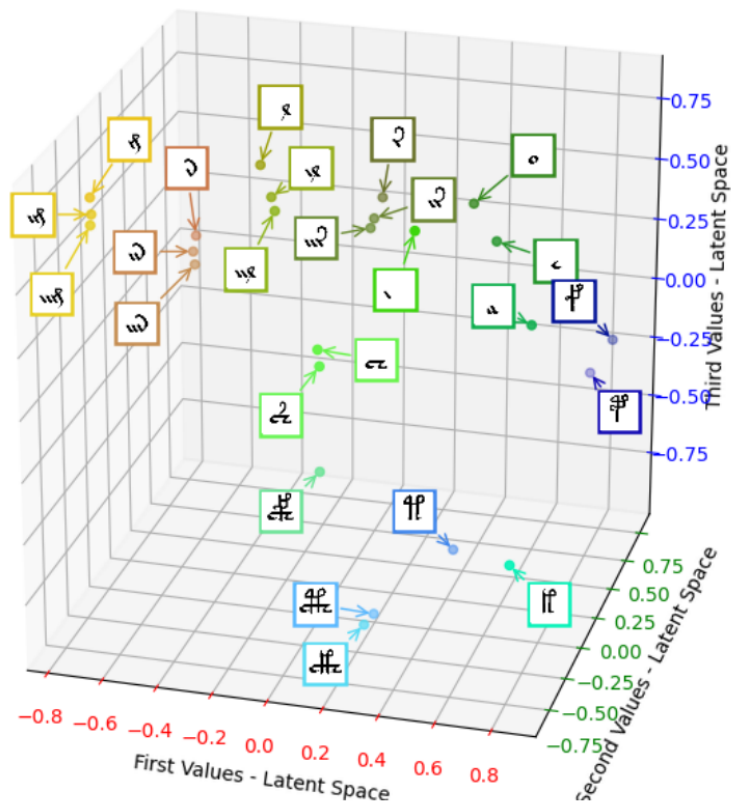
Obrázek 6.8: Vizualizace výsledku autoenkodéru z jiného pozorovacího úhlu.



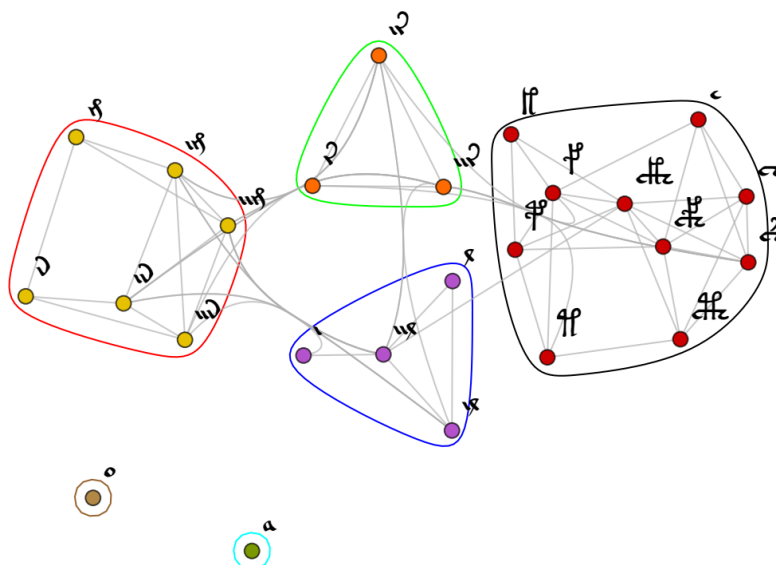
Obrázek 6.9: Vizualizace výsledku autoenkodéru pro neomezená data.

### 6.3.1 Podobnosti písmen porovnání

V této sekci jsou uvedeny výsledky a porovnání pokusů, jež prováděl prof. Zelinka a kol. [38]. Na obrázcích 6.10 a 6.11 lze vidět jejich výsledky. Na obrázku 6.11 lze vidět rozdělení do skupin na základě podobnosti vygenerovaný v softwaru Mathematica. Na obrázku 6.10 je vizualizace výsledku jejich neuronové sítě. Oni však aplikovali neuronovou síť na samotné obrázky Voynichovy abecedy ve 2D formátu. Při porovnání s obrázkem 6.8 lze vidět, že se zde nacházejí podobná uskupení, avšak na obrázku 6.10 jsou více jednoznačné shluky, většinou o velikosti 2-3 písmen.



Obrázek 6.10: Ukázka podobnosti písmen mezi po aplikování neuronové sítě [38].



Obrázek 6.11: Podobnost písmen ve Voynichově abecedě. Písmena jsou seskupená do shluků v závislosti na vzájemné podobnosti [38].

### 6.3.2 Zhodnocení výsledků

Celková validační chyba sestrojené neuronové sítě se pohybovala okolo 0.008. Tuto chybu lze jistě o dost vylepšit, jelikož existuje velké množství možností, jakými způsoby ji lze sestavit. Nicméně v takto nízkých hodnotách to již nehraje příliš velkou roli. Z tohoto důvodu je tato síť považována za dostačující. Z výsledků je patrné, že podobně vypadající znaky jsou zařazeny do stejných shluků. Neplatí to však pro všechny znaky. Některé ač vypadají podobně, můžou mít třeba "kudrlinku" navíc a Corresponding Points pro dané písmeno budou vypadat velmi odlišně. Písmena jsou ve 3D prostoru, proto je důležité dívat se na graf z různých úhlů. Na obrázku 6.8 lze vidět, že písmena ve shluku modré barvy jsou odlišná od všech ostatních zatímco na obrázku 6.7 to poznat nejde.

## Kapitola 7

# Závěr

V této práci byl nejprve popsán samotný Voynichův rukopis. Byla shrnuta historická fakta a základní vlastnosti knihy. Dále byl zhodnocen současný stav na vědecké půdě, co se Voynichova rukopisu týče. Byly ukázány a stručně popsány různorodé pokusy a jejich výsledky, či dedukce. Bylo zjištěno, že naprostá většina vědců a badatelů zabývajících se tímto tématem, se shoduje, že se nejedná o náhodný text, či podvrh.

Následně byly popsány technologie, jež byly využity při samotné analýze. Byly porovnány vlastnosti sítí textů různých rukopisů a náhodného textu. Vlastnosti Voynichova rukopisu se jednoznačně svými znaky od náhodného textu lišily a naopak byly velice podobné klasickým textům psaným přirozeným jazykem. V poslední části byla sestrojena neuronová síť, která byla poté aplikována na písmena Voynichovy abecedy, která byla ve vektorovém formátu, jenž vznikl po aplikování metody Corresponding Image Points. Následně byla všechna písmena vizualizována na 3D grafu a zařazena do shluků. V grafech byla viditelná seskupení písmen na základě podobnosti.

Na světě již bylo provedeno mnoho analýz různého druhu, které zde nebyly zmíněny, ale zatím nikdo nedokázal přijít s uspokojivým vysvětlením co je obsahem, či jaký je původ tohoto záhadného spisu. Je zajímavé sledovat, jak se s časem dešifrovací metody vyvíjí a přístup k dekódování mění. Tato práce byla zaměřena na analýzu textu a jednotlivých písmen, ale rukopis je zvláštní i kvůli kresbám, které se v něm nachází. Byla už provedena různá porovnání obrázků z textu s rostlinami, které známe. Ale i v tomto směru by se daly využít neuronové sítě. Nicméně i přes všechny analýzy je zde možnost, že pokud by se podařilo text dekódovat, nemusel by obsahovat žádné zajímavé informace. K dekódování je však stále daleko a zda-li se to povede, to ukáže až čas.



# Literatura

1. EINSTEIN, Albert. "Eistein citát". [B.r.]. Dostupné také z: <https://www.forbes.com/quotes/190/>.
2. VAŠÍČEK, Arnošt. *Nedobytná šifra. Tajemství Voynichova rukopisu*. Ostrava: Mystery Film, 2011.
3. MANUSCRIPTUM. *VOYNICH MANUSCRIPT*. [B.r.]. Dostupné také z: <https://manuscriptum.ae/product/voynich-manuscript/>. [Online; accessed 15-November-2021].
4. ZANDBERGEN, René. "Description of the Voynich MS". [B.r.]. Dostupné také z: <http://www.voynich.nu/descr.html>. [Online; accessed 20-April-2022].
5. JONES, Josh. *The Mysterious Voynich Manuscript Gets Digitized: Explore the 15th-Century Text That Linguists Code-Breakers Can't Understand*. [B.r.]. Dostupné také z: <https://www.openculture.com/2017/09/the-mysterious-voynich-manuscript-the-15th-century-manuscript-written-in-a-language-linguists-code-breakers-have-never-been-able-to-decipher.html>. [Online; accessed 25-March-2022].
6. BLUMBERG, Naomi. "Voynich manuscript". *Encyclopedia Britannica*. 2016.
7. ACEDO, Luis. A hidden Markov model for the linguistic analysis of the Voynich manuscript. *Mathematical and Computational Applications*. 2019, roč. 24, č. 1, s. 14.
8. JASKIEWICZ, Grzegorz. Analysis of letter frequency distribution in the Voynich manuscript. 2011, s. 250–261.
9. STERNECK, Rachel; POLISH, Annie; BOWERN, Claire. Topic modeling in the Voynich manuscript. *arXiv preprint arXiv:2107.02858*. 2021.
10. AMANCIO, Diego R; ALTMANN, Eduardo G; RYBSKI, Diego; OLIVEIRA JR, Osvaldo N; COSTA, Luciano da F. Probing the statistical properties of unknown texts: application to the Voynich manuscript. *PLoS One*. 2013, roč. 8, č. 7, e67310.
11. HROMADA, Daniel Devatman. Can Evolutionary Computation Help us to Crib the Voynich Manuscript? *arXiv preprint arXiv:2107.05381*. 2021.

12. LAYFIELD, Colin; PLAS, Lonneke van der; ROSNER, Michael; ABELA, John. Word Probability Findings in the Voynich Manuscript. In: *Proceedings of LT4HALA 2020-1st Workshop on Language Technologies for Historical and Ancient Languages*. 2020, s. 74–78.
13. GLADYŠEVA, Alisa. VOYNICH MANUSCRIPT-ANALYSIS OF THE CODIFICATION ALGORITHM WITH THE ENCRYPTION METHODS KNOWN IN THE MEDIEVAL TIME AND RESULTS OF THE MARGINALIES THAT WERE NOT ENCRYPTED. 2020.
14. LANDINI, Gabriel. Evidence of linguistic structure in the Voynich manuscript using spectral analysis. *Cryptologia*. 2001, roč. 25, č. 4, s. 275–295.
15. "What is Crowdsourcing?" [B.r.]. Dostupné také z: <https://crowdsourcingweek.com/what-is-crowdsourcing/>.
16. LIVESCAULT, Jonathan. "What is Crowdsourcing? (in 2022)". [B.r.]. Dostupné také z: <https://www.braineet.com/blog/crowdsourcing>.
17. "What is the Zooniverse?" [B.r.]. Dostupné také z: <https://www.zooniverse.org/about>.
18. "Voynich Manuscript". [B.r.]. Dostupné také z: <https://www.zooniverse.org/projects/ivka/voynich-manuscript/classify>.
19. OPLETALOVÁ, Bc. Aneta. "Analýza Voynichova rukopisu z pohledu crowdsourcingu". [B.r.].
20. "7 Popular Software Programs Written in Python". [B.r.]. Dostupné také z: <https://codeinstitute.net/global/blog/7-popular-software-programs-written-in-python/>.
21. PRAMANICK, Sohom. "History of Python". 2022. Dostupné také z: <https://www.geeksforgeeks.org/history-of-python/>.
22. *Here are the Top 5 Languages for Machine Learning, Data Science*. [B.r.]. Dostupné také z: <https://www.aitrends.com/data-science/here-are-the-top-5-languages-for-machine-learning-data-science/>. [Online; accessed 24-March-2022].
23. "About pandas". [B.r.]. Dostupné také z: <https://pandas.pydata.org/about/>.
24. SIKDAR, Satyaki. "NetworkX Graph Library". [B.r.]. Dostupné také z: <https://www3.nd.edu/~kogge/courses/cse60742-Fall12018/Public/StudentWork/ParadigmPaperv1/Sikdar-NetworkX.pdf>.
25. "What Is Matplotlib In Python?" [B.r.]. Dostupné také z: <https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/>.
26. BASTIAN, Mathieu; HEYMANN, Sebastien; JACOMY, Mathieu. Gephi: an open source software for exploring and manipulating networks. In: *Proceedings of the international AAAI conference on web and social media*. 2009, sv. 3, s. 361–362. Č. 1.

27. GARBADE, Dr. Michael J. "History of Python". 2022. Dostupné také z: <https://blog.educationecosystem.com/what-is-google-colab/>.
28. *Tensorflow*. [B.r.]. Dostupné také z: <https://www.tensorflow.org/>.
29. "Keras". [B.r.]. Dostupné také z: <https://keras.io/>.
30. VOTAVA, Jiří. "Zajímavosti o mozku". [B.r.]. Dostupné také z: <https://www.brainmarket.cz/nase-novinky/vse--co-jeste-chteli-vedet-o-mozku-a-bali-jste-se-zeptat/>.
31. MELCHER, Kathrin. *A Friendly Introduction to [Deep] Neural Networks*. 2021. Dostupné také z: <https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks>. [Online; accessed 23-March-2022].
32. KAY, Alexx. "Co jsou to umělé neuronové sítě?" [B.r.]. Dostupné také z: <https://www.scienceworld.cz/technologie/co-jsou-to-umele-neuronove-site-4077/>.
33. "Six Types of Neural Networks You Need to Know About". 2021. Dostupné také z: <https://www.sabrepc.com/blog/Deep-Learning-and-AI/6-types-of-neural-networks-to-know-about>.
34. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
35. BANDYOPADHYAY, Hmrishav. "An Introduction to Autoencoders". 2022. Dostupné také z: <https://www.v7labs.com/blog/autoencoders-guide>.
36. DISNEY, Andrew. "Diferent Centralities". *Cambridge Intelligence*. 2020. Dostupné také z: <https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/>.
37. ImageCorrespondingPointsZkopírovat do schránky.ImageCorrespondingPoints. [B.r.]. Dostupné také z: <https://reference.wolfram.com/language/ref/ImageCorrespondingPoints.html>.
38. ZELINKA, Ivan; ZMESKAL, Oldrich; WINDSOR, Leah; CAI, Zhiqiang. Unconventional methods in voynich manuscript analysis. In: *Mendel*. 2019, sv. 25, s. 1–14. Č. 1.
39. JONES, Josh. *vms EVA alphabet*. [B.r.]. Dostupné také z: [http://voynich.webpoint.nl/?page\\_id=138](http://voynich.webpoint.nl/?page_id=138). [Online; accessed 09-April-2022].
40. "Keras - Dense Layer". [B.r.]. Dostupné také z: [https://www.tutorialspoint.com/keras/keras\\_dense\\_layer.htm](https://www.tutorialspoint.com/keras/keras_dense_layer.htm).
41. "Conv1D layer". [B.r.]. Dostupné také z: [https://keras.io/api/layers/convolution\\_layers/convolution1d/](https://keras.io/api/layers/convolution_layers/convolution1d/).
42. "MaxPooling1D layer". [B.r.]. Dostupné také z: [https://keras.io/api/layers/pooling\\_layers/max\\_pooling1d/](https://keras.io/api/layers/pooling_layers/max_pooling1d/).

43. RESTREPO, Ronny. *Derivative of the Sigmoid function*. [B.r.]. Dostupné také z: [http://ronny.rest/blog/post\\_2017\\_08\\_10\\_sigmoid/](http://ronny.rest/blog/post_2017_08_10_sigmoid/). [Online; accessed 09-April-2022].
44. CHEN, B. "7 popular activation functions you should know in Deep Learning and how to use them with Keras and TensorFlow 2". [B.r.]. Dostupné také z: <https://towardsdatascience.com/7-popular-activation-functions-you-should-know-in-deep-learning-and-how-to-use-them-with-keras-and-27b4d838dfe6>.
45. LIU, Danqing. *A Practical Guide to ReLU*. [B.r.]. Dostupné také z: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>. [Online; accessed 09-April-2022].
46. BROWNLEE, Jason. "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning". [B.r.]. Dostupné také z: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.
47. "Mean squared error". [B.r.]. Dostupné také z: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-error>.

## Seznam příloh

V přiloženém zipu se nacházejí tyto soubory:

- **/Knihy** – Adresář obsahující knihy na nichž byla prováděna analýza.
- **/Skripty** – Zde se nacházejí skripty zmiňované v kapitole 5.2.1.
- **/Colab** – Zde se nacházejí vyexportované sešity z Google Colab.
- **/ImageCorrespondingPoints** – Zde se nacházejí vstupní data zmiňované v kapitole 6.1.2
- **/voynich.ttf** – Font Voynichovy abecedy.
- **/readme.txt** – V tomto souboru je podrobnější popis příloh.