

GRAPH OF AGH

**STATYCZNA ANALIZA POWIĄZAŃ AUTORÓW I
WSPÓŁAUTORSTWA PRAC NAUKOWYCH AGH**

STUDIO PROJEKTOWE 1

Autorzy projektu

Krystian Sitarz
Jakub Wądrzyk

Opiekun projektu

Dr hab. inż. Tomasz Hachaj



EAIIB / Katedra Informatyki Stosowanej
Akademia Górnictwo-Hutnicza im. Stanisława Staszica w
Krakowie
Kraków, Polska

28 stycznia 2024 r.

Spis treści

1	Wprowadzenie	1
2	Projekt Systemu	2
2.1	Baza Danych	2
2.2	Wykorzystane Technologie	3
2.3	Projekt Klas	3
3	Proces Pobierania Danych	5
3.1	Autorzy	5
3.2	Artykuły i linki	6
4	Uzyskane Dane	7
5	Tworzenie Grafu Powiązań	8
5.1	Matematyczny model grafu	8
5.2	Węzły i Krawędzie	8
6	Klasteryzacja	11
7	Wizualizacja	12
8	Podsumowanie	16
	References	17

1. Wprowadzenie

Projekt ma na celu przeprowadzenie statycznej analizy powiązań między autorami prac naukowych w ramach Akademii Górnictwo-Hutniczej (AGH). Analiza opiera się na relacjach współautorstwa, umożliwiając identyfikację grup projektowych wynikających z współpracy między autorami.

Ostatecznym rezultatem projektu jest graf utworzony za pomocą programu Gephi, umożliwiający dowolną analizę i wyciąganie wniosków.

2. Projekt Systemu

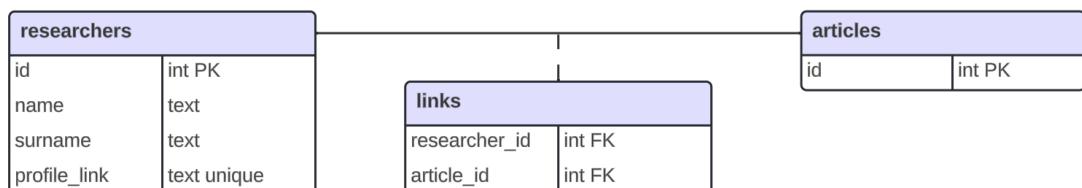
Jednym z głównych wyzwań podczas realizacji projektu było pozyskanie danych, ponieważ nie było dostępu do nich poprzez publiczne API. W związku z tym zdecydowaliśmy się na wykorzystanie techniki Web Scrapingu [1], co pozwoliło nam efektywnie zdobyć potrzebne informacje.

Informacje o publikacjach naukowych są udostępniane na stronie: Baza Danych o Autorach i Publikacjach AGH (BaDAP) [2].

2.1 Baza Danych

Do wykonania projektu konieczne były informacje o ilości wspólnych artykułów wiążących poszczególnych autorów.

W tym celu zaprojektowaliśmy bazę danych przedstawioną na rysunku 2.1 Poprzez podzielenie bazy na trzy tabele ułatwiliśmy odpytywanie bazy danych o współautorstwo. Przy projektowaniu braliśmy również pod uwagę sam proces pobierania danych, stąd w tabeli researchers kolumna przechowująca link do profilu każdego z autorów, z którego braliśmy informacje o identyfikatorach jego prac naukowych.



Rysunek 2.1: Diagram bazy danych (Źródło: opracowanie własne)

2.2 Wykorzystane Technologie

Do samego procesu scrapowania danych i zapisywania ich do bazy użyliśmy języka [Python \[3\]](#) w wersji 3.12.2 oraz następujących technologii:

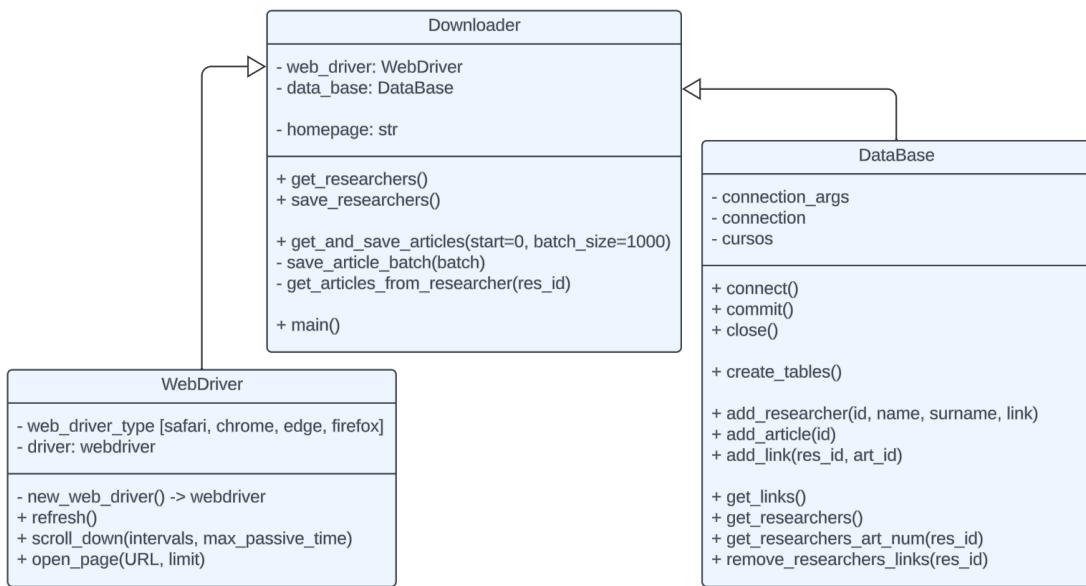
- [Python \[3\]](#)
 - [Python Logging \[4\]](#) (*zapis logów z działania programu do pliku*)
 - [Selenium \[5\]](#) (*biblioteka pozwalająca na automatyczną obsługę przeglądarki internetowej*)
 - [Beautiful Soup \[6\]](#) (*parsowania kodu HTML*)
 - [Psycopg2 \[7\]](#) (*integracja PostgreSQL z Python*)
- [PostgreSQL \[8\]](#)

Aby w łatwy sposób debugować częste problemy przez łącze internetowe posłużyliśmy się modułem logging do zapisywania statusu programu.

2.3 Projekt Klas

Zaprojektowaliśmy system współpracujących ze sobą klas zaprezentowany na rysunku [2.2](#). Każda klasa obsługuje jeden z elementów projektu:

- połaczenie z bazą danych
- automatyzację przeglądarki internetowej - [Selenium \[5\]](#)
- scrapowanie danych



Rysunek 2.2: Diagram UML zaimplementowanych klas (Źródło: opracowanie własne)

3. Proces Pobierania Danych

Wykorzystując w ten sposób zaprojektowany w poprzednim rozdziale system, w prosty sposób mogliśmy scrapować dane ze strony AGH z pracami naukowymi [2].

3.1 Autorzy

Na początku pobraliśmy wszystkich autorów i linki do ich profilów ze strony głównej, oraz zapisaliśmy w pamięci podręcznej spodziewaną ilość artykułów.

Klasa WebDriver (zob. rys. 2.2) zręcznie obsłużyła te wymagania poprzez przewinięcie całej strony w dół, aby później wygodnie sparsować dane z pobranego kodu źródłowego całej strony internetowej z autorami dzięki Selenium [5] i Beautiful Soup [6]. Następnie używając metod klasy DataBase zapisaliśmy autorów do bazy danych w PostgreSQL [8].

Część strony internetowej zawierającej tabelę autorów wraz z liczbą ich publikacji można zobaczyć na rysunku 3.1, imiona i nazwiska osób zostały zanonimizowane.

Autor	Jednostka	Pozycje	Pozycje
		afiliowane	nieafiliowane
Andrzej Gajda	ACMIn	6 poz.	—
Aleksander Kowalewski	WH	1 poz.	—
Andrzej Zdziarski	WFIIIS	—	—
Andrzej Zdziarski-Kowalewski	WGGiŚ	1 poz.	—
Andrzej Zdziarski-Kowalewski	WEAiIB-keaspe	5 poz.	—
Andrzej Wójcik	WI	—	—
Andrzej Zdziarski	WILiGZ-kezp	3 poz.	1 poz.
Andrzej Zdziarski - zaliczeni	WGGiOŚ-khgi	14 poz.	—
Andrzej Zdziarski	WIMiC-kchk	132 poz.	2 poz.
Andrzej Zdziarski	*WEAiE-kee	1 poz.	9 poz.
Andrzej Kowalewski	WEiP-kzre	24 poz.	—
Andrzej Gajda	*WIMiC-kb	2 poz.	1 poz.

Rysunek 3.1: Część strony z tabelą autorów (Źródło: BaDAP [2])

3.2 Artykuły i linki

Następnie iterując po każdym elemencie tablicy autorów, otwieraliśmy ich stronę i parsowaliśmy identyfikatory artykułów wylistowanych dla danego autora.

Aby przyspieszyć proces scrollowania w dół zaimplementowaliśmy algorytm estymujący czas takiego przewijania na bazie zapisanych w pamięci lokalnej informacji o oczekiwanej ilości artykułów, zaprezentowany wzorem 3.1.

$$T = \begin{cases} 1 & N \leq 256 \\ \lfloor \sqrt{\frac{N}{64}} \rfloor & N > 256 \end{cases} \quad (3.1)$$

Po załadowaniu całej strony parsowaliśmy id i zapamiętywaliśmy je lokalnie w tablicy aby móc zapisać seriami po 1000 do bazy danych. Przykładowe źródło strony z artykułami naukowymi danego autora zostało przedstawione na rysunku 3.2

```
<div class="infinite-scroll-component__outerdiv">
  <div class="infinite-scroll-component " style="height:auto;overflow:auto;-webkit-overflow-scrolling:touch">
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
      ><div class="flex px-2 sm:px-5">...</div> flex
      ><div class="flex px-2 sm:px-5 pt-5 break-words">...</div> flex
      ><div class="flex justify-between items-end h-full">...</div>
        <div class="flex">...</div> flex
        ><a target="_blank" class="font-bold p-2 hover:underline details" href="/publikacja/148385">...</a>
      </div>
    </div>
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
    ><div class="flex flex-col bg-white group my-5 border rounded border-stone-300 hover:shadow-lg duration-100">...</div> flex
  </div>
</div>
```

Rysunek 3.2: Przykładowe źródło strony internetowej z publikacjami wybranego autora (Źródło: BaDAP [2])

Przy wykorzystywaniu webdriver-a spotkaliśmy się z problemem, że po jakimś czasie przestawał on odpowiadać, dlatego podczas zapisywania serii identyfikatorów do bazy danych jednocześnie przeładowywaliśmy okno webdriver-a, żeby go odświeżyć.

4. Uzyskane Dane

W rezultacie otrzymaliśmy:

12 314 autorów
146 968 artykułów
322 779 powiązań autor-artykuł

Przykładowa część tabeli autorów wycięta z bazy danych w PostgreSQL [8] została przedstawiona na rysunku 4.1 (rekordy danych zostały zanonimizowane).

id [PK] integer	name text	surname text	profile_link text
2135	Irena	Drummier	https://herdo.agn.edu.pl/autor/edmund-herdo-hanna-052877
2136	Oktawia Folt	Eferamo	https://herdo.agn.edu.pl/autor/eferamo-oktawa-folt-082584
2137	Małgorzata	Fliszt-Rudecka	https://herdo.agn.edu.pl/autor/eklent-malgorzata-fliszt-002907
2138	Sadik S. Hamad	Elawgali	https://herdo.agn.edu.pl/autor/elawgali-sadik-s-hamad-005292
2139	Rami A.	Elkayed	https://herdo.agn.edu.pl/autor/elkayed-rami-a-005477
2140	Safya Samal	Elsharawy	https://herdo.agn.edu.pl/autor/elsharawy-safya-samal-060017
2141	Gabriela	Elstebnick	https://herdo.agn.edu.pl/autor/elstebnick-gabriela-010124
2142	Zbigniew Witold	Engel	https://herdo.agn.edu.pl/autor/engel-zbigniew-witold-002545
2143	Weronika	Enomoto	https://herdo.agn.edu.pl/autor/enomoto-weronika-0072413
2144	Efe	Engin	https://herdo.agn.edu.pl/autor/engin-efe-04697
2145	Emilia	Ermen-Kowalczenko	https://herdo.agn.edu.pl/autor/ermen-kowalczenko-emilia-002320
2146	Sedatcan	Erol	https://herdo.agn.edu.pl/autor/errol-sedatcan-005848
2147	Kings	Erdneys	https://herdo.agn.edu.pl/autor/erdneys-kings-001577
2148	Maria	Esmond	https://herdo.agn.edu.pl/autor/esmond-maria-059487

Rysunek 4.1: Część tabeli autorów z bazy danych (Źródło: opracowanie własne)

5. Tworzenie Grafu Powiązań

Do stworzenia grafu wykorzystaliśmy program Gephi [9]. Jest to zaawansowane narzędzie dostarczające algorytmy klasteryzacji i wizualizacji grafów, radzi sobie nawet z naprawdę dużymi sieciami - idealne dla naszych potrzeb.

5.1 Matematyczny model grafu

Niech graf nieskierowany G będzie zdefiniowany jako para (V, E) , gdzie:

- V to niepusty zbiór wierzchołków, będących autorami pod patronatem AGH [10].
- $E \subseteq \{\{u, v\} : u, v \in V \wedge uRv \leftrightarrow \text{węzły } u \text{ oraz } v \text{ są współautorami artykułu}\}$.

5.2 Węzły i Krawędzie

Najprostszym sposobem na zimportowanie danych do programu Gephi [9] było przekazanie mu dwóch plików z węzłami i krawędziami grafu, wybraliśmy do tego format CSV [11].

Aby w rezultacie otrzymać graf którego węzłami są poszczególni autorzy, a każda krawędź ich łącząca reprezentuje wspólnie napisaną publikację, należało zaprojektować odpowiednie zapytania do bazy danych w PostgreSQL [8].

Stworzenie pliku węzłów było bardzo proste, zapytanie zostało zaprezentowane na rysunku 5.1

```
SELECT
    r.id, r.name || ' ' || r.surname AS full_name,
    COUNT(l.article_id) AS art_num
FROM researchers AS r
LEFT JOIN links AS l ON r.id = l.researcher_id
GROUP BY r.id ORDER BY r.id;
```

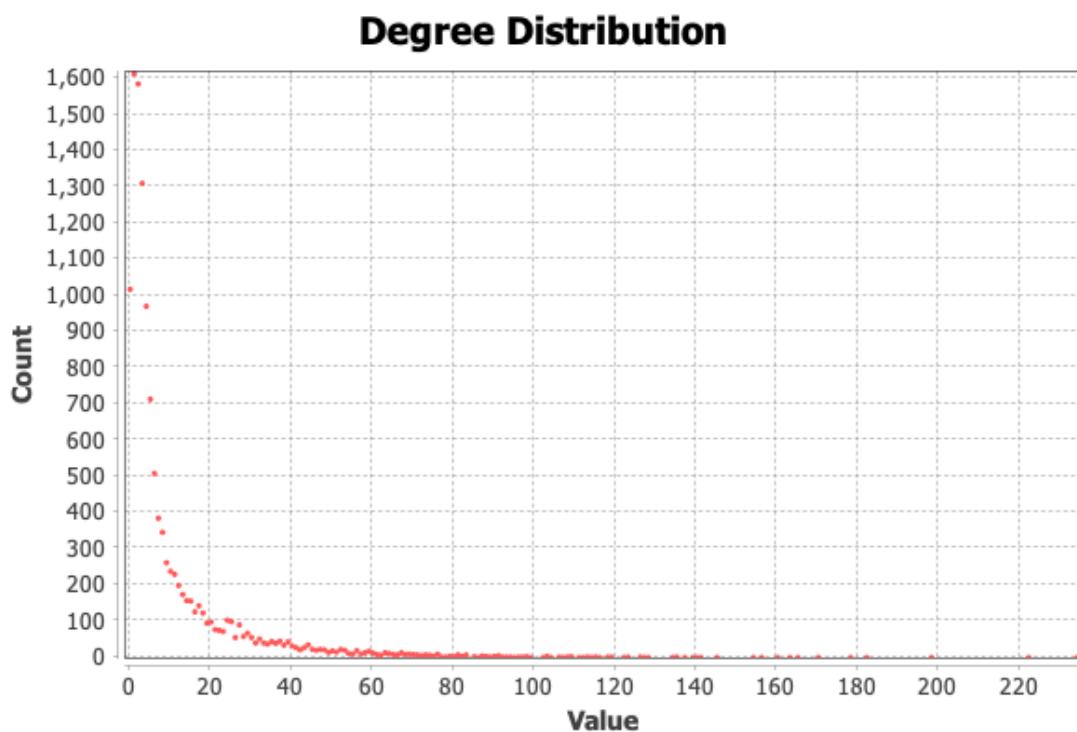
Rysunek 5.1: Zapytanie do bazy danych o autorów (Źródło: opracowanie własne)

Zapytanie mające zwrócić powiązania między autorami, a dodatkowo policzyć ilość współpracy między każdą parą, jest dużo bardziej złożone. Należało połączyć tabele autorów ze sobą samą poprzez tabelę zawierającą linki między autorem, a publikacją. Powstałe w ten sposób zapytanie widnieje na rysunku 5.2.

```
SELECT
    r1.id AS researcher1_id,
    r2.id AS researcher2_id,
    COUNT(DISTINCT l1.article_id) AS common_articles_count
FROM
    researchers r1
JOIN
    links l1 ON r1.id = l1.researcher_id
JOIN
    links l2 ON l1.article_id = l2.article_id AND l1.researcher_id < l2.researcher_id
JOIN
    researchers r2 ON l2.researcher_id = r2.id
GROUP BY
    researcher1_id, researcher2_id
ORDER BY
    researcher1_id, researcher2_id;
```

Rysunek 5.2: Zapytanie do bazy danych o powiązania (Źródło: opracowanie własne)

Rezultaty tych zapytań zapisaliśmy w plikach CSV [11], które z kolei zainportowaliśmy do programu Gephi [9]. Powstał nieskierowany graf zawierający 12 314 węzły oraz 63 913 krawędzie. Rozkład stopni wierzchołków został przedstawiony na wykresie z rysunku 5.3 gdzie średni stopień węzła wynosi 10.381.



Rysunek 5.3: Rozkład stopni węzłów w grafie (Źródło: Gephi [9])

6. Klasteryzacja

Jedną z funkcjonalności programu Gephi [9] jest możliwość klasteryzacji grafu algorytmem *modularity* opisany w artykule [12]. Wykrywa on społeczności w sieci na podstawie modularity, której wyższa wartość oznacza lepszy podział grafu.

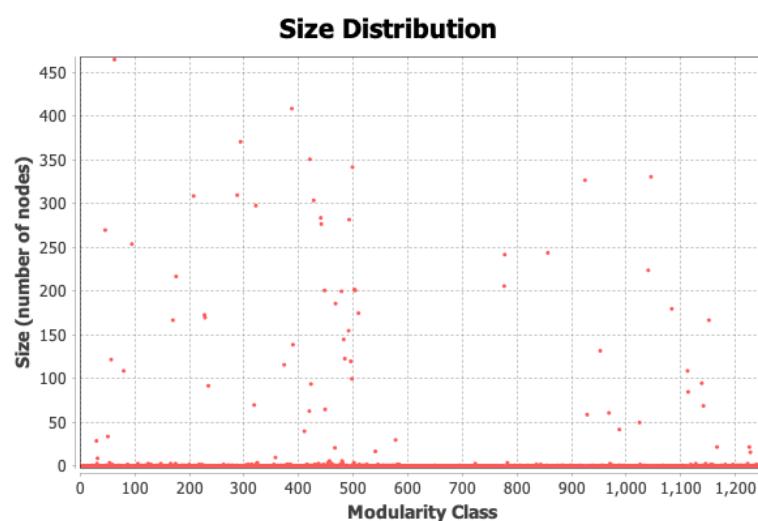
Algorytm został uruchomiony z parametrami z tabeli 6.1, a w rezultacie otrzymaliśmy modularność o cechach pokazanych w tabeli 6.2 oraz na wykresie z rysunku 6.1.

<i>Randomize</i>	On
<i>Use edge weights</i>	On
<i>Resolution</i>	0.25

Tabela 6.1: Parametry do algorytmu modularity (Źródło: opracowanie własne)

<i>Modularity</i>	0.794
<i>Modularity with resolution</i>	0.160
<i>Number of Communities</i>	1251

Tabela 6.2: Rezultaty algorytmu modularity (Źródło: algorytm modularity [12])

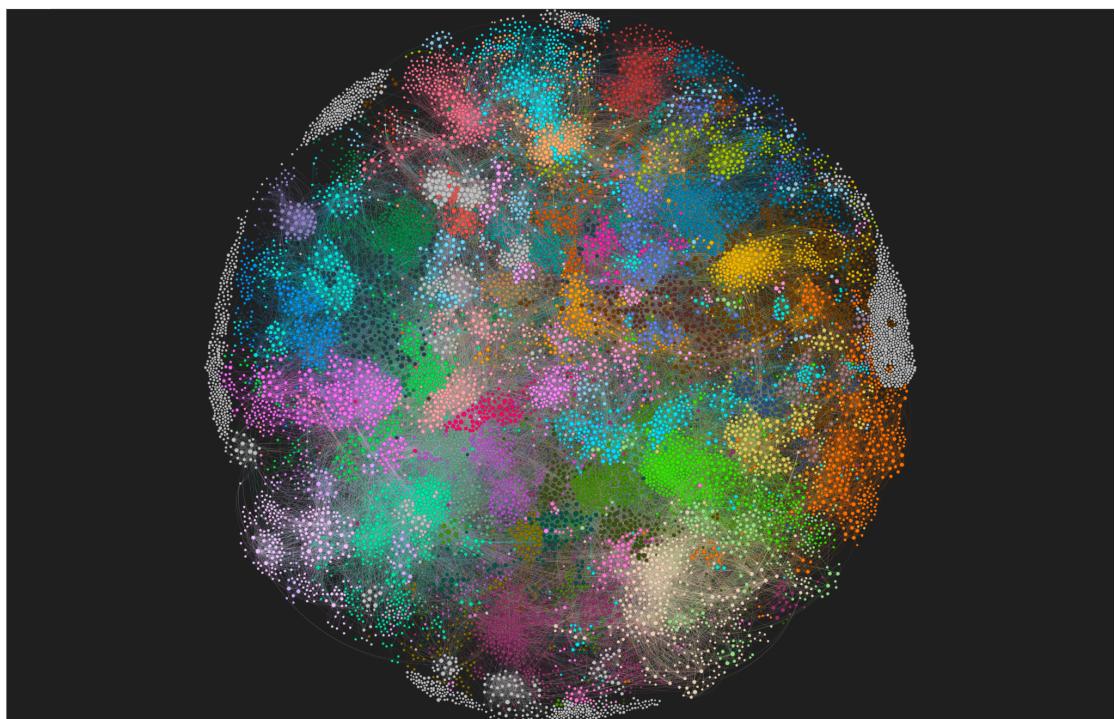


Rysunek 6.1: Rozkład rozmiaru społeczności w zależności od klasy modularności, gdzie os pionowa to rozmiar (liczba węzłów), a os pozioma to klasa modularności. (Źródło: algorytm modularity [12])

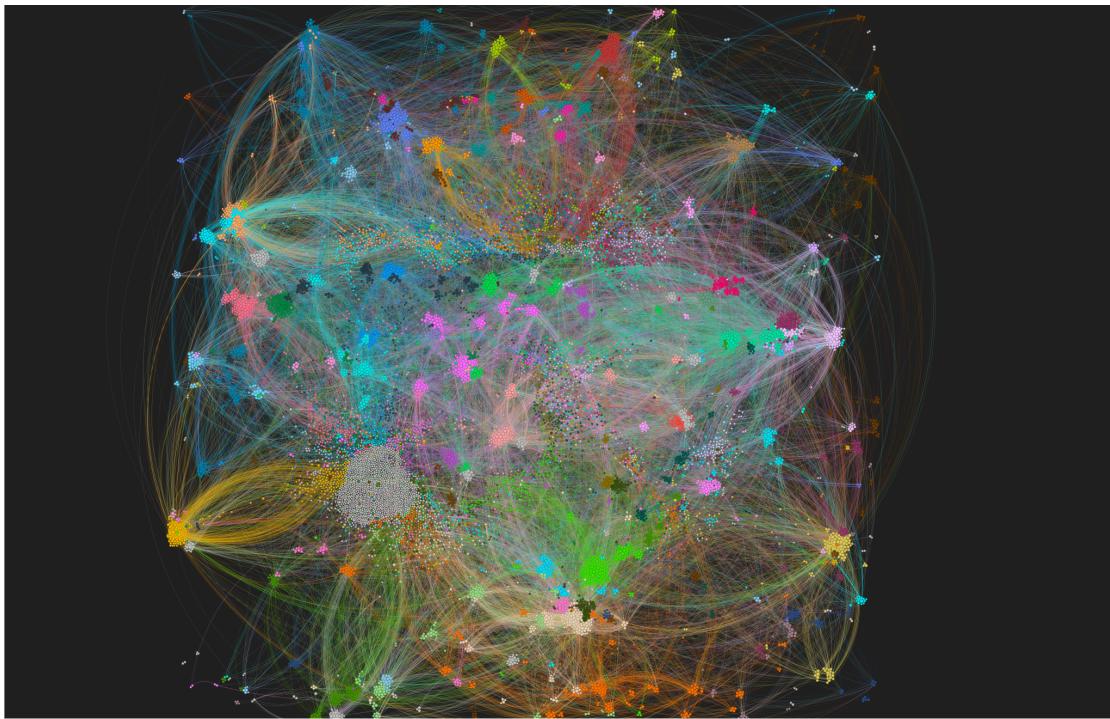
7. Wizualizacja

Stworzony graf został dodatkowo pokolorowany zgodnie z przynależnością do odpowiedniej klasy (wyznaczonej w 6) i zwizualizowany w ten sposób, iż wierzchołki reprezentujące autorów o większej liczbie publikacji są wizualnie większe, a krawędzie łączące autorów o większej liczbie wspólnych publikacji - wizualnie grubsze.

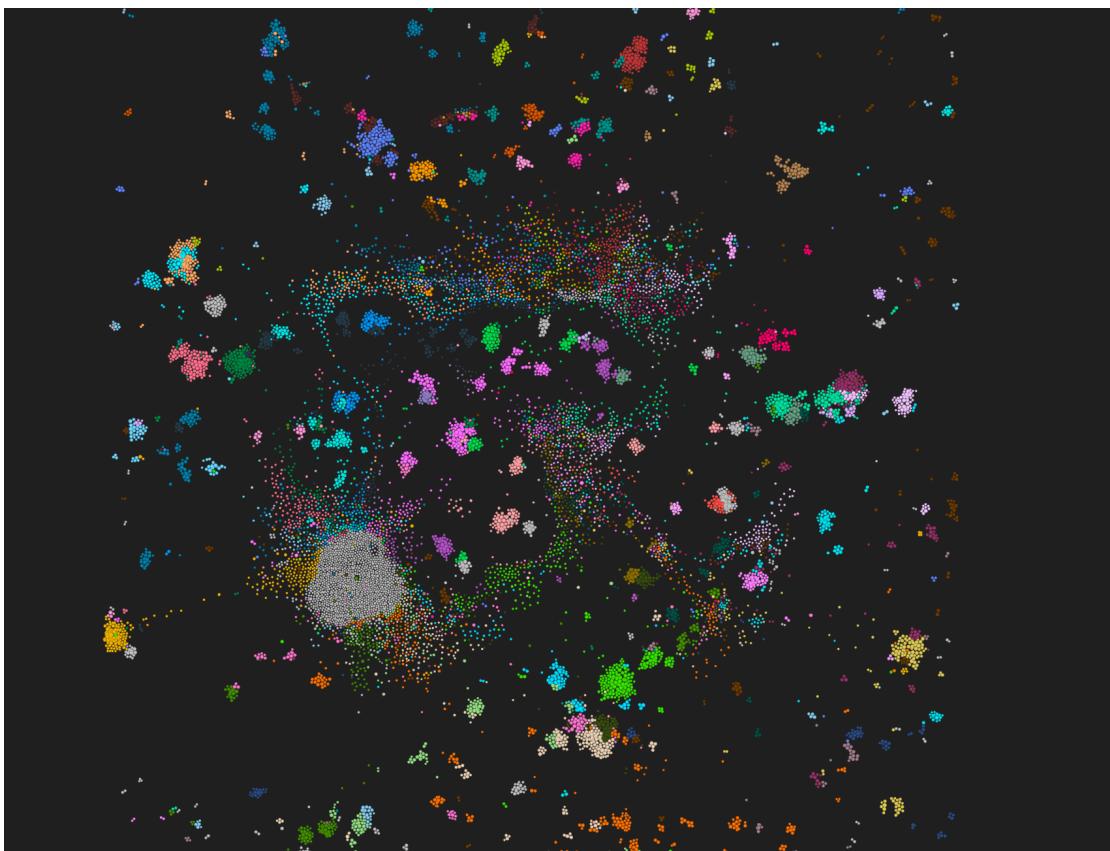
W rezultacie otrzymaliśmy obraz przedstawiony na kolejnych rysunkach 7.1, 7.2, 7.3.



Rysunek 7.1: Graf podzielony na klastry i pokolorowany



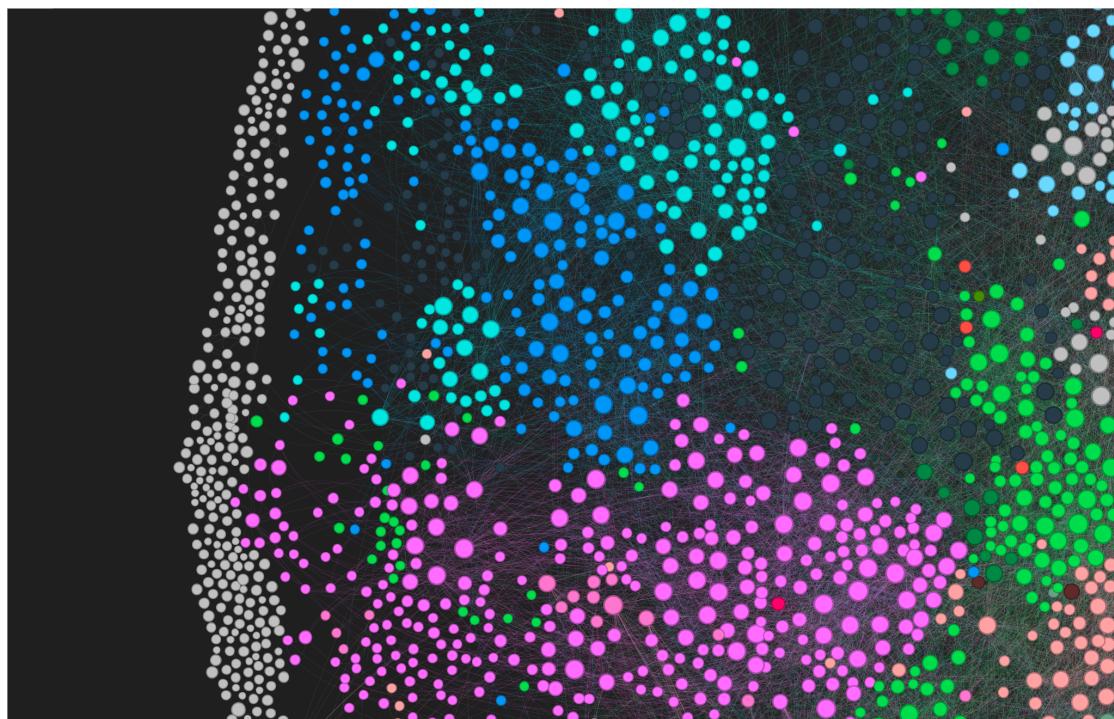
Rysunek 7.2: Graf podzielony na klastry z większymi odstępami



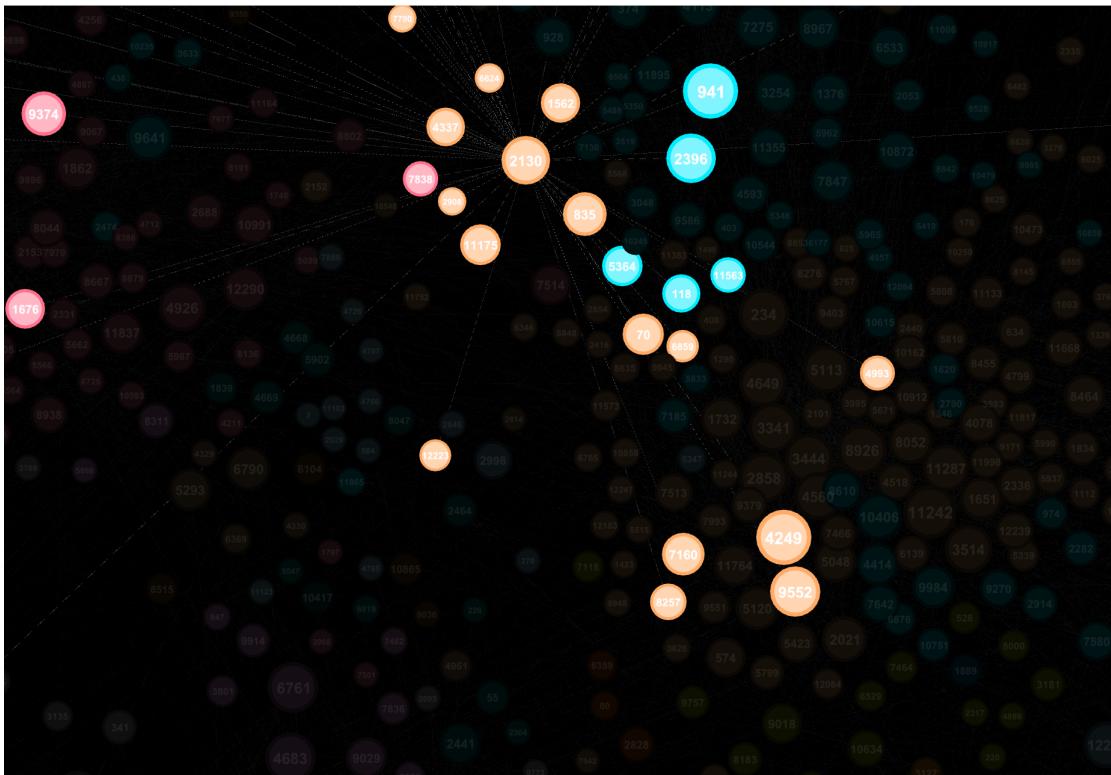
Rysunek 7.3: Graf z większymi odstępami bez krawędzi

Program Gephi [9] daje możliwość dowolnego manipulowania grafem, przybliżania oddalania, przesuwania krawędzi, itp. Możliwość podświetlenia sąsiadów danego wierzchołka znacznie ułatwia analizę grafu.

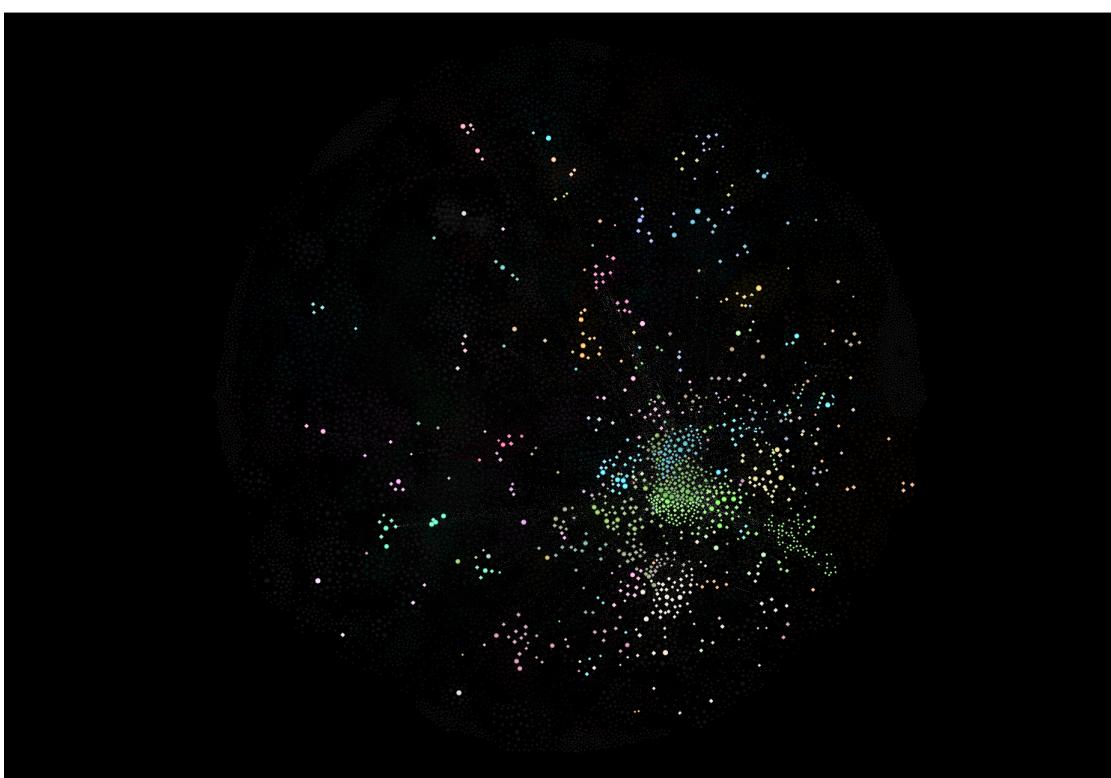
Takie możliwości zaprezentowano na rysunkach 7.4, 7.5, oraz 7.6.



Rysunek 7.4: Graf zbliżony



Rysunek 7.5: Zaznaczony jeden wierzchołek z sąsiadami



Rysunek 7.6: Zaznaczone wiele wierzchołków z sąsiadami

8. Podsumowanie

Wynikiem projektu jest zwizualizowany, sklastrowany nieskierowany graf powiązań pomiędzy osobami publikującymi artykuły naukowe pod patronatem uczelni AGH [10]. Wierzchołkami grafu są osoby publikujące, natomiast połączenie ustalane jest na podstawie wspólnej ilości wydań.

Największym problemem napotkanym podczas realizacji projektu było pobranie danych dotyczących wydających oraz ich prac. Problem rozwiązyano za pomocą techniki "data scraping". Stworzono program, który na wstępie pobiera listę wszystkich odnośników URL do stron internetowych autorów. Następnie automatyzuje przeglądarkę internetową, która odwiedza owe strony i pobiera dane dotyczące publikacji. Przetwarza je, zapisując następnie w bazie danych.

Po zdobyciu danych są one przetwarzane wynikiem czego są dwa pliki CSV [11] definiujące graf. Pliki te służą do uzyskania grafu podzielonego na klastry. Pozwala na to oprogramowanie Gephi [9].

References

- [1] CodersLab. Web scraping – co to jest? scrapowanie danych ze stron internetowych. (data dostępu: 2024-01-25). [Online]. Available: <https://coderslab.pl/pl/blog/web-scraping-co-to-jest-scrapowanie-danych-ze-stron-internetowych>
- [2] A. BD, CRI. Badap. (data dostępu: 2024-01-25). [Online]. Available: <https://badap.agh.edu.pl/>
- [3] Python Software Foundation, *Python 3.12 Documentation*, (data dostępu: 2024-01-25). [Online]. Available: <https://docs.python.org/3.12/>
- [4] ———, *Python Documentation - Logging*, (data dostępu: 2024-01-25). [Online]. Available: <https://docs.python.org/3/library/logging.html>
- [5] S. F. Conservancy, *Selenium*, (data dostępu: 2024-01-25). [Online]. Available: <https://www.selenium.dev/about/>
- [6] L. R. R. 546, *Beautiful Soup*, (data dostępu: 2024-01-25). [Online]. Available: <https://beautiful-soup-4.readthedocs.io/en/latest/>
- [7] T. P. T. Daniele Varrazzo, *Psycopg2*, (data dostępu: 2024-01-25). [Online]. Available: <https://www.psycopg.org/>
- [8] T. P. G. D. Group, *PostgreSQL*, (data dostępu: 2024-01-25). [Online]. Available: <https://www.postgresql.org/about/>
- [9] Gephi.org, *Gephi*, (data dostępu: 2024-01-25). [Online]. Available: <https://gephi.org/>
- [10] Akademia Górnictwo-Hutnicza, (data dostępu: 2024-01-25). [Online]. Available: <https://www.agh.edu.pl/>
- [11] R. 4180, *CSV*, (data dostępu: 2024-01-25). [Online]. Available: <https://www.rfc-editor.org/rfc/rfc4180>

- [12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.