

A PYTHON SCRIPT TO DATAMINE SCIENTIFIC LITERATURE - USING THE EXAMPLE OF AQUEOUS BATTERIES

By Irem Aydingun

University of Kent, School of Physical Sciences –
2025

Total: 79 Pages, 14,394 words

Abstract

Li-Ion batteries have been the preferred chemistry for secondary batteries for many years, due to its high energy capacity, longer lifetime and light weight. Many scientists are trying to aid the development towards safer and more cost-effective systems. Researchers are increasingly interested in aqueous rechargeable lithium-ion batteries (ARLB) to avoid using flammable and costly electrolytes. Many materials are suitable in the use of salt-water batteries as anode and cathode materials and are being experimented with. In recent years, many research papers have been published with experimental and theoretical data for the suitability of these materials. The materials with an interest have a low instalment cost, long cycling life, high energy efficiency and are sustainable. The aim of this project is to produce a proof of concept python script to extract exemplar data from previously published articles and to build a database of potential electrode materials and electrolytes. The data collected should include electrochemical properties, e.g. potential range, capacity retention and current density. The proposed method used to obtain this data into a database, is by web scraping using a variety of online articles. The data in the articles are presented to the user depending on the criteria selected at the initial input level by the user. The extracted information is also filtered to the criteria chosen by the user.

Declaration of Authenticity

I, Irem Aydingun, declare that this thesis titled 'A PYTHON SCRIPT TO DATAMINE SCIENTIFIC LITERATURE - USING THE EXAMPLE OF AQUEOUS BATTERIES' and all the research presented within, were produced by me.

I declare that I have acknowledged all contributions by others, and unless otherwise stated, all work presented is my own.

This work has not been previously submitted, in part or in whole, for any other academic degree. The copyright of this degree rests with the author.

A handwritten signature in black ink, reading "I. Aydingun" with a stylized flourish underneath.

Irem Aydingun

27/03/2025

Table of Contents

Abstract	1
Declaration of Authenticity	2
Table of Contents	3
Table of Figures.....	4
Table of Tables	6
Abbreviations.....	7
Publications.....	8
1. Introduction.....	9
1.1 Introduction to Aqueous Batteries	9
1.2 How Aqueous Batteries Work	13
1.3 Aims and Objectives	17
2. Method	19
2.1 Tools and Licensing	19
2.2 Google Scholar as the Source of Literature	22
2.3 Building a Manual Database	33
2.4 Python Libraries.....	39
2.5 Data Mining	42
2.6 Flowchart.....	45
2.7 Method Conclusion	47
3. Results.....	48
3.1 Results from Code.....	48
3.2 Results from Manually Created Database	53
3.5 Results Conclusion	58
4. Conclusions and Summaries.....	59
5. Future Work and Discussion.....	60
5.1 Building a Standardised Database	60
5.2 Applications of AI.....	63
5.3 Improving the Scope of the Project	68
References.....	70
Appendix.....	74

Table of Figures

Figure 1 - The electrochemical stability window in relation to pH. Image reproduced from Sui and Ji (2021) (ref. 5)	11
Figure 2 – The advantages and disadvantages of Aqueous Battery Systems. Image reproduced from Ahn et al. (2023) (ref. 6).....	12
Figure 3 - Salt to solvent molar and weight ratios for the NaOTf–H ₂ O binary system with corresponding molality. Image reproduced from Zhang et al. (2021) (ref. 7)	13
Figure 4 – The “rocking chair” and “hybrid” type aqueous rechargeable batteries – shown schematically. Image reproduced from Liu et al. (2018) (ref. 9)	15
Figure 5 – Role of a proxy in accessing internet browsing – shown schematically. Reproduced from Fortinet (ref. 20).....	21
Figure 6 – Royal Society of Chemistry Webpage showing the webpage which would need to be web scraped if the RSC website is the source of literature.....	23
Figure 7 – Royal Society of Chemistry website – showing the search filters already built into the website.	23
Figure 8 – Royal Society of Chemistry website – showing the final webpage accessed before a PDF copy of the article could be downloaded.....	24
Figure 9 – The difference in the amount of search results for Web of Science in comparison to Google Scholar for different search parameters.	26
Figure 10 – Percentage comparison of relevant articles produced from different search parameters for Web of Science in comparison to Google Scholar.	27
Figure 11 – Comparison of how many times a specific article was cited by other articles. This was used to determine how frequently others accessed the same article via Web of Science in comparison to Google Scholar.	28
Figure 12 – The number of total articles published on ‘Aqueous Battery Review’ between 2014 and 2022 for (a) Web of Science and (b) Google Scholar.	30
Figure 13 – A flowchart of how the articles from the search results were manually sorted and categorised to differentiate between anode and cathode materials.	34
Figure 14 - Steps involved in Knowledge Discovery in Databases to extract valuable and relevant data. Reproduced from Johnson (ref. 42)	43
Figure 15 – The expected flow of the python script and the steps in which different tasks should be executed.	46
Figure 16 – How the data was handled and the format of the raw data obtained leading to the format of the processed data.	50
Figure 17 – A Google Chrome tool which allows you to view the HTML tags for different elements on a webpage. Useful in determining the tags for the elements that being web scraped.....	52
Figure 18 – Comparison of the number of times “Li” appeared in the title results obtained by the program, in comparison to “Zn”. This was used as a proof of concept to compare to the human obtained results.	55
Figure 19 – How frequently each material appeared, represented by the size and frequency of the word for (a) Cathode materials and (b) Anode materials, using a visual representation	57
Figure 20 – Live data counts for the PubChem databases on 2 nd April 2024.....	61

Figure 21 – A screenshot of a search query in Google Scholar Labs when searching for articles related to the research topic. 67

Table of Tables

Table 1 - Ionic conductivity for different electrolytes used in various battery technologies. ⁵	
.....	10
Table 2 - Electrochemical performances of different cathode materials for aqueous zinc-ion batteries. Table reproduced from Selvakumaran et al. (2019) (ref. 11)	17
Table 3 – Partial representation of the main manually built database table - produced as a guideline and training set for the program produced database.	36
Table 4 – Results showing the associated variables obtained from the web scrape for each article.	38

Abbreviations

AI – Artificial Intelligence

ESW – Electrochemical Stability Window

HER – Hydrogen Evolution Reaction

OER – Oxygen Evolution Reaction

SEI – Solid Electrolyte Interface

CEI – Cathode Electrolyte Interface

ARHBs – Aqueous Rechargeable Hybrid Batteries

IDE – Integrated Development Environment

API – Application Programming Interface

RSC – Royal Society of Chemistry

AI – Artificial Intelligence

VPN – Virtual Private Network

WOS – Web Of Science

CSS – Cascading Style Sheets

MEE – Material for Energy and Electronics

LLMs – Large Language Models

SIW – Salt in Water

WISE – Water in Salt Electrolyte

Publications

<https://github.com/IA250/Web-scraping-Scientific-Literature>

1. Introduction

1.1 Introduction to Aqueous Batteries

Due to the reducing supply of fossil fuels and the environmental impacts caused by traditional energy sources, a great amount of research has been directed towards alternative sustainable chemistries. The increased demand for battery systems with high energy density, improved cycle stability and good energy efficiency has led to the popular use of lithium-ion batteries worldwide.¹ The many applications include transportation, such as car batteries and e-scooters, portable devices including phone batteries and toys as well as many more applications.^{2,3} Despite having a high energy density, the environmental and safety impacts due to the use of volatile, flammable, and toxic organic electrolytes makes their safety an issue.⁴ This has led to research moving away from these chemistries and instead focusing on safer and more sustainable systems, such as, aqueous batteries.

Aqueous electrolytes are a technology that are gaining increased interest due to the abundance of water along with the cheap cost of purifying it in order to use it as an electrolyte in aqueous batteries.⁵ This is a contrast to ionic or organic solvents which require a lot of energy to manufacture and often use petroleum products, which have negative environmental impacts.⁵ This leads on to another reason for the increased interest in aqueous battery chemistries. They are environmentally friendly and non-hazardous, the electrolysis products, once the electrolyte breaks down, is hydrogen and oxygen. None of the elements are harmful to the environment, making them a safer alternative. Because the non-flammable electrolyte, aqueous rechargeable lithium-ion batteries (ARLB) also demonstrate improved

battery safety.⁵ Another advantage of using aqueous batteries is their high ionic conductivity in comparison to other electrolytes.⁶

Table 1 - Ionic conductivity for different electrolytes used in various battery technologies.⁵

Electrolyte type	Electronic conductivity
Ionic Electrolytes	$\sim 10^{-1} \Omega^{-1} \text{ cm}^{-1}$ ⁶
Organic Electrolytes	$10^{-3} - 10^{-2} \Omega^{-1} \text{ cm}^{-1}$ ⁶
Polymer Electrolytes	$10^{-7} - 10^{-3} \Omega^{-1} \text{ cm}^{-1}$ ⁶
Inorganic Solid Electrolytes	$10^{-7} - 10^{-2} \Omega^{-1} \text{ cm}^{-1}$ ⁶

A high ionic conductivity is advantageous in battery chemistries, as it results in large dielectric constants, large dipole moments, and excellent acceptor/donor numbers.⁵

However, there are multiple limitations for the use of aqueous batteries at grid level and for large-scale energy storage. One of the main limitations includes the narrow electrochemical stability window (ESW) for H₂O, which is around 1.23V.⁷ At low potentials, water can be reduced, releasing H₂ gas due to the hydrogen evolution reaction (HER).⁵ This process happens on the surface of the anode material. Two electrons are transferred from the electrode to surface-adsorbed H₂O molecules or protons. At high potentials, water is oxidized, setting free O₂ gas in the oxygen evolution reaction (OER). This process happens on the cathode surface, where four electrons are transferred and H₂O molecules are converted to OH⁻ ions which are then absorbed on the electrode's surface to form the oxygen gas.⁵ These reactions are pH dependant and can be summarised in **Figure 1**.

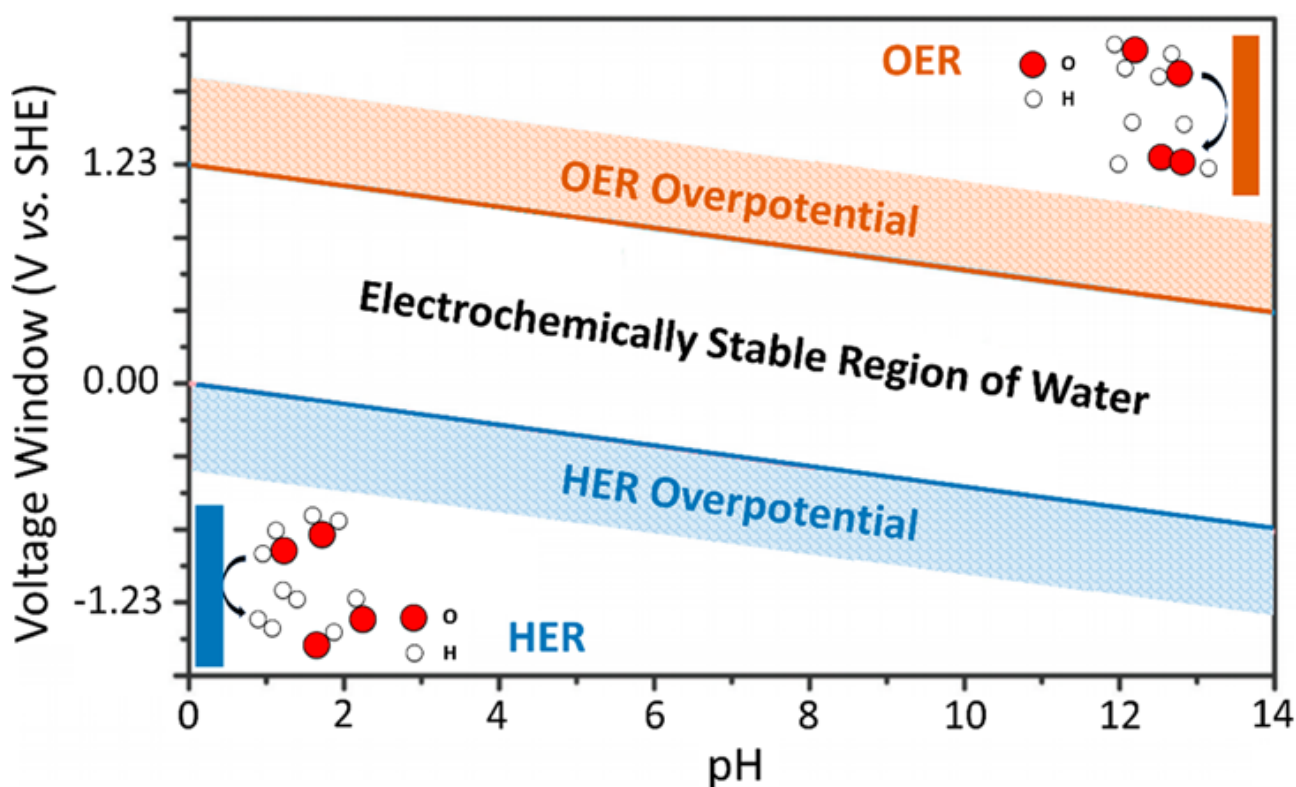


Figure 1 - The electrochemical stability window in relation to pH. Image reproduced from Sui and Ji (2021) (ref. 5)

Another issue impacting the cycling of aqueous batteries is that the Solid Electrolyte Interface (SEI) is unstable. If the SEI is not stable, the battery performance is limited because the SEI will form and break and might grow very thick not allowing the ions taking part in the transport process to be diffused through the SEI.⁸ On the other hand, without the SEI the electrode can start leaking into the electrolyte consuming the active material and causing battery failure. On the cathode side, the cathode electrolyte interface (CEI) is in general thinner and electrochemically more stable and easier to diffuse. While on the anode side the SEI is much thicker and often less stable leading to battery failure. This issue causes electrode corrosion and stops the battery from working. This is a general problem in all batteries, however, this is accelerated in aqueous batteries.⁶ This means that the output voltages of aqueous batteries have to be kept low, which limits energy density and the usage of highly

reducing and highly oxidizing electrodes.⁵ These concerns lead to the short cycling of the batteries making them unsuitable to many applications discussed at the start of this thesis.

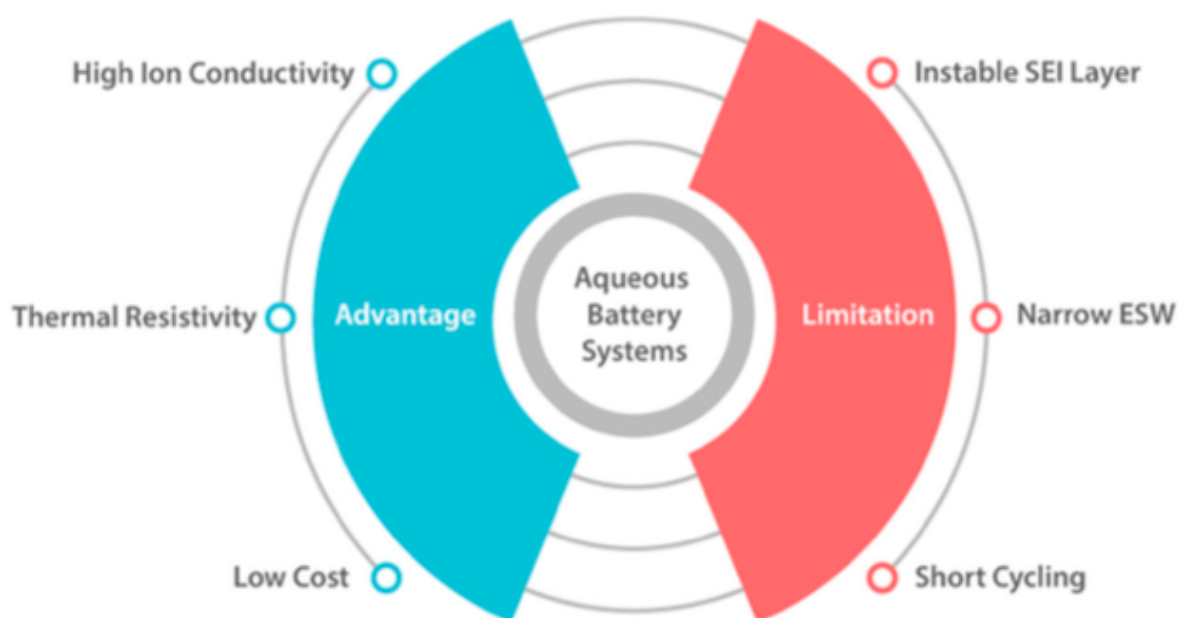


Figure 2 – The advantages and disadvantages of Aqueous Battery Systems. Image reproduced from Ahn et al. (2023) (ref. 6)

Overall, the advantages and disadvantages of aqueous systems can be summarised in **Figure 2**. There are many strategies currently being researched to improve the limitations of aqueous batteries, including improving the materials used in electrodes to ensure they have redox potentials within the narrow electrolysis potential window of water. This is to prevent a water-splitting reaction from occurring.⁶ This thesis will be a preliminary approach aimed to collate the researched and published strategies in to a database, which can further be used to improve aqueous battery chemistries by making results easily accessible. The database can in the future be combined with machine learning algorithm. The produced script will

demonstrate the feasibility of a workflow, used further to predict and optimise the aqueous battery chemistries.

1.2 How Aqueous Batteries Work

An aqueous battery relies on a suitable salt dissolved in water. The higher the concentration of ions the higher the ionic conductivity and higher the voltage. However, as the concentration increases, the ions start forming ion pairs and larger ionic complexes and as the size grows the conductivity slows down resulting in decreased ionic conductivity. This leads to an optimised salt concentration to achieve improved ionic conductivity. Even though higher ionic concentrations might slow down ionic conductivity, they protect the water and stop the decomposition of the water. This in turn increases the working potential of the battery, which is a property that needs to be increased for the utilisation of aqueous batteries at grid level. This can be shown in the voltage difference between water in salt and salt in water batteries as seen in **Figure 3**.⁷

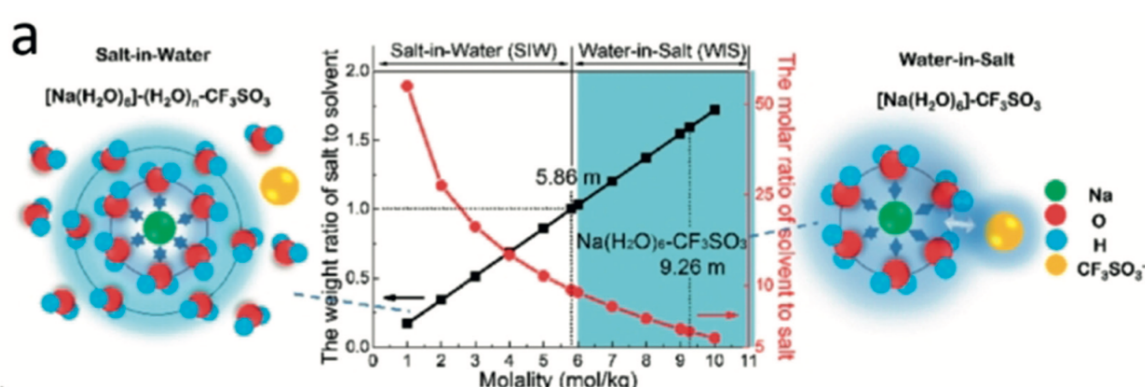
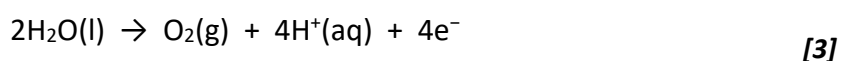
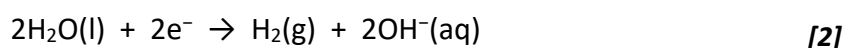


Figure 3 - Salt to solvent molar and weight ratios for the NaOTf–H₂O binary system with corresponding molality. Image reproduced from Zhang et al. (2021) (ref. 7)

When a potential is applied, the bonds in the water break and form OH^- and H_3O^+ species, which further breaks down into hydrogen and oxygen gas. The equation for the overall reaction can be seen in [1]. Whereas the cathode half reaction can be seen in [2] and the anode half reaction in [3].



This is not ideal for the system as the produced gas can be flammable.⁵ In addition, salt in water (SIW) electrolytes fail to reach higher potentials because the extensive hydration spheres around the cations and anions, which will break down forming oxygen and hydrogen gas. While in the water in salt electrolytes (WISE) the water molecules are protected by the ions in solution, reducing the breakdown of the water.⁷

The properties of aqueous batteries are complex and the reaction mechanism depends on the battery chemistry. In this thesis the zinc anode will be used as an example. There are two types of aqueous rechargeable batteries which focus on different principles. The first one of the chemistries is referred to as the rocking chair formation or sometimes as the intercalation-chemistry type.⁹ This is the chemistry used in the original aqueous rechargeable batteries in 1994 by Wainwright and coworkers using LiMnO_4 (LMO) as the anode and VO_2 as the cathode

in a 5M LiNO_3 electrolyte.¹⁰ Further research has given rise to different, more complex chemistries. These are referred to as the hybrid design and work via coupling an intercalation cathode with a metal anode or combining an intercalation anode with a metal oxide/sulphide. The reason these chemistries are different to traditional methods is because the batteries operate based on two reversible electrochemical redox processes, which involve the anode and cathode separately. The electrolyte acts as conducting ions and act as the energy storage component by undergoing reactions involving the electrodes rather than just acting as guest ions. Instead, the reaction of zinc deposition-dissolution and/or proton-induced oxidization/reduction occurs. The comparison between the reaction scheme of these chemistries can be seen in **Figure 4.a.**⁹ **Figure 4.b** shows the zinc deposition-dissolution reaction and **Figure 4.c** shows the proton-induced oxidization/reduction reaction. These hybrid chemistries are used in batteries referred to as aqueous rechargeable hybrid batteries (ARHBs).

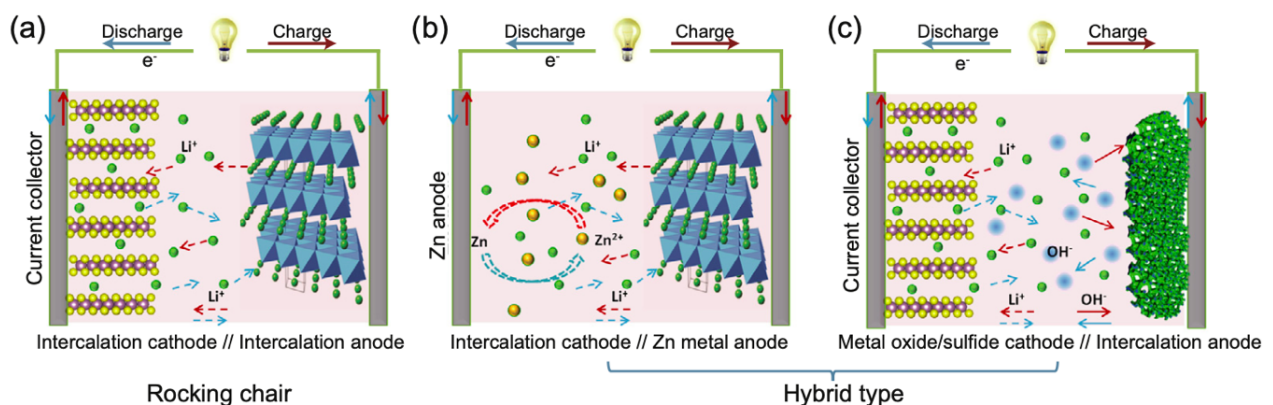


Figure 4 – The “rocking chair” and “hybrid” type aqueous rechargeable batteries – shown schematically. Image reproduced from Liu et al. (2018) (ref. 9)

The variation in the suitable chemistries used for aqueous rechargeable batteries leads to many different materials being used, which also means that the chemistries for different batteries are not uniform. This makes categorising different battery materials challenging. **Table 2** introduces different cathode materials used in aqueous zinc ion batteries.¹¹ **Table 2** compares the differences in potential range, initial discharge capacity and current density for different combinations of cathodes and electrolytes, whilst keeping the zinc anode unchanged. From **Table 2**, it is shown that there are many different combinations of cathodes and electrolytes available when studying zinc anodes leading to a large number of combinations available when the anodes is also changed. This gives rise to many more combinations which can be studied and explored. **Table 2** is used as a skeleton template for the desired results produced as an output from the python script, created in this project. In the method section, you will see how this template was also used when building the manual database. This was so that the format of the results obtained were presented in a consistent way.

Table 2 - Electrochemical performances of different cathode materials for aqueous zinc-ion batteries. Table reproduced from Selvakumaran et al. (2019) (ref. 11)

Cathode material	Electrolyte	Potential range (V)	Initial discharge capacity (mA h g ⁻¹)	Current density (mA g ⁻¹)	Ref.
MnO ₂	ZnSO ₄	1.0–1.9	187	100	28
α-MnO ₂	Zn(SO ₄) ₂ + MnSO ₄	0.8–1.9	405	150	106
α-MnO ₂	ZnSO ₄	1.0–1.8	233	83	44
α-MnO ₂ @C	ZnSO ₄	1.0–1.8	272	66	56
α-MnO ₂ /CNT	ZnSO ₄ + MnSO ₄	1.0–1.9	665	100	54
γ-MnO ₂	ZnSO ₄	1.0–1.8	285	50	52
β-MnO ₂	ZnSO ₄	1.0–1.8	270	100	133
δ-MnO ₂	ZnSO ₄	1.0–1.8	122	83	45
λ-MnO ₂	ZnSO ₄ + MnSO ₄	0.8–1.9	122	100	134
Todorokite MnO ₂	ZnSO ₄	0.7–2.0	98	50	43
Mn ₂ O ₃	ZnSO ₄ + MnSO ₄	1.0–1.9	78	100	58
Mn ₃ O ₄	ZnSO ₄	0.8–1.9	232	200	59
NaMnO ₂	Zn(CH ₃ COO) ₂ + CH ₃ COONa	1.0–2.0	60	120	135
ZnHCF	ZnSO ₄	–0.2 to 1.2	73.7	300	74
ZnHCF@MnO ₂	ZnSO ₄	1.4–1.9	118	100	136
CuHCF	ZnSO ₄	0.0–1.1	56	20	76
NiHCF	Zn(ClO ₄) ₂ in acetonitrile	0.6–1.8	55.6	60	40
Zn _{0.25} V ₂ O ₅ ·nH ₂ O	ZnSO ₄	0.5–1.4	282	300	84
Ca _{0.25} V ₂ O ₅ ·nH ₂ O	ZnSO ₄	0.6–1.6	340	60	86
CaV ₆ O ₁₆ ·3H ₂ O	Zn(CF ₃ SO ₃) ₂	0.2–1.6	367	50	87
Mg _{0.34} V ₂ O ₅ ·nH ₂ O			353	100	97
Li _x V ₂ O ₅ ·nH ₂ O	ZnSO ₄	0.4–1.4	407.6	1000	98
H ₂ V ₃ O ₈	Zn(CF ₃ SO ₃) ₂	0.2–1.6	423.8	100	88
H ₂ V ₃ O ₈ /graphene	Zn(CF ₃ SO ₃) ₂	0.2–1.6	394	100	89
V ₂ O ₅ ·nH ₂ O/graphene	Zn(CF ₃ SO ₃) ₂	0.2–1.6	381	60	85
V ₂ O ₅	ZnSO ₄	0.4–1.4	224	100	96
V ₂ O ₅	LiTFSI/Zn(CF ₃ SO ₃) ₂	0.2–1.6	238	50	9
V ₆ O ₁₃ ·nH ₂ O	Zn(CF ₃ SO ₃) ₂	0.2–1.4	395	100	99
NH ₄ V ₄ O ₁₀	ZnSO ₄	0.4–1.4	380.3	100	138
Zn ₂ V ₂ O ₇	ZnSO ₄	0.4–1.4	248	50	10
Zn ₃ V ₂ O ₇ (OH) ₂ ·2H ₂ O	ZnSO ₄	0.2–1.8	213	50	39
NaV ₃ O ₈ ·1.5H ₂ O	ZnSO ₄ /Na ₂ SO ₄	0.3–1.3	380	50	36
VO ₂	Zn(CF ₃ SO ₃) ₂	0.7–1.7	265	100	139
VO ₂ (B)	Zn(CF ₃ SO ₃) ₂	0.3–1.5	357	100	140
rGO/VO ₂	Zn(CF ₃ SO ₃) ₂	0.3–1.3	276	100	95
VS ₂	ZnSO ₄	0.4–1.4	190.3	50	100
Na ₃ V ₂ (PO ₄) ₃	Zn(CH ₃ COO) ₂	0.8–1.7	97	50	103
Na ₃ V ₂ (PO ₄) ₂ F ₃ @C	Zn(CF ₃ SO ₃) ₂	0.8–1.9	75	80	137
Mo ₆ S ₈	ZnSO ₄	0.25–1.0	79	45	109
ZnMo ₆ S ₈	ZnSO ₄	0.25–1.0	57	128	110
Calix[4]quinone	Zn(CF ₃ SO ₃) ₂	0.2–1.8	335	20	116
1,4-Bis(diphenylamino)benzene	LiTFSI/Zn(CF ₃ SO ₃) ₂	0.6–2.0	125	26	117
P-NiCo ₂ O _{4-x}	KOH + Zn(CH ₃ COO) ₂	1.4–2.0	309.2	6000	119
MoS _{2-x} nanosheets	Zinc trifluoromethanesulfonate	0.2–1.2	128.23	100	123

1.3 Aims and Objectives

Due to the voltage range, capacity and cycle life currently achievable with aqueous batteries, are not suitable for the same applications as Lithium-ion batteries. The desired characteristics which would make aqueous batteries more suitable would include high electrochemical performance, otherwise known as high energy density, long term sustainability and environmental sustainability.⁶ However, with the development of the following database and the completion of a centralised database, systems which receive a higher voltage range and

capacity can be identified without the need to build each battery system and cycle each battery physically in a lab. Using theoretical values can outline the dominant chemistries, which can then be used as a starting point for running practical experiments on the battery systems. This thesis will outline the steps in how this can be done.

With this thesis, the aim is to see if a more suitable and improved rechargeable aqueous battery can be designed and be used in the applications of research for car batteries, small portable devices or at grid level. To achieve the aims outlined below tools, such as databases and database analysing scripts were developed. The main focus of this thesis is on the objectives in order to create transferable tools, which could further be applied to different chemistries, making it a useful tool in research development. To summarise, the aims and objectives of the thesis are:

- To produce a proof of concept python script which can be developed to identify the most promising aqueous battery chemistries
- Build a script which can access already published articles.
- Create an initial centralised database of values extracted from already published articles.
- Develop code to automate the creation of a centralised database which can be further developed.
- To use extracted information to design better aqueous batteries

2. Method

In this section the tools and approaches used to build the databases and python scripts developed in this thesis are discussed in further detail. This section will provide detail on the approach taken to meet the aims and objectives discussed in the previous section.

2.1 Tools and Licensing

In order to complete this project, the chosen programming language was Python. Specific to the current project, the advantages of using python were the available scientific open source libraries, which included: NumPy, Matplotlib, TensorFlow and Pandas.¹² Also, python's ability to handle large amounts of numerical data was specifically useful due to the large amounts of literature that would be accessed by the programme.¹² Some of the general benefits of using python as a programming language include the simplicity of learning python, along with its ability to combine structures to achieve complex tasks.¹³ In comparison to other object oriented languages, python requires less lines of code to complete a task. This meant that using python was a more efficient way of building the programme.¹² Python is also an interpreted language making compiling and debugging easier as it will indicate the errors identified on the line.¹⁴ The final advantage that lead to python being the chosen language was the available resources, exemplar source code and tutorials available, specifically for scientific applications.¹³ However, the disadvantages that had to be considered for this task, when using python, was the speed that it would take to compile large amounts of code due to python being an interpreted language.¹² For this project the advantages were more impactful than the disadvantages.

The chosen integrated development environment (IDE) was Anaconda, as it includes 250 already installed anaconda packages, along with further 7500 machine learning and data science packages available over a cloud service.¹⁵ Anaconda allows for the easy management of multiple data environments. The version of Anaconda used was the python 3.7 as this was the recommended version for deep learning and artificial intelligence.

An important part of the project was to develop a web-scraping tool requiring access to scholarly articles. Despite many funding bodies adopting open-access policies, most scientific articles remain restricted. The University of Kent has identified a list of publishers (or journals) for which they pay a remuneration fee. Not having the correct licensing, will introduce limitations in the access to data required to populate the databases. University of Kent credentials were used to solve these problems. This was easier to do from the University of Kent campus, as direct access is only possible when the user is accessing the articles behind the University of Kent's firewall. This meant access was permitted without inputting credentials every time. This allowed for access to the articles for research and personal use but not for commercial use. A list of databases used included Web of Science and Google Scholar, which were accessed using Safari, which is the built-in web-browser available on Mac Operating System.^{16,17}

When accessing the articles generated by the Google Scholar search, rotating proxies were used. This is because Google Scholar can recognise that a call is being done by a computer programme and in turn block the request.¹⁸ Google Scholar employ the policy that only a certain amount of requests can be made using the same IP address.¹⁹ Proxy servers act as a middle ground between the user and the internet and maintains anonymity when making a

request. Rotating proxies ensure that the IP address is changed, and a different IP is chosen from a list of IP addresses before the user can be detected and blocked. Rotating proxies are particularly useful when running a high volume of requests continuously. The role of a proxy is summarised in **Figure 5**.²⁰

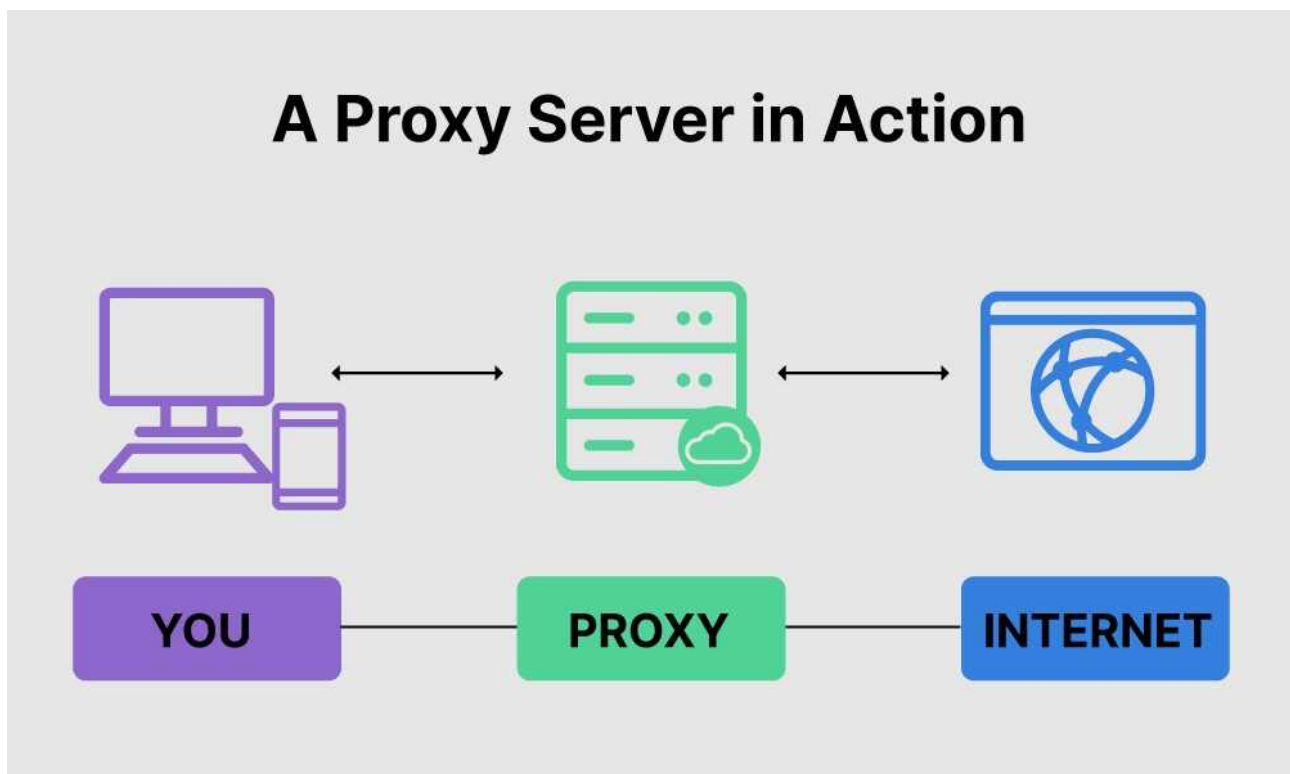


Figure 5 – Role of a proxy in accessing internet browsing – shown schematically. Reproduced from Fortinet (ref. 20)

In python coding rotating proxies can be used in several different methods. One of the approaches is by using the BeautifulSoup project and passing the user agent headers to simulate a browser call.²¹ This works along with the *requests* module in python. BeautifulSoup is a python project developed for the parsing of data, which allows the user to interact with dynamic websites. Using *requests* returns both html as well as JavaScript codes. BeautifulSoup manage the information and codes collected using the *requests* library by executing the

codes in a similar manner as to a web-browser, returning user friendly information. Another method of doing this is using SerpApi. Application programming interfaces (APIs) are software developed for the exchange of data, as well as functionalities, securely and effortlessly. As a result, many queries can be handled without restrictions, including access to information presented as JavaScript and HTML codes. This approach requires the use of a further extensions and there are many possible APIs to choose from.

It is important to note that the need for proxies is also important when the programme is being built, as this is when there are frequent and continuous requests for the purpose of testing. This is less likely to affect the end user as they are less likely to do as many requests as possible in a short amount of time, therefore Google Scholar is less likely to block the search for the user. However, the program will be more robust if these are still built in.

2.2 Google Scholar as the Source of Literature

Initially, the project proposal intended the use of the Royal Society of Chemistry (RSC) database as the single source of scientific articles accessed, and web scraped by the software developed in this project.²² The justification for this approach is because the RSC is an accredited society with peer-reviewed publishing policies. The scientific content of the journals in the RSC database can, therefore be considered as a reliable resource. This idea was abandoned due to limitations performing a data search directly from the RSC webpage. As journals on the RSC website are displayed in an A-Z format, this complicates the web-scraping and could potentially lead to the web scrape not being successful and a reduced variety of articles, focusing on aqueous batteries being accessed. Even though there are

around 115 different journals available on the RSC website, the format of how these journals is accessed and how articles are displayed on the website, is seen in **Figure 6**. This format

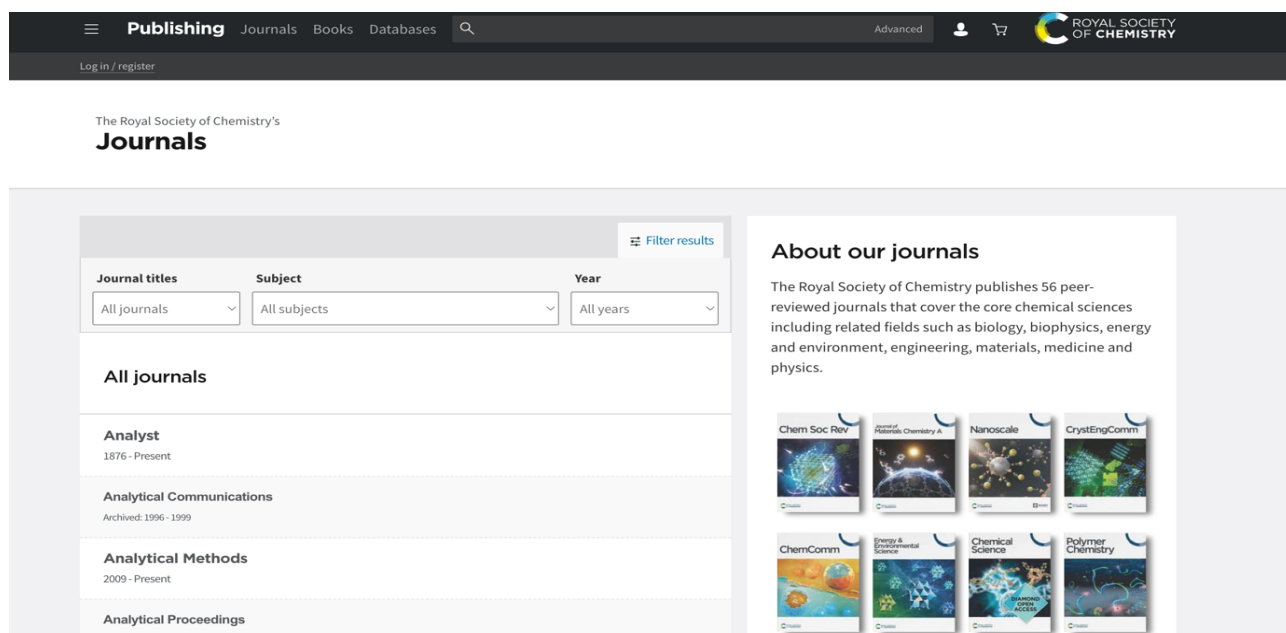


Figure 6 – Royal Society of Chemistry Webpage showing the webpage which would need to be web scraped if the RSC website is the source of literature.

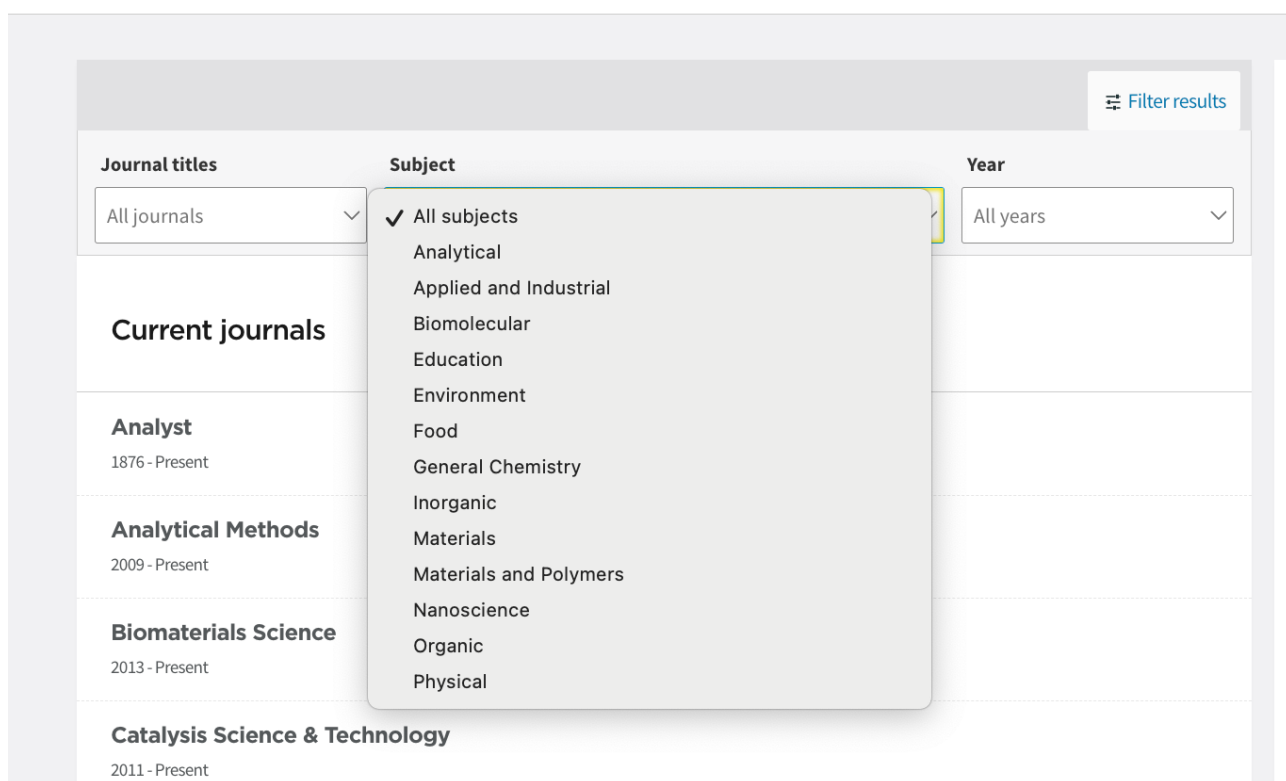


Figure 7 – Royal Society of Chemistry website – showing the search filters already built into the website.

This is because each journal would need to be searched with the key words separately to find the relevant articles and this would need to be done by the program as there is not a way to search and display all articles containing a key word, for example 'Aqueous' using the RSC website. Instead, the filter options when searching for results on RSC can be seen in **Figure 7**.

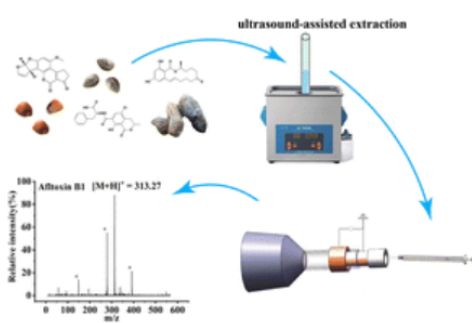
This meant that more complex programming would be needed in order to use RSC as the source of articles and therefore more time and resources. However, this was still considered due to the format in which RSC displayed article details (**Figure 8**) being more detailed and how easy it was to access the PDF versions of the articles.



Rapid detection of aflatoxin B1, zearalenone and ochratoxin A in grains by thermal desorption dielectric barrier discharge ionization mass spectrometry

Xiangxu Zhao, Xiaokang Guan, Qiao Lu, Xiaowen Yan and Renato Zenobi

Extraction followed by thermal desorption and dielectric barrier discharge ionization-MS is able to quantify mycotoxins in grains down to the ppm level.



The article was first published on 22 Aug 2024

Analyst, 2024, Advance Article

<https://doi.org/10.1039/D4AN00977K>

Download PDF

Article HTML

Figure 8 – Royal Society of Chemistry website – showing the final webpage accessed before a PDF copy of the article could be downloaded.

In order to prove that the computer program was successful in mining relevant data, there would need to be large observational data sets, *i.e.* a training sets, which would need to be manually built as a baseline comparison.²⁴ Using the RSC search engine introduced the complications outlined above making it harder to build both the manual database and the programme produced database. Moreover, in order to view publications stored in the RSC database the user is requested to undergo a “challenge-response” authentication.²⁵ Challenge-response authentication is the use of a set of protocols that helps validate access to protect digital assets and services from unauthorized access. This protocol usually asks the user a question and requires a response, the user must provide a correct answer for authentication to be complete. Challenge-response protocols can also be a password or a dynamically generated request.²⁵ Due to the time constraints of the project and the complications outlined above, the use of RSC was decided against and a more suitable search engine to access articles was used.

Instead, larger search engines (databases), such as Google Scholar and Web of Science, were used as alternatives.^{16,17} When determining which of these search engines would be the most suitable, a variety of things were considered. These included: *i)* the number of articles; *ii)* more recently published articles; and *iii)* citation status. To determine which search engine yields more results, identical searches were performed on both sites. The results in **Figure 9** show that Google Scholar consistently produced more results and had access to a larger variety of articles, also allowing the reader to preview more of the abstract in comparison to Web of Science. Furthermore, the accessibility of these articles using a web scrape program was considered. This included whether the format of the webpage was suitable to web scrape using the resources for this project. The results shown in **Figure 9** were obtained by

conducting different searches with different key words and parameters and comparing the number of articles returned for each search made. Google Scholar produced more articles for each search showing that there were more articles available if Google Scholar was used. However, this did not determine if all the search results were relevant to the chosen research area of aqueous batteries. Further analysis into the search results would be needed to determine this.

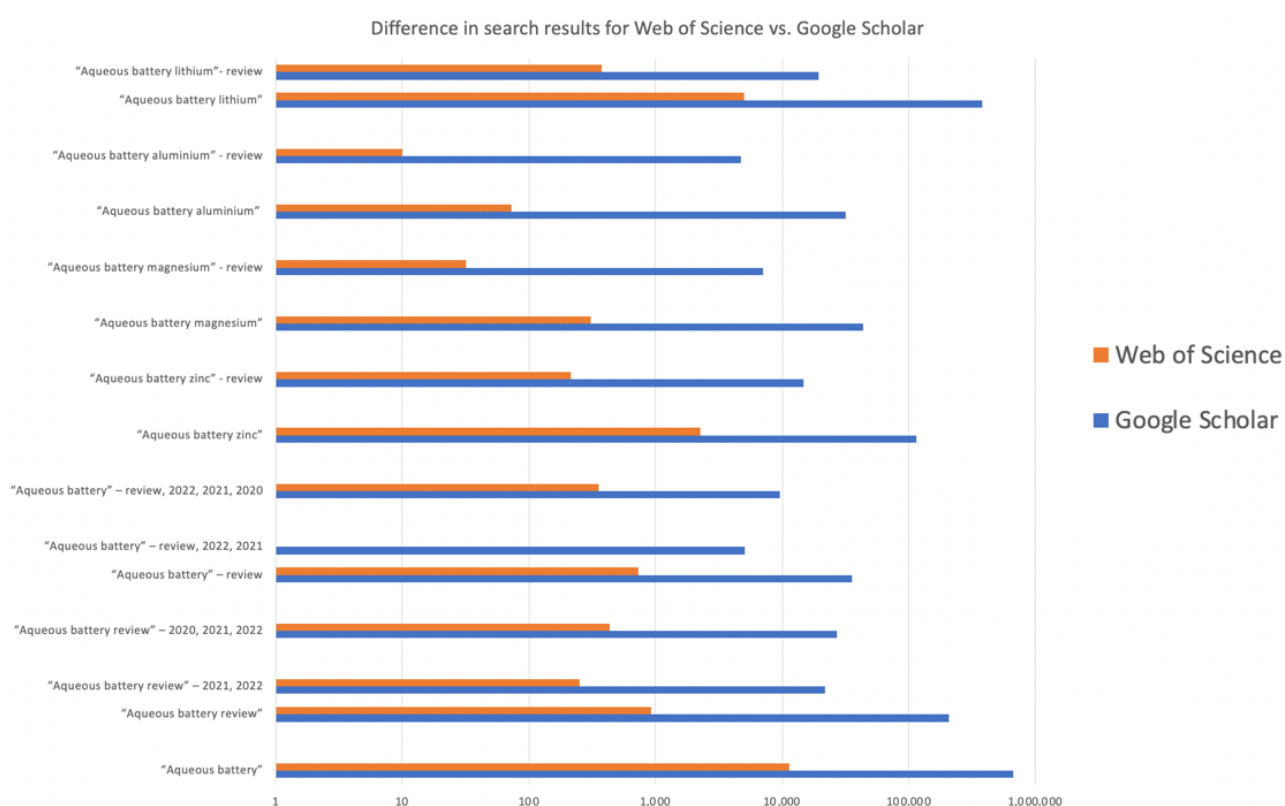


Figure 9 – The difference in the amount of search results for Web of Science in comparison to Google Scholar for different search parameters.

Based on the first 50 results of some of the searches, Web of Science had a higher percentage of relevant articles. This was discovered by manually analysing and reading parts of the first 50 articles and abstracts on both Web of Science and Google Scholar. This was done to identify

if the articles were about aqueous rechargeable batteries and included values for potential range, initial discharge capacity, and current density, which were used to build the training data set (manual database). In **Figure 10** you can see the first 50 articles under different search parameters were looked at. The percentage of how many of the first 50 articles were relevant to aqueous batteries is shown in a bar graph.

From these results we can see that Web of Science had results that were more targeted to aqueous batteries in comparison to Google Scholar. For example, Google Scholar would show articles that were related to non-aqueous batteries as these articles also contained “aqueous” in the title. The results from Web of Science produced more relevant articles which could contribute to building a database storing the outlined chemical properties, however this was only based on the first 50 articles and not all results. This was a factor to consider when deciding which search engine would be the most suitable. Both the quantity and the quality of the articles would need to be considered.

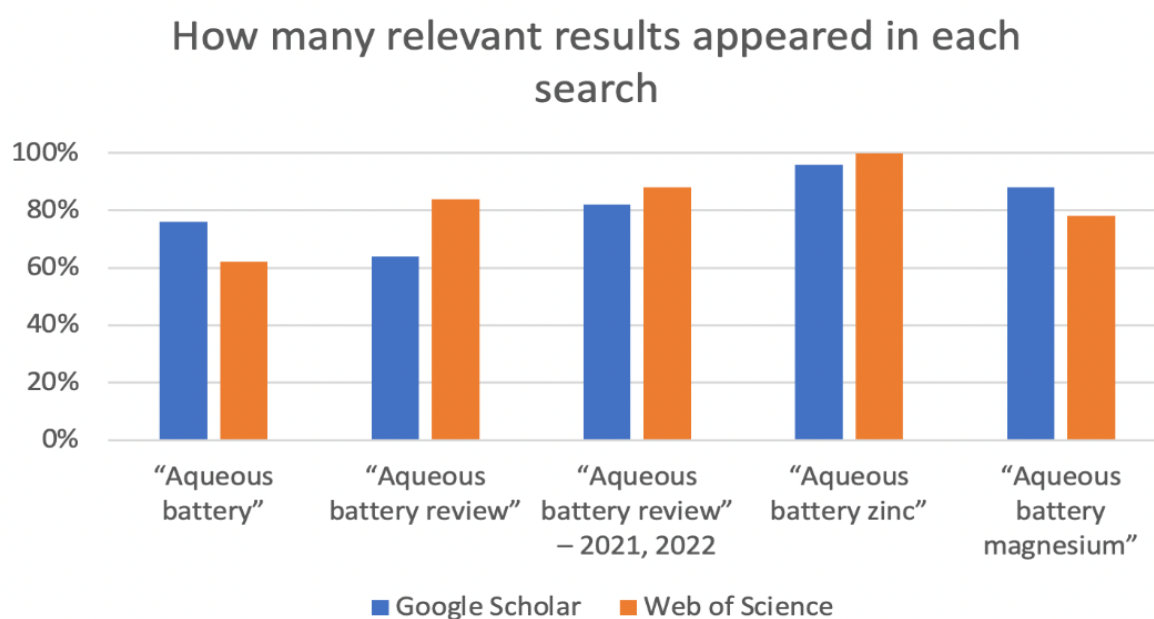


Figure 10 – Percentage comparison of relevant articles produced from different search parameters for Web of Science in comparison to Google Scholar.

Another factor which was considered was how many times the resulted articles were cited within other articles (not including self-citation). This would assess the popularity of the search results shown. Google Scholar showed more citations for each article as opposed to Web of Science. This meant that Google Scholar was a more utilised platform, meaning it could potentially have a wider range of available literature. The difference in the number of citations is shown in **Figure 11**. This was not a “high weight” determining factor as the results were relatively close.

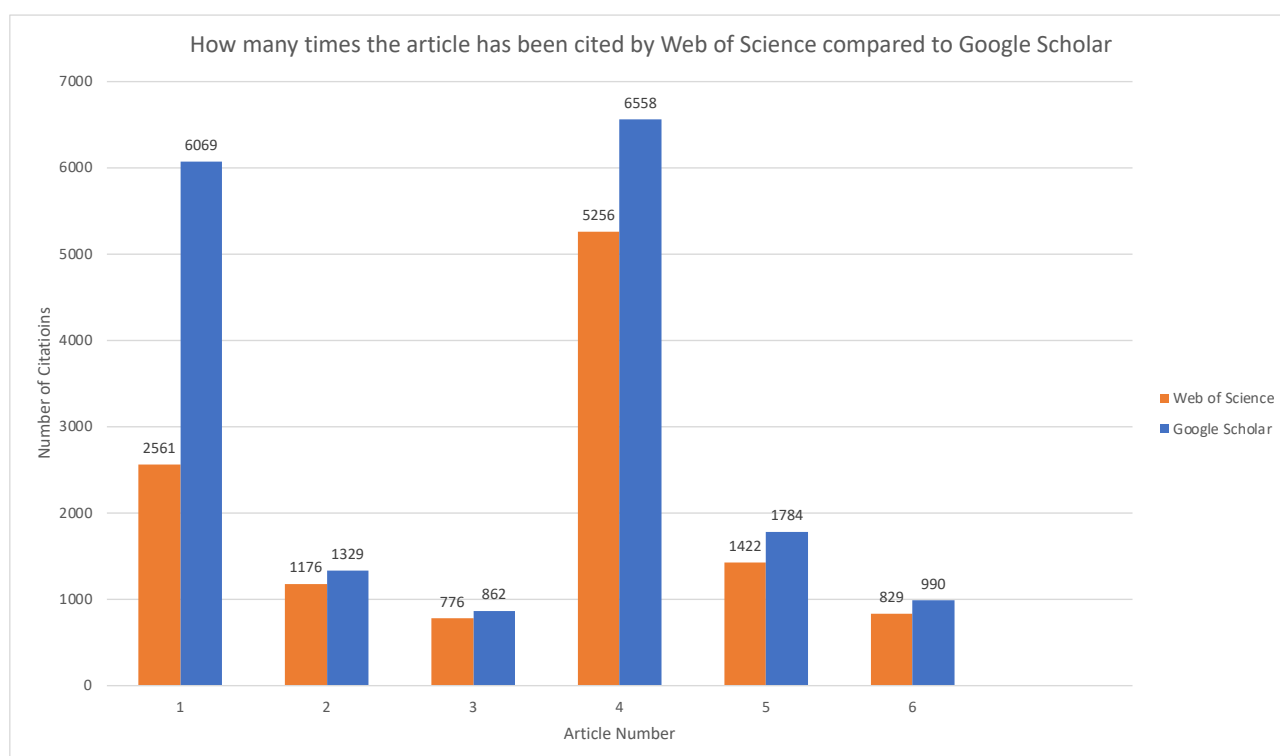
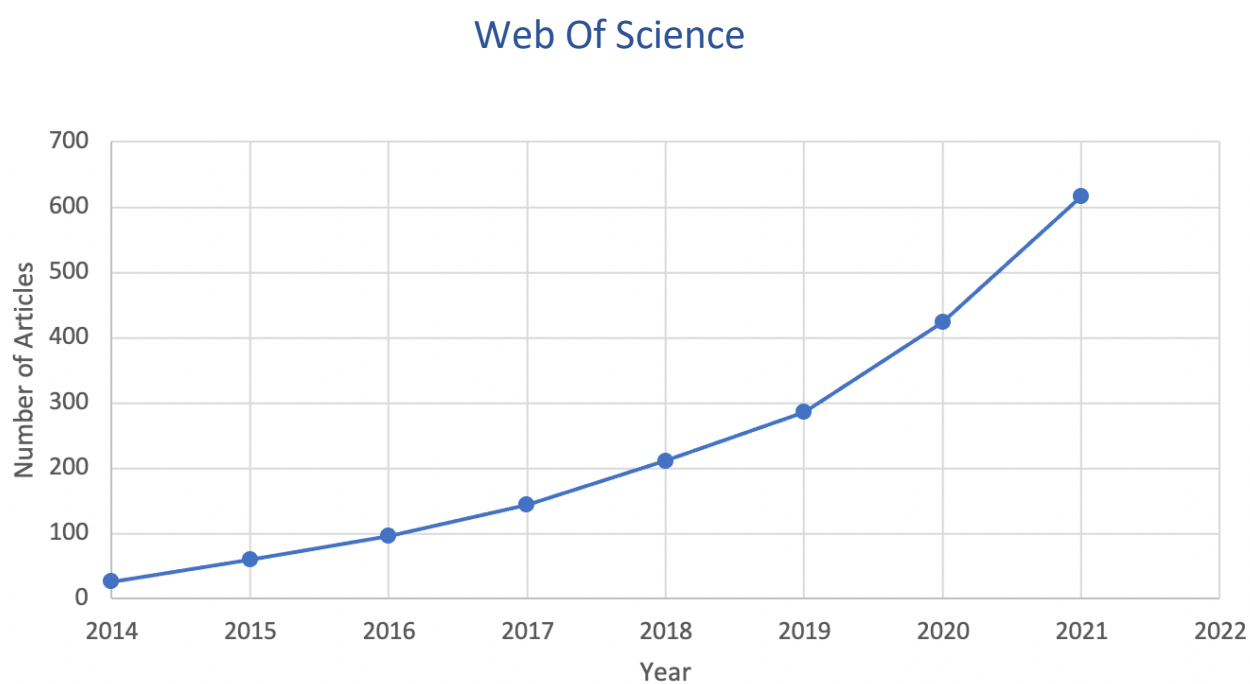


Figure 11 – Comparison of how many times a specific article was cited by other articles. This was used to determine how frequently others accessed the same article via Web of Science in comparison to Google Scholar.

Additionally, the rate in which number of articles in the field of “aqueous batteries” increased since 2014 was considered and it was found that it is higher for Web of Science than for Google Scholar. Since 2014 the number of relative articles increased by 24 times for the Web of Science (WOS) data base, whereas for Google Scholar (GS) this was only by 11 times. Nonetheless, regardless of having a larger increase, Web of Science did not have nearly as many articles available as Google Scholar. As seen in **Figure 12.a** and **Figure 12.b**, Google Scholar (GS) showed 10 times more articles available in 2021, i.e. 600 and 6200 articles for WOS and GS, respectively.

a)



b)

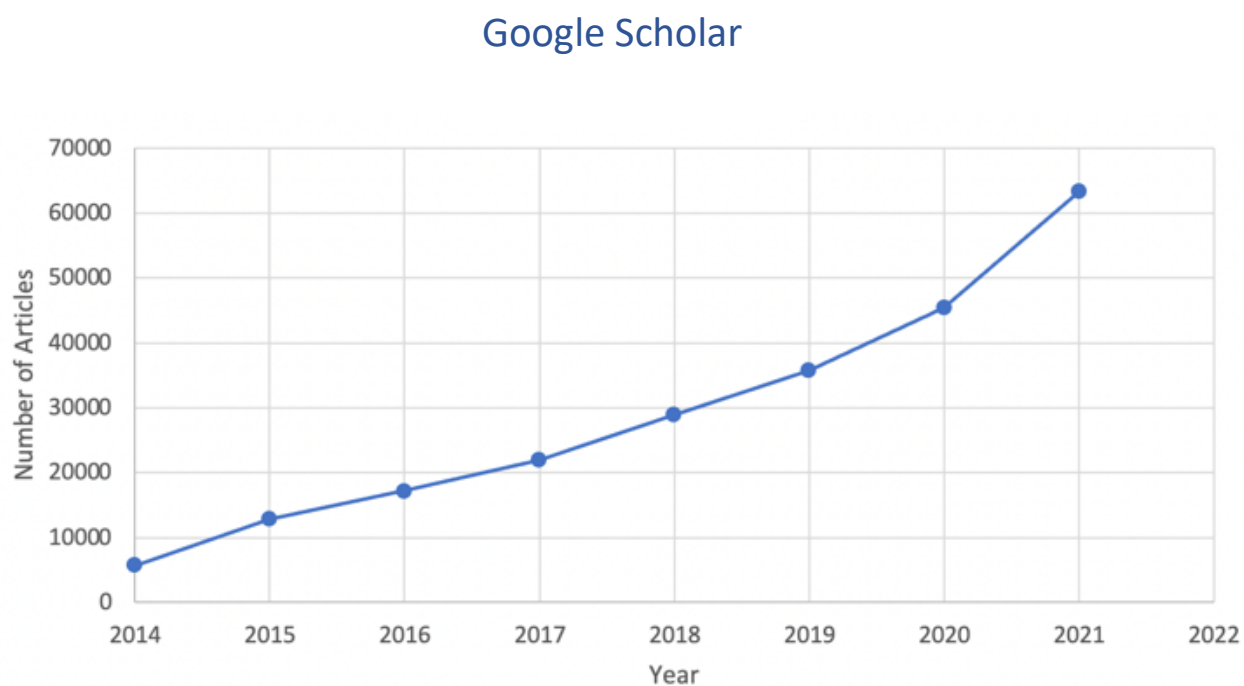


Figure 12 – The number of total articles published on ‘Aqueous Battery Review’ between 2014 and 2022 for **(a)** Web of Science and **(b)** Google Scholar.

An advantage of using the Web of Science data base is the query function, which allows the user to combine different searches to give a result that contains either all the search words or just some. Whilst this feature, along with Web of Science providing more relevant articles, was an advantage for creating a manual database, it was not a required feature when building the programme. Although this is an advantage to achieve the set out aims, this was not a vital feature required by the search engine, as the script will be programmed to do these routines and search criteria, to identify and collate the relevant articles. Concluding, this meant that the disadvantages of using Web of Science outweighed the advantages.

According to the definitions of data mining, "Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data".²⁶ The purpose of the search engine is, therefore, to provide access to the largest amount of data possible. Based on the latter criteria, using Google Scholar ensures that the program can access larger amounts of data than Web of Science. Furthermore, Google Scholar overcomes an issue with entering credentials to access articles. If the program is accessed from the University of Kent campus VPN, no passwords will be required when using Google Scholar, however this was not the case for Web of Science as at times this was not fool proof, and the website would still require credentials. Google Scholar was the chosen source as it would contain the articles available from the Royal Society of Chemistry and Web of Science, whilst also allowing for the program to be adapted to have a wider application in different specialties. Google Scholar also has a built-in function to allow for non-keywords, such as "non-aqueous" to be excluded from the search. This exclusion was done by both Google Scholar and the program. However, when querying Google Scholar via requests, the web

scrape eventually faces IP blocks. To overcome this, rotating proxies were required. This is later explained in greater detail.

This work incorporates automated data extraction from Google Scholar to enable large-scale identification of relevant literature on battery materials, which is central to the proposed optimization framework. While this approach provides broad coverage and rapid access to publicly available academic data, it operates within certain practical and procedural constraints, as automated querying is not formally supported by the platform's usage policies. To ensure responsible use, data collection in this study was conducted conservatively, with limited request rates and a focus on non-sensitive, publicly accessible information, in line with applicable data protection regulations such as GDPR. Nevertheless, this introduces a methodological limitation: the stability and reproducibility of the data pipeline may be affected by potential changes in access conditions or platform structure over time. Accordingly, the approach should be understood as a proof-of-concept demonstrating the feasibility of automated literature-driven discovery in the context of battery materials research. Future work could strengthen robustness and long-term applicability by integrating officially supported data sources, such as publisher APIs, CrossRef, or other open scholarly databases, thereby enhancing transparency, reproducibility, and compliance while maintaining the benefits of large-scale data-driven analysis. The conducted research is outlined as a proof of concept, showing that web scraping and data mining can be used to populate databases. This can be further developed with the assistance of artificial intelligence tools which are advancing very quickly.

2.3 Building a Manual Database

As a validation for the database that would be created by the program, a database was also built using manual searches to find articles. The manually built database was used to benchmark the program generated database. The benchmark criteria, when creating the manual database and filtering relevant articles, is outlined below. This is the same criteria required for the results produced from the program, therefore we would expect the program to automatically filter a range of articles in the same way to output relevant literature. The relevant articles should include the following words in the title or abstract:

- Aqueous
- Battery/Batteries
- Rechargeable
- Salt – water electrolyte

However, the following “non” key words would not appear in the title or abstract:

- Flow
- Non-aqueous
- Air
- Redox

These results still produced a vast amount of search hits, which meant that the results were further categorised into articles, either focusing on the cathode, anode, electrolyte or the system as a whole.

These results were further categorised by the materials that the articles were focusing on for each cathode, anode and electrolyte. A schematic diagram showing the categorisation can be

seen in **Figure 13**. This would allow for the user to further analyse the search results by identifying the most common materials reported in the literature.

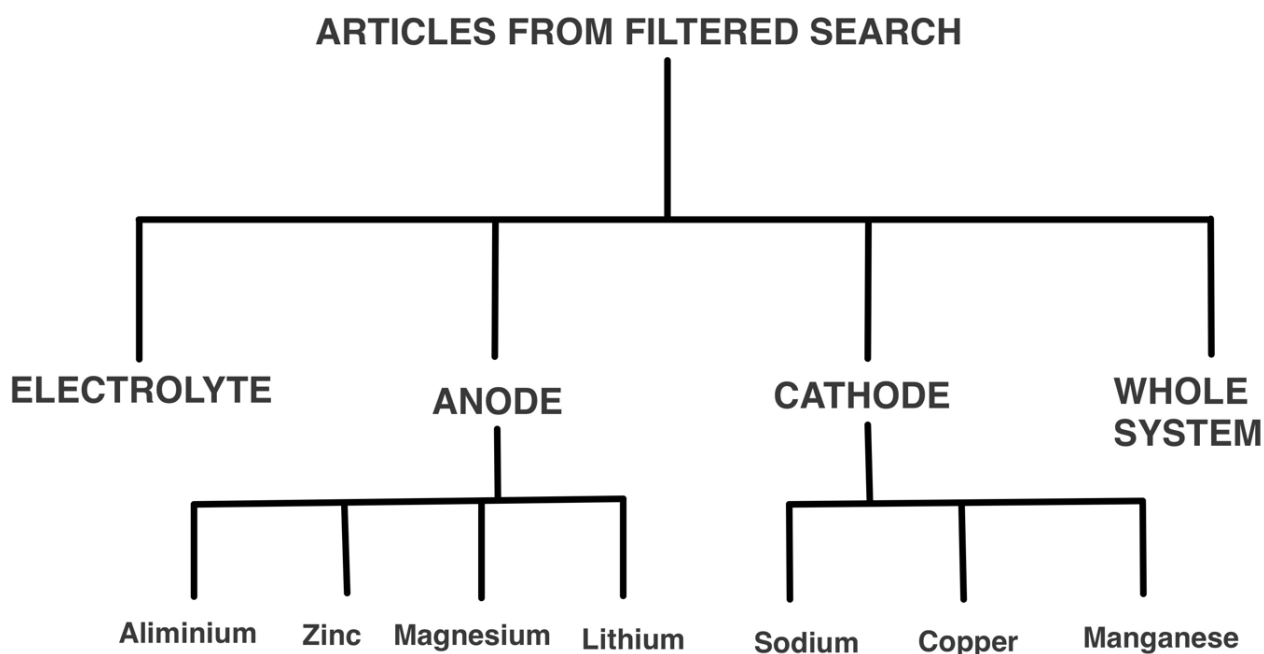


Figure 13 – A flowchart of how the articles from the search results were manually sorted and categorised to differentiate between anode and cathode materials.

However, to create a database from these articles, we would also need to identify which articles contained relevant properties on the battery systems. These properties should be stored in the database as variables. Examples of properties include capacity, capacity retention, and current density. This was a more complex task as it meant that each article would need to be read or skimmed for values. With the manual database, this would require more time and allowed space for human error. The objective of the code written is to streamline this process and make it more reliable. The script aims to automatically create a table similar to **Table 3** based on the user's search criteria.

The benchmarking exercise also served as a tool in determining the method that the web-scraping program would use in constructing a reliable database. The benchmarking process included the list of keywords that would be included in the search, and which non-keywords would be considered in order to explore and refine the search to include the most relevant articles. The manual database was built by searching “aqueous battery” and refining the results by removing articles containing the non-keywords in the title or abstract. The non-keywords identified in the manual search were: “non-aqueous”; “redox”; “air”; and “flow”. By completing the exercise for the manual search, possible limitations which were not previously considered were identified. It was found that extra care had to be taken when excluding articles that contained “non-aqueous” in the title or abstract, as some articles had information about both aqueous and non-aqueous batteries in the same article. The conclusion was that this would need to be considered to not lose valuable data. A manual solution to this was that when skimming the articles, if the article had both aqueous and non-aqueous within the title or abstract then it would still be considered as a relevant article. In order for the program to automate this process, a condition was applied to still include articles containing ‘non-aqueous’, only if the title or abstract also contains the word ‘aqueous’. As an option to implement this, machine learning and AI could be incorporated into the program. This would require further research and more scope for the project.

Table 3 – Partial representation of the main manually built database table - produced as a guideline and training set for the program produced database.

Anode	Cathode	Electrolyte	Potential range (V)	Capacity Retention (%) (per cycle)	Current density (mA g ⁻¹)	Energy Density (W h kg ⁻¹)	Capacity (mA h g ⁻¹)	Reference
Zn	MnO ₂	ZnSO ₄	1.0–1.9	187	100			3
Zn	α-MnO ₂ (alpha)	Zn(SO ₄) ₂ + MnSO ₄	0.8–1.9	405	150			3
LiTi ₂ (PO ₄) ₃	LiFePO ₄	Li ₂ SO ₄ (1 M)		90% (1000)	6.0 C		55	7
VO ₂	LiFePO ₄ /C	LiNO ₃ (Saturated)		94% (50)	C/3		106	7
Mo ₆ S ₈	LiMn ₂ O ₄	LiTFSI (21 m)	2.3			84		14
Mo ₆ S ₈	LiFePO ₄	LiTFSI (21 m)	1.3			47		14

When relevant articles were identified, they were scanned for the expected information to build the database. This included information on anode and cathode materials, electrolyte, potential range, capacity, capacity retention, current density and energy density. This was then put into a table with the correct electrode and electrolyte combinations and their relevant properties. The table also contained a key to show which article the data values were taken from for reference purposes, this is the last column title 'Reference'. A section of the first couple of rows from the manually built database is shown in **Table 3**. The full table can be found in **Appendix 1**.

As a comparison data set, a table similar to **Table 4** should have been created based on the manually obtained articles. **Table 4** is a table created from the results produced by the python script. This would mean a database of the Title, Abstract and URL of the relevant manually obtained articles should have been produced. This would mean that we would be able to do a comparison of results earlier on in the development process. However, the format in which the program would extract details from the articles was only established once the python script was created.

Table 4 – Results showing the associated variables obtained from the web scrape for each article.

Title	Abstract	URL
A 63 m Superconcentrated Aqueous Electrolyte for High-Energy Li-Ion Batteries²⁷	A water-in-salt electrolyte (WiSE) offers an electrochemical stability window much wider than typical aqueous electrolytes but still falls short in accommodating high-energy anode materials, mainly because of the enrichment of water molecules in the primary solvation sheath of Li ⁺ . Herein, we report a new...	https://pubs.acs.org/doi/10.1021/acsenenergylett.0c00348?ref=pdf
A review on recent developments and challenges of cathode materials for rechargeable aqueous Zn-ion batteries¹¹	Owing to their high cost and safety hazards, and the low abundance of Li in natural resources, the future of Li-ion batteries is becoming difficult. To replace Li-ion batteries, aqueous multivalent ion batteries are considered to be a worthwhile choice. In particular, aqueous zinc ion batteries are becoming an attractive option due to...	https://pubs.rsc.org/en/content/articlelanding/2019/ta/c9ta05053a
Advances of aqueous rechargeable lithium-ion battery: A review²⁸	The electrochemical characteristic of the aqueous rechargeable lithium-ion battery has been widely investigated in efforts to design a green and safe technology that can provide a highly specific capacity, high efficiency and long life for high power applications such as the smart grid and electric vehicle. It is believed that the advantages...	https://www.sciencedirect.com/journal/journal-of-power-sources

2.4 Python Libraries

One of the main reasons Python is a widely used language in scientific programming, is due to the largely available libraries. Python has over 137,000 available libraries.²⁹ These are a collection of codes containing a pre-set of combined modules which can be reused for certain functions. These libraries can be imported when they are required and is beneficial to programmers as it is time efficient due to its reusability.

Some of the libraries that can be used for web scraping are BeautifulSoup, Scrapy, Selenium and Requests, among many others.³⁰ The functions of each library and the reasons to use each one for web scraping are outlined below:

- BeautifulSoup – BeautifulSoup is used to extract and navigate HTML content. It allows for the navigation, searching and modification of documents.³¹ It uses different Cascading Style Sheets (CSS) selectors to query the page and extract information.²¹ This will be useful when instructing the program to read in different parts of the HTML webpage. It will allow for only the relevant data to be extracted creating efficiency within the code.
- Scrapy – Scrapy can be used to extract data from websites by creating a website friendly spider. It allows for custom exporting options and can also be used to cache webpages and add parallelism to the produced code.³² For example, it can be used to create an array of all the URLs to the relevant articles. These can then be outputted to

the user so that the user can manually explore the articles that the presented data is extracted from.

- Requests – the reason requests provides useful in web scraping is because it allows the user to make HTTP (Hypertext Transfer Protocol) requests.³³ It provides a simple method of sending and receiving data from websites. It does this by sending a request to the website and by expecting a response. It also allows for the user to specify which fields of the webpage will be requested and fed back into the program. This is useful with interacting with webAPIs.³⁴ Examples of how the requests module is applied to complete tasks, will be explored later in this thesis.
- Selenium – Selenium is an open-source web driver which allows for the automatic navigation of webpages. It allows for different links and buttons on the webpage to be opened and clicked, therefore simplifying the navigation process through webpages. It is a library that is often used for automated testing of web applications.³⁵ Its main function of use was to eradicate manual testing in the software development life cycle.

Some libraries used in Data Mining include:

- NumPy – NumPy emerged in 2005 as a combination of the Numeric package and Numarray.³⁶ As one of the fundamental libraries used in scientific computing, NumPy executes high level mathematical functions efficiently.²⁹ It allows for efficient navigation of array-oriented programming. It can create powerful N-dimensional arrays (also known as tensors or matrices) using well optimised code for efficiency. This, combined with the high-level mathematical functions it has available, makes it a very valuable tool in scientific programming. An example of the importance of NumPy

is the role it played in the software stack used to discover gravitational waves and in the first imaging of a black hole.³⁶

- Pandas – Pandas is an open source library that is used for data analysis, data structuring and other machine learning activities.³⁷ It has been in development since 2008 and has created an efficient approach to working with multi-dimensional arrays. Pandas has simplified many data analytical tasks, including data cleansing, data normalisation, data visualisation and statistical analysis.³⁸ A feature of Pandas, specifically useful when it comes to data mining scientific literature, is its function to read and write data between in memory data structures and different formats.³⁹ This will allow for the different formats of information available online to be standardised and structured, making it easier to produce valuable data from the collated information online.
- SciPy – SciPy is an extension of NumPy, providing highly optimised algorithms for scientific calculations, such as integration, interpolation, eigenvalue problems, algebraic equations, differential equations and can also be used for multidimensional image operations.⁴⁰ This is a library that may not have direct function when data mining scientific literature until later in the programming stages, however it would provide useful when it comes to applying the data extracted, in order to obtain results from the information.
- Matplotlib – Matplotlib is used for creating static, animated, and interactive visualizations when using python.⁴¹ It uses very few lines of code to generate visual data representations. It also has functions that allow for the programmer to customise different dimensional plots, for example the type of graph, the axis, the colours and the layout. Some examples of the plots that can be created includes pie charts, bar

graphs and line graphs. When paired with NumPy's ability to execute mathematical functions, many different statistical analyses can be achieved. The reason this library is useful when creating a data mining program is because it allows for a concise representation of the results to be displayed to the end user. The results can then be further analysed, and valuable data can be extracted.

- TensorFlow – TensorFlow is a machine learning module which allows the user to use pretrained models or custom ones. It uses dataflow and differentiable programming to create deep learning algorithms and perform high level numerical functions.⁴² Machine learning is the programs ability to perform a task without explicit programming or rules.⁴³ TensorFlow gets its name due to its use of Tensors. A tensor is a vector or matrix of n-dimensions that represents all types of data.⁴² TensorFlow was created by Google and is used by many well recognised companies to incorporate machine learning in to their software. An example of this is how google uses TensorFlow to predict searches and provide recommendations to what could be the next word in the search.⁴⁴ This is a useful library for the program being created, as this will allow for the software to recognise valuable information in the web scraped materials, even when the information is not uniform and standardised. As not all information is presented in the same format, explicitly coding the program to recognise this information will be a challenge. TensorFlow will provide useful when trying to overcome this challenge.

2.5 Data Mining

Data mining is the process of discovering patterns, relationships, and insights from large sets of data. It involves extracting valuable information from vast amounts of data, often stored

in databases or data warehouses, and transforming it into a more understandable and usable format. Data mining techniques typically involve applying statistical and mathematical algorithms, machine learning, and artificial intelligence to analyse the data. These techniques can help identify correlations, associations, clusters, and trends within the data, which can provide valuable insights and actionable information.

Data mining is defined as the computational process of analysing large amounts of data in order to extract patterns and useful information.⁴⁵ Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying novel, valid, potentially useful, and ultimately understandable patterns in data.

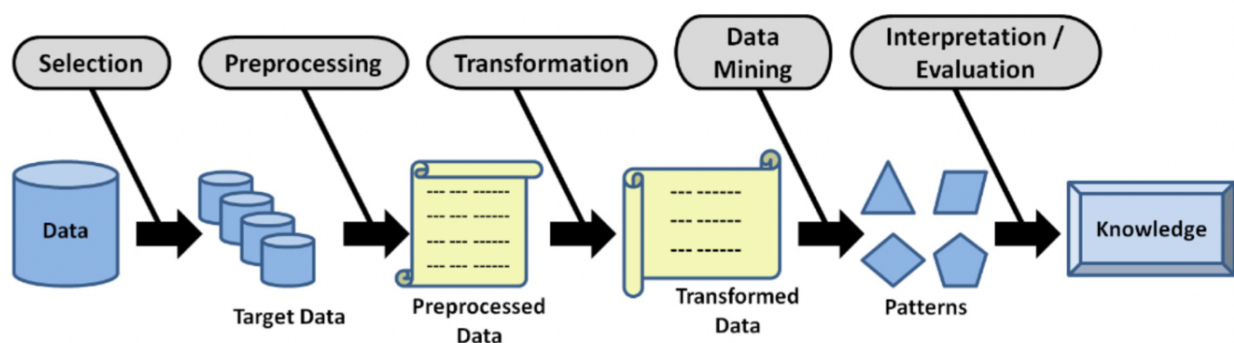


Figure 14 - Steps involved in Knowledge Discovery in Databases to extract valuable and relevant data. Reproduced from Johnson (ref. 42)

The process is an iterative and interactive sequence of the following main steps (**Figure 14**):

- Selection – this is the process of creating a specifically tailored data set from the main data set.
- Pre-processing - which aims to “clean” data by performing various operations, such as noise modelling and removal, defining proper strategies for handling missing data fields, accounting for time-sequence information.⁴⁵

- Transformation - In this step, the pre-processed data is converted into a suitable representation that enables effective analysis. It may involve converting the data into a different format, scaling or normalizing the values, and performing feature engineering to extract relevant features or create new ones.
- Data Mining - This step involves applying various data mining methods to the transformed data, to extract interesting patterns. Examples of some data mining algorithms include classification, clustering, association rule mining, summarisation and anomaly detection.
- Interpretation/Evaluation – This is the process of extracting knowledge and interpreting the patterns created. It is done by the user who can apply context to the patterns. This can be done by visualizing the results, evaluating the quality of the discovered patterns and identifying relationships and associations among the data.

The reason why knowledge discovery in databases and data mining are becoming increasingly more valuable, is due to the vast amount of data being produced and made available. To extract valuable data this process needs to be majorly automated and sped up. This must be done with a combination of tasks being delegated to both the program and the user. Once complete useful information can be extracted and exploited by humans. The extracted knowledge is very valuable and is used for important decision making. This is why it is important that the process is accurate and as precise as possible.⁴⁶ Data mining is a useful tool used in many industries for example banking, financial data analysis, marketing industry, telecommunication industry and many more.⁴⁷

2.6 Flowchart

To ensure that the script being produced is targeting the aims set out by the user, a flowchart was produced as shown in **Figure 15**. The flowchart in **Figure 15** outlines which tasks the program should complete and the order in which the tasks should be execute in. This is a useful way to plan which subroutines in the program will need to be built first and to create a project timeline.

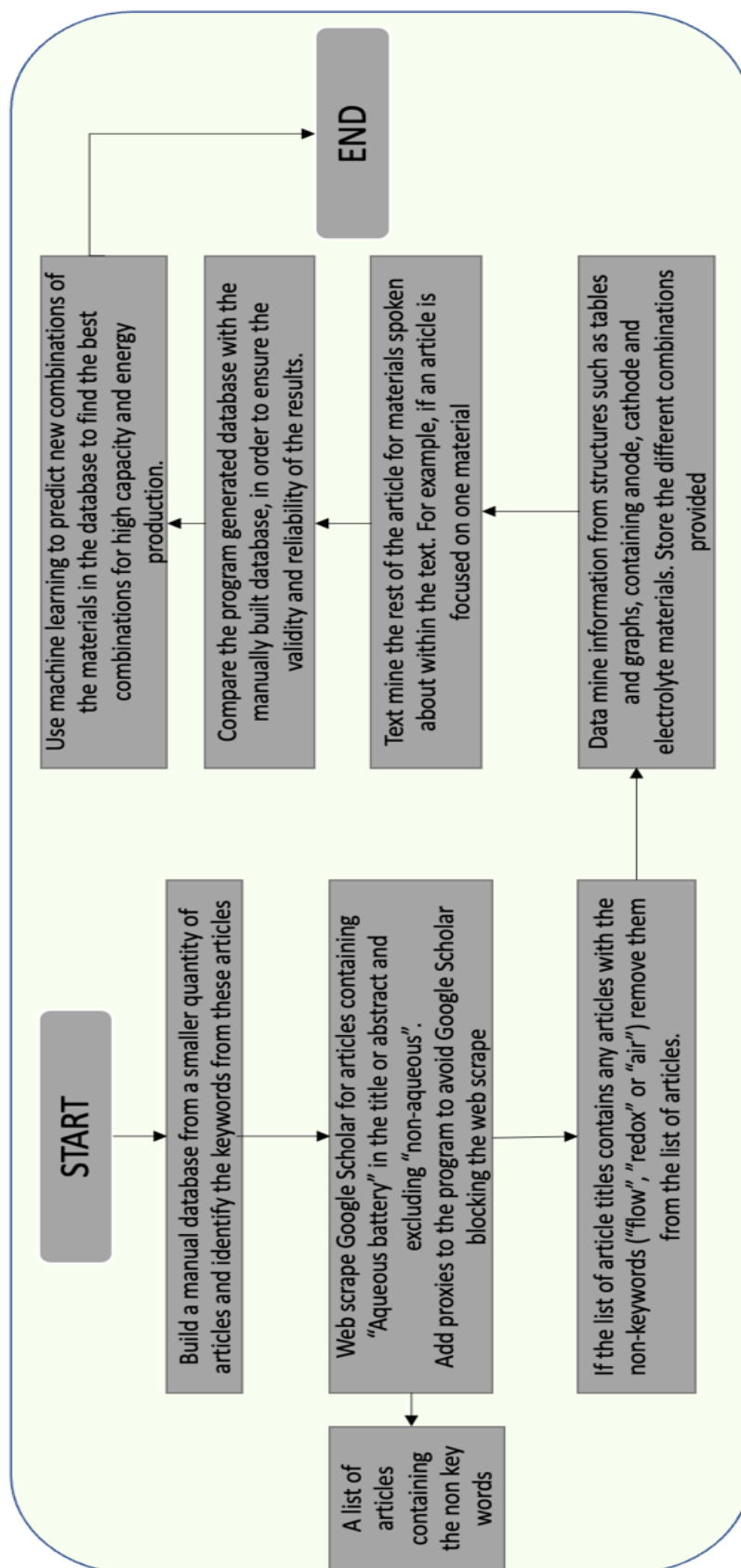


Figure 15 – The expected flow of the python script and the steps in which different tasks should be executed.

2.7 Method Conclusion

Overall, in this section, we were able to gather resources, knowledge and materials which were used to meet the aims of the project. A plan to meet the aims was established by creating a flowchart based on conducted research. Methods to meet the aims, such as web scraping and datamining, were identified and studied.

As well as using many articles which were available online on Google Scholar and other websites.¹⁷ A different approach was used to learn the technical skills needed to produce the code needed. One of the main ways this was done was by using online courses available on Coursera.⁴⁸ An example of a course used was the Data Mining Specialization course offered online by the University of Illinois at Urbana-Champaign.⁴⁹ The course aimed to teach data-mining skills that could be applied to both structured and unstructured data. These included topics such as:

- Pattern discovery
- Clustering
- Text retrieval
- Text mining and analytics
- Data visualization

Alongside this, another way research was conducted was by attending weekly meetings with the postgraduate MEE group to outline which areas of battery science did not have as extensive research. This then helped determine which field of research the code would be most valuable in.

3. Results

This part of the thesis discusses how the aims outlined in the introduction section were achieved as well as any limitations which were faced. It aims to determine whether the aims of the project are realistic and achievable.

3.1 Results from Code

To start building the required script, the first task was inputting the required libraries. The libraries used were Pandas, NumPy, Requests and Beautiful Soup. Pandas was used to create data frames to import web scraped data into a multi-dimensional array. The array was a 2-D array with 3 columns, i.e. a column each for the title, abstract and URL of each article. The number of rows corresponds with the number of articles found. For the purposes of building the script, only the first 100 articles were read into the script from Google Scholar. The reason for doing this was so that when unit testing was carried out, the code would be quicker and more efficient to run. This meant that test runs would not take as long. NumPy and Matplotlib were used to produce graphs from the results to compare the frequency of different anode materials that came up in the search. These results could then be compared to a manual search and sorting that was done separately by a human and used to see the accuracy of the script being created.

Requests were used to make a HTTP request from Google Scholar which returned a response object retrieved from the data from the URL. Next, the relevant fields to carry out the outlined research, such as the title of the articles, the abstracts and the URLs, were sorted and isolated from all the requested content. This was done using Beautiful Soup to parse the data and

isolate the different sections from the webpage. The code to implement this is shown in **Appendix 2**. The code also shows how rotating proxies were used; this was done by configuring individual requests. This was used because Google Scholar would block requests after the script was run a certain number of times. This was first noticed during the development testing when the program would produce an error message, and data would not be retrieved.

The webpage being web-scraped was a Google Scholar search for 'aqueous battery review' with a search parameter excluding 'non-aqueous'. The reason these were the chosen search parameters, was so that the search outputted review articles. This would mean the results did not show proposed experiments and rather experiments that had already been carried out. It also meant that many articles would contain reviews of multiple different materials as opposed to focusing on one. The reason 'non-aqueous' was introduced as a non-keyword at this stage was to do a first cycle of filtering out irrelevant articles. This is then repeated later in the code when the google results are requested into the program. However, the disadvantages of this are that some articles may still contain relevant information regarding both aqueous and non-aqueous batteries as previously discussed. This would mean that potentially useful and relevant articles would be excluded from the results. The origin data was the search results from Google Scholar, which was read into the program. The results showed the title of each article (for the first 100 articles) all on separate lines. This was also repeated for the URLs and Abstracts for each article. The program grouped all titles, abstracts and URLs separately. A schematic of how the data was handled is shown in **Figure 16**.

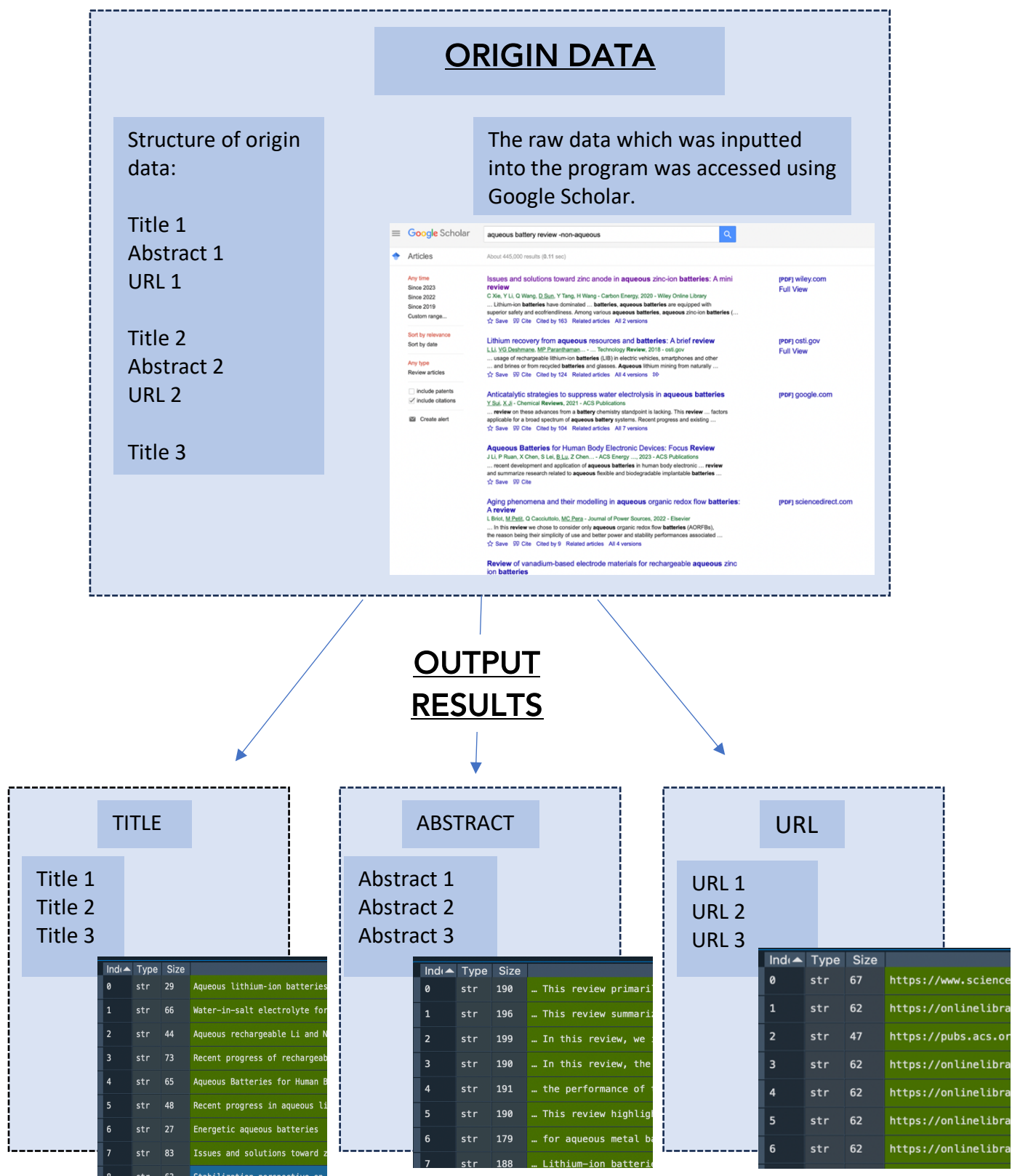


Figure 16 – How the data was handled and the format of the raw data obtained leading to the format of the processed data.

However, a problem faced with the reading in the abstracts, was that only a section of the abstract which previewed on the google search, was inputted into the code. This was due to how the webpage was being scraped and the way in which information was read into the script.

In **Figure 16** it can be seen how the program also reports back the index of each variable and the length. This is useful for our purposes as it would allow us to reference the results and see which abstracts were longer and subsequently could contain more detail about what the article is about. For example, some abstracts contained information on the type of batteries and the materials focused on in the research. These could then be read into a file and utilized to identify articles of interest and to compare to the manually created database. This exercise was done to validity the results obtained by the programme and to prove agreement with the manually created database.

Information was read into the script by using the web page tag for each section on the website, that we were interested in. The tags for each section were determined using Google Chrome Developer tools.⁵⁰ This tool allows you to view the HTML code for each element on the web pages as seen in **Figure 17**.

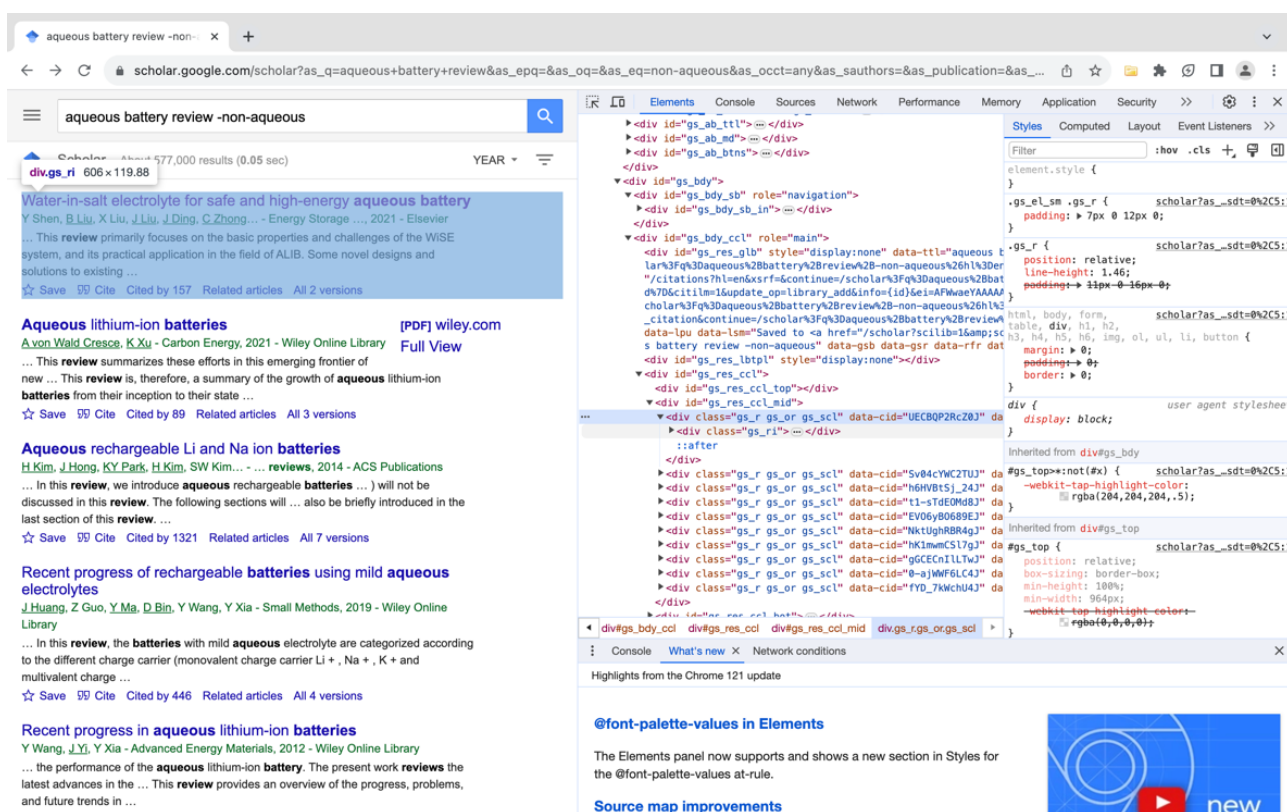


Figure 17 – A Google Chrome tool which allows you to view the HTML tags for different elements on a webpage. Useful in determining the tags for the elements that being web scraped.

To ensure that results from the next page of search results were also inputted, a loop was incorporated into the code so the page number in the URL would increase every time data was read from a page. This meant that the data from the next page would be read into the program. To navigate webpages in more complex ways, web scraping techniques would need to be used. This advancement in the code would enable the program to access more varieties of data sources, such as webpages, and in turn source more data. More advanced web scraping techniques would also mean that individual articles could also be scraped, and data mined.

Once the information was web scraped into the program, the abstracts were searched to see if any non-keywords were contained in the title. In this case, as a proof of concept, the non-keyword used was 'redox'. However, the code can be adapted according to the specifics of the research being conducted and the field of research. To allow for the function to exclude certain words from the title line, every title had to be split into separate words. Each word was then matched against the non-keyword. A count of how many titles contained the non-keyword was held. The program would then output the relevant number of articles from the web-scrape. Out of the 100 articles which were used as test data, the results showed that 3 articles contained a non-keyword and was not relevant to the search. This means that 97% of the results were relevant. The full code used to achieve this can be seen in **Appendix 3** and the results shown in **Appendix 5**

The next steps would be to use subroutines such as *NumPy* and *Pandas* to compare the results. This would require the data to be treated and presented in a format that can be recognised and analysed using code. For example, in an array. There would be many ways to explore building a standardised databases that can be analysed, such as machine learning, as part of the development in the field of AI. The full code showing how the result in this section were achieved can be seen in **Appendix 3**.

3.2 Results from Manually Created Database

As the research data was not prepared enough to analyse fully, the analysis was completed using the manual data base results. This was to prove that the concept would work, if the data

could be presented in the same format as the manually created database (**Table 3**). The code used to do the analysis is shown in **Appendix 4**.

The code works by splitting all the data in each column of the manual table (**Table 3**) in to separate entities per row. Any time the contents of the array variable is the same as 'Zn' the Zinc anode count would increase by 1. In the end, this would provide a count of how many times Zinc appeared in the anode data. This was then repeated for Lithium anodes. A final count of each material was printed to the user. The results showed that there were 17 Lithium Anodes and 44 Zinc anodes. A graph showing the difference in the number of times each material appeared was outputted as a visual reference to the user. This can be seen in **Figure 18**. The results showed that Zinc anodes appeared more than twice as often than the Lithium anodes. Potential reasons for this were because aqueous Zinc – Ion batteries have a large variety of advantages. Some of the advantages include being low cost, high safety and environmentally friendly, as aqueous Zinc based salts are soluble. Another advantage is that they are resource abundant and do not have costly materials.⁵¹ However, Lithium – Ion batteries have still come up often in the literature and this is because there are some disadvantages to Zinc – Ion batteries, the main one being, not being able to achieve high voltages. Other disadvantages include water evaporation and the hydrogen evolution reaction.

A graph to show how frequently each material appeared in the results.

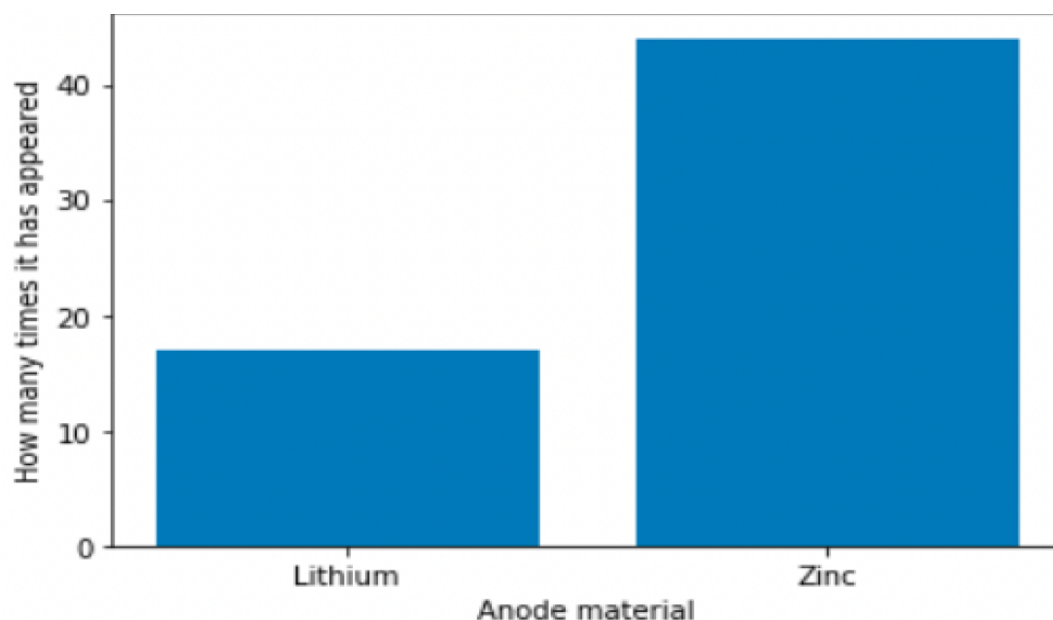


Figure 18 – Comparison of the number of times “Li” appeared in the title results obtained by the program, in comparison to “Zn”. This was used as a proof of concept to compare to the human obtained results.

Using the data from the manually created database. The word clouds in **Figure 19.a** and **Figure 19.b** were created, to show the most common materials that appeared in the database. For the cathode materials the most common was LiMn_2O_4 . For the anode materials this was zinc.

[illegible]

b)



Figure 19 – How frequently each material appeared, represented by the size and frequency of the word for (a) Cathode materials and (b) Anode materials, using a visual representation

Figure 19.b confirms that Zinc was a material that appeared often. As some materials appeared more than once in the database, they have appeared multiple times in the diagram. The most common cathode material was LiMn_2O_4 . When building the manual database, it was found that metal organic frameworks were also common. However, for our purposes these were not the materials we were researching.

The code used to analyse the database can also be adapted to do the same for the database created by the program. This would eventually allow the program to determine which materials are found most in literature and which areas have less research. Eventually guiding the direction of research taken by the user.

3.5 Results Conclusion

In this section we discussed the results achieved in this project. To conclude, we will summarise whether these results met the aims of the project as outlined in the introduction. In the results we were able to build a script which accessed previously published articles using web scraping. By accessing this literature, we were manually able to build a centralised database. For the purposes of this project, due to time and resource constraints, a smaller database was built as a proof of concept. This further allowed us to identify some of the promising aqueous battery chemistries. These chemistries are outlined in this section. However further research would be needed to find more specific materials and chemistries.

However, for some aims of the project, there was still room for improvement allowing for the potential of further scope. This included developing further code to automate the creation of a centralised database. Although this aim was worked on and progress was made, more time in the project would allow for further progress. Even though we were able to extract some useful information on aqueous batteries, creating a fully automated and centralised database would mean that we would be able to extract more useful information to create better aqueous batteries.

4. Conclusions and Summaries

During the conducted research many different combinations of anode, cathode and electrolyte chemistries were explored. There were a lot of available chemistries which were organised and sorted into a database.

One of the aims of the project was to identify the most promising aqueous battery chemistries. This aim was met by exploring published literature and identifying the properties of different material combinations. As outlined in **Figure 18**, Zinc was one of the most common anode materials in rocking chair aqueous batteries, which was readily available in published literature. The most common cathodes in “rocking chair” the Zn-batteries are based on manganese oxides with an ZnSO_4 electrolyte.

From **Figure 19.a** it can also be extracted that LiMn_2O_4 was a common cathode material. Even though these were the most common materials that appeared in literature. The energy density achieved is not suitable for the grid level and energy dense applications discussed at the beginning of this thesis.

Another aim of this thesis was to use extracted information to design aqueous batteries with higher energy density. However, given the time constraints of this project this would need further research and development in order to be completed. Even though the objective to build a script where published articles could be accessed was met, the second objective – to create a centralised database of values extracted from already published articles – would also

need to be met in order to accomplish the second aim. This would require more time and the possibilities for this development will be explored in the final section of this thesis titled 'Future Work'

5. Future Work and Discussion

This project could further be improved by using machine learning to analyse the results obtained in a standardised database. The standardised database would need to be created using a python script which can extract values from published articles and input them into the database under the correct columns for each property. This could be done by building further on to the python script created in this project. The analysis of the database would include combining different materials, which are in the database and by performing calculations to predict the properties and values for different combinations. This is complex project that would need to combine machine learning and artificial intelligence along with many more resources including time. However, this development would have major impacts not only in the applications of aqueous batteries but could also be adapted to be used in many fields of chemistry, wider science and even business and economical predictions.

5.1 Building a Standardised Database

Scientists rely on different sources of data to curate their experimental methods and hypothesis. A lot of this is based on openly accessible data and further so, databases which allow for theoretical calculations and modelling. These theoretical experimental methods rely on databases which contain values regarding different properties. An example of a standard databases which is commonly used is PubChem.⁵² PubChem is an open chemistry database

which provides data on many classifications, such as chemical structures, identifiers, chemical and physical properties, biological activities, patents, health, safety, toxicity data, and many others.⁵³ As of 2nd April 2024, the data counts for PubChem can be seen in **Figure 20**.⁵⁴

DOCS

PubChem Data Counts

Collection	Live Count	Description
Periodic Table of Elements	118	Interactive periodic table with up-to-date element property data collected from authoritative sources
Compounds	117,700,392	Unique chemical structures extracted from contributed PubChem Substance records
Substances	316,000,490	Information about chemical entities provided by PubChem contributors
BioAssays	1,627,767	Biological experiments provided by PubChem contributors
Bioactivities	294,156,375	Biological activity data points reported in PubChem BioAssays
Genes	113,157	Gene targets tested in PubChem BioAssays and those involved in PubChem Pathways
Proteins	241,899	Protein targets tested in PubChem BioAssays and those involved in PubChem Pathways
Taxonomy	107,823	Organisms of targets tested in PubChem BioAssays and those involved in PubChem Pathways
Pathways	240,992	Interactions between chemicals, genes, and proteins
Cell Lines	2,005	Information about cell lines
Literature	40,513,339	Scientific publications with links in PubChem
Patents	50,836,952	Patents with links in PubChem
Data Classifications	71	Browse the distribution of PubChem data among nodes in the hierarchy of interest
Data Sources	984	Organizations contributing data to PubChem

Figure 20 – Live data counts for the PubChem databases on 2nd April 2024.

Data is collected from various sources including journal publishers. Data is open and submissions can be made by providing organizational information and meeting pre-outlined criteria depending on the findings being published. This ensures the standardization of the database to avoid having missing fields or duplicated values in the data.

The advantages of creating a similar database to include the electrochemical performance of different aqueous battery materials, would be that all data could be accessed from the same

place allowing for further discoveries to be made. It would also mean that data can be accessible to all scientists around the world. This will lead to scientists being exposed to more research and therefore not repeating studies which have already been done. Instead, allowing scientists to spend more time on new discoveries.

As useful as this data is in collating thousands of records in one place, it does arise questions such as how standardised can the data be? Experimental results will always vary depending on the lab conditions, experimental parameters and the instruments used. Furthermore, experimental data does not always coincide with theoretical data due to many factors including human error. Will this require the need for experiments to also become standardised?

These problems would also mirror in the standard database that we are building. Data must be cleared. The process of doing this would include only using data from sources which contained values for all the parameters being observed. This was not the case in the manual database built, as not all articles reported on the same properties. However, as not a lot of articles reported the data in the format being searched for, many articles could be disregarded. This could eventually lead to inconsistencies in the program when reporting data back to the user or when calculations and analysis is performed on the data.

Other issues to look out for include:

- Data entries must be monitored. This process can take time, especially if a peer review method is used – however, automating this process using artificial intelligence could reduce the time. This is a crucial step to maintain the quality of the database.

- Depending on the research area, different parameters will be measured, and different units used. Data may need to be converted to the same units however in some cases this may not be possible. It may also mean the accuracy of the data is compromised.
- As the database is anticipated to hold a lot of data, the most effective storage options will need to be explored. This may include cloud data storage.
- Standardising experiments will mean fewer discrepancies in results. However, observing and analysing discrepancies can lead to findings which were not originally anticipated yet are useful in the study.

5.2 Applications of AI

The workflow discussed in this thesis focuses on deterministic web scraping and keyword filtering techniques. This approach prioritises transparency, reproducibility and computational efficiency as the inclusion criteria can be directly determined by the user. In this section we will discuss the possible integrations of AI to this approach and the potential advances and limitations of AI integration.

Recent advances in AI and LLMs allows for new advances in scientific research models.

Unlike traditional keyword-based methods, LLMs can process contextual information and identify semantically related concepts even when exact search terms are absent.

Applications include semantic search, automated summarisation, topic extraction, question answering, and knowledge graph construction from scientific documents. However the functionality of this feature relies on the ability of AI to handle domain specific knowledge and extensive context windows, which are not always readily available online for AI to learn

from.⁵⁵ Though some of these features can allow for improvements in efficiency, they can lead to further limitations in the project.

Limitations:

- One important criterion is ensuring that AI systems do not rely on non-academic sources, outdated studies, or obsolete findings. Effective source validation is essential to ensure that only peer-reviewed, credible, and relevant literature is included in source articles. Source validation ensures the inclusion of only peer-reviewed, reliable, and relevant resources in the literature review. AI technologies may sometimes source articles from predatory journals or inferior databases; hence, it is incumbent upon the researcher to evaluate the credibility of each source or apply the relevant criteria to ensure the algorithm considers this.⁵⁶ It is also important to ensure that AI is able to access the most recent findings available on the internet.⁵⁷
- Another important consideration is the reproducibility of AI-assisted literature reviews. Traditional database searches can be replicated by documenting search terms, databases, inclusion criteria, and dates of retrieval. In contrast, AI-generated outputs may vary across sessions due to model updates, changes in training data, algorithm modifications, or the inherent non-deterministic nature of large language models.⁵⁸ Future studies should delve into the reasons behind these discrepancies and their implications for reliability and usability in academic contexts. Which also leads on to the ethicality of using AI and the biases it can lead to in results.⁵⁹ As some materials may be researched more extensively, they may not always be the most suitable materials to study in the user's case. It is important to verify these

biases in results by human oversight, however as many AI systems work as ‘black boxes’ it is difficult to verify how sources are selected or ranked.

- When used for scientific literature review and analysis, AI systems can introduce inaccuracies due to errors in data interpretation, limitations in reasoning, and inherent shortcomings of current machine learning models. Despite extensive training, AI tools may still generate unpredictable mistakes and misleading outputs.

⁵⁹ These challenges are particularly as they can produce factual inaccuracies, misinterpret research findings, and generate incorrect or fabricated citations. A notable concern is the generation of hallucinated references —citations that appear credible and properly formatted but do not correspond to actual published sources. Consequently, all AI-generated references, summaries, and analyses should be independently verified against original publications. Human oversight remains essential to ensure the reliability, validity, and ethical use of AI in scientific literature review and research synthesis. ⁵⁷

Advances:

- An advantageous way of incorporating AI into the project is to use AI to help classify articles based on what materials they relate to. For example, instead of manually sorting whether each article relates to the anode, cathode, electrolyte or whole system, AI could be given specific instructions on how to sort and classify each article in preparation for the building the database. In order to do this, we would need to find a suitable AI tool which can follow very specific criteria. Especially when trying classify articles based on the material or materials they are discussing within these

categories. As AI systems become more advanced, their ability and accuracy to do this will also increase.⁶⁰

- Another benefit of applying AI as assistance in the project would be to use AI in assisting with building databases once the research sources have been determined. This can be useful in future work in the application of building a standardised database.⁶¹ In order to do this machine learning models such as TensorFlow and PyTorch libraries, which were previously mentioned, can be used to populate the database as well as identify and prevent errors in the database.⁶² This method would create efficiency in storing, processing and managing data, creating more efficiency in the project and reducing the time taken to populate a standardised database. A proof of concept for this work is the work done by **GeoDeepShovel**, where they explore the use of AI for data extraction, data aggregation and scientific database construction.⁶¹

However, in all the instances above AI would need to be manually checked by human oversight to ensure the reliability of the produced script before applying the use of AI at a greater level.

The use of Google Scholar Labs:

Google Scholar Labs⁶³ is an experimental AI-powered feature from Google that extends the capabilities of Google Scholar. Instead of relying only on keyword searches, it allows researchers to ask complex research questions in natural language and receive AI-assisted literature recommendations and explanations. The main difference is that **Google Scholar** is

a traditional academic search engine, while **Google Scholar Labs** is an AI-assisted research discovery tool built on top of Google Scholar. The key difference is that papers are selected, and an AI assisted summarization is provided along with the relevant papers related to the question.

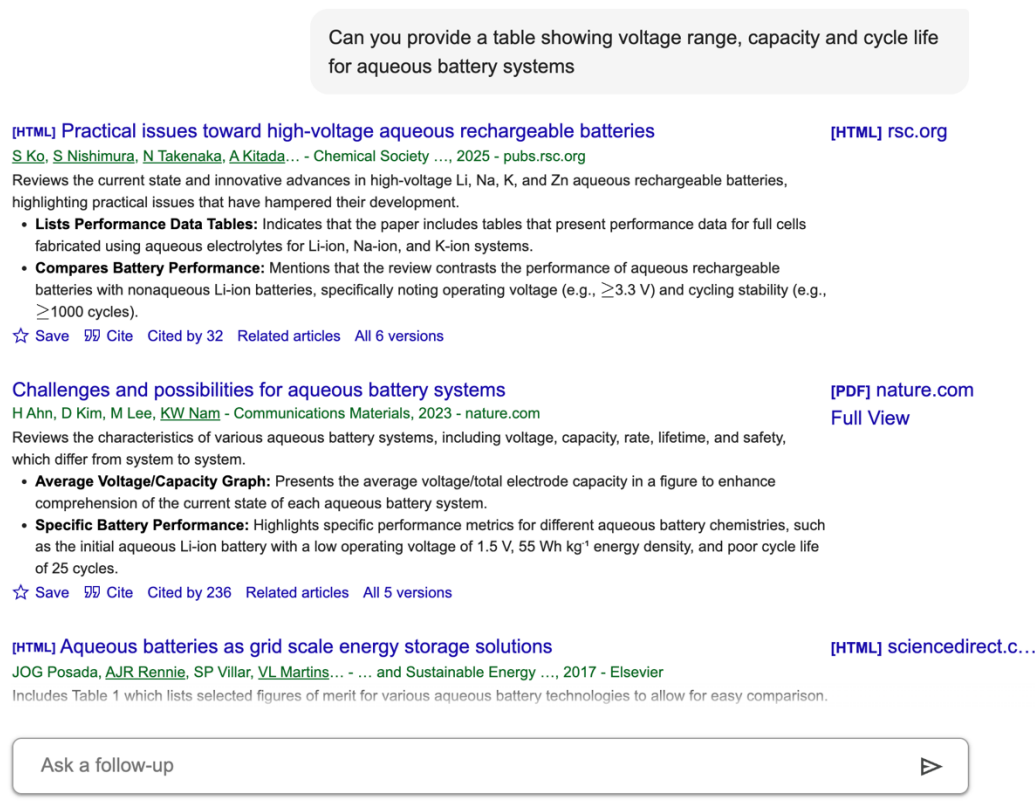


Figure 21 – A screenshot of a search query in Google Scholar Labs when searching for articles related to the research topic.

The query entered in **Figure 21** – A screenshot of a search query in Google Scholar Labs when searching for articles related to the research topic was “Can you provide a table showing voltage range, capacity and cycle life for aqueous battery systems”. As seen Google Scholar Labs did not produce a table as a result of the query. However, it did provide articles that contained tables. Therefore, this tool could be useful in determining the source articles to be used.

However, a factor to consider is that some of these articles focused on only a few potential material combinations as opposed to comparing the voltage range, capacity and cycle life for various combinations. Therefore, using a traditional google scholar search which looks for the word 'review' was more effective in finding articles which are reviewing multiple material combinations instead of in-depth research on one combination. Similarly, to a standard search without using Google Scholar Labs, in order for the specific values to be populated on to the database, these articles would need to be human reviewed and selected as source articles. A potential for future work would be to define source articles using Google Scholar Labs and as mentioned above explore the applications of AI in populating the database by using AI to review the articles and extract the relevant information.

The methodology presented in this thesis is primarily rule-based, relying on web scraping and keyword filtering to identify relevant publications. This contrasts with recent LLM-based approaches that perform semantic search and contextual analysis of scientific literature. While LLMs can identify conceptually related content beyond explicit keyword matches, they typically require greater computational resources and may present challenges related to interpretability and reproducibility. The workflow developed here emphasises transparency and ease of validation, but future work could investigate the integration of LLM-based semantic retrieval and automated summarisation to complement the database generation process.

5.3 Improving the Scope of the Project

Some of the problems faced during this project have already been outlined during the method. An issue that caused significant delays in the project was when the code would not

web-scrape Google Scholar after a certain number of test runs. This is when proxies were introduced to resolve this problem. This required further research to find the most suitable solution for our aims whilst still considering licensing and accessibility issues.

However, one of the major problems that would need to be tackled to progress the project would be accessing the full abstracts for each article rather than a preview. This can then be developed to read data from the full article, which will then help to build a standardised database. This will require advanced web scraping and data-mining techniques. This will allow standardised databases to be created for different topics, fields and even subjects.

References

- 1Z. Liu, Y. Huang, Y. Huang, Q. Yang, X. Li, Z. Huang and C. Zhi, Voltage issue of aqueous rechargeable metal-ion batteries, *Chem. Soc. Rev.*, 2020, **49**, 180–232.
- 2G. E. Blomgren, The Development and Future of Lithium Ion Batteries, *J. Electrochem. Soc.*, 2017, **164**, A5019–A5025.
- 3B. Scrosati, J. Hassoun and Y.-K. Sun, Lithium-ion batteries. A look into the future, *Energy & Environmental Science*, 2011, **4**, 3287–3295.
- 4D. Chao, W. Zhou, F. Xie, C. Ye, H. Li, M. Jaroniec and S.-Z. Qiao, Roadmap for advanced aqueous batteries: From design of materials to applications, *Sci. Adv.*, 2020, **6**, eaba4098.
- 5Y. Sui and X. Ji, Anticatalytic Strategies to Suppress Water Electrolysis in Aqueous Batteries, *Chem. Rev.*, 2021, **121**, 6654–6695.
- 6H. Ahn, D. Kim, M. Lee and K. W. Nam, Challenges and possibilities for aqueous battery systems, *Commun Mater*, 2023, **4**, 1–19.
- 7H. Zhang, X. Liu, H. Li, I. Hasa and S. Passerini, Challenges and Strategies for High-Energy Aqueous Electrolyte Rechargeable Batteries, *Angewandte Chemie International Edition*, 2021, **60**, 598–616.
- 8Y. Shen, B. Liu, X. Liu, J. Liu, J. Ding, C. Zhong and W. Hu, Water-in-salt electrolyte for safe and high-energy aqueous battery, *Energy Storage Materials*, 2021, **34**, 461–474.
- 9J. Liu, C. Xu, Z. Chen, S. Ni and Z. X. Shen, Progress in aqueous rechargeable batteries, *Green Energy & Environment*, 2018, **3**, 20–41.
- 10 W. Li, J. R. Dahn and D. S. Wainwright, Rechargeable Lithium Batteries with Aqueous Electrolytes, *Science*, 1994, **264**, 1115–1118.
- 11 D. Selvakumaran, A. Pan, S. Liang and G. Cao, A review on recent developments and challenges of cathode materials for rechargeable aqueous Zn-ion batteries, *J. Mater. Chem. A*, 2019, **7**, 18209–18236.
- 12 A. Nagpal and G. Gabrani, in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 140–145.
- 13 A. Bogdanchikov, M. Zhaparov and R. Suliyev, Python to learn programming, *J. Phys.: Conf. Ser.*, 2013, **423**, 012027.
- 14 T. E. Oliphant, Python for Scientific Computing, *Computing in Science & Engineering*, 2007, **9**, 10–20.
- 15 Anaconda | Anaconda Commercial Edition FAQ, <https://www.anaconda.com/blog/anaconda-commercial-edition-faq>, (accessed 30 July 2022).
- 16 Document search - Web of Science Core Collection, <https://www.webofscience.com/wos/woscc/basic-search>, (accessed 13 May 2022).
- 17 Google Scholar, <https://scholar.google.com/>, (accessed 13 May 2022).
- 18 Scrape Google Scholar with Python, <https://dev.to/dmitryzub/scrape-google-scholar-with-python-32oh>, (accessed 30 July 2022).
- 19 A. Pathak, 10 Best Rotating Proxy for Web Scraping, SEO, and More..., <https://geekflare.com/best-rotating-proxy/>, (accessed 30 July 2022).
- 20 What is a Proxy Server?, <https://www.fortinet.com/resources/cyberglossary/proxy-server>, (accessed 30 July 2022).

- 21 M. Ganesan, Scraping Google Scholar with Python and BeautifulSoup, <https://proxiesapi-com.medium.com/scraping-google-scholar-with-python-and-beautifulsoup-850cbdfedbcf>, (accessed 30 July 2022).
- 22 The Royal Society of Chemistry, <https://www.rsc.org/>, (accessed 19 November 2021).
- 23 RSC Journals Home, <https://pubs.rsc.org/en/journals?key=title&value=current>, (accessed 27 August 2024).
- 24 I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.-F. Chang and L. Hua, Data Mining in Healthcare and Biomedicine: A Survey of the Literature, *J Med Syst*, 2012, **36**, 2431–2448.
- 25 C. C. Editor, Challenge-Response Protocol - Glossary | CSRC, https://csrc.nist.gov/glossary/term/challenge_response_protocol, (accessed 15 July 2022).
- 26 M. Sharma, Data Mining: A Literature Survey, 2014, **3**, 5.
- 27 L. Chen, J. Zhang, Q. Li, J. Vatamanu, X. Ji, T. P. Pollard, C. Cui, S. Hou, J. Chen, C. Yang, L. Ma, M. S. Ding, M. Garaga, S. Greenbaum, H.-S. Lee, O. Borodin, K. Xu and C. Wang, A 63 m Superconcentrated Aqueous Electrolyte for High-Energy Li-Ion Batteries, *ACS Energy Lett.*, 2020, **5**, 968–974.
- 28 N. Alias and A. A. Mohamad, Advances of aqueous rechargeable lithium-ion battery: A review, *Journal of Power Sources*, 2015, **274**, 237–251.
- 29 G. L. Team, Top 30 Python Libraries To Know in 2023, <https://www.mygreatlearning.com/blog/open-source-python-libraries/>, (accessed 21 March 2023).
- 30 *Data Scraping & Data Mining for Beginners to Pro with Python - Udemy Online Course*, .
- 31 G. L. Hajba, in *Website Scraping with Python: Using BeautifulSoup and Scrapy*, ed. G. L. Hajba, Apress, Berkeley, CA, 2018, pp. 41–96.
- 32 G. L. Hajba, in *Website Scraping with Python: Using BeautifulSoup and Scrapy*, ed. G. L. Hajba, Apress, Berkeley, CA, 2018, pp. 97–167.
- 33 R. V. Chandra and B. S. Varanasi, Python Requests Essentials.
- 34 A. Ginsburg, B. M. Sipőcz, C. E. Brasseur, P. S. Cowperthwaite, M. W. Craig, C. Deil, J. Guillochon, G. Guzman, S. Liedtke, P. L. Lim, K. E. Lockhart, M. Mommert, B. M. Morris, H. Norman, M. Parikh, M. V. Persson, T. P. Robitaille, J.-C. Segovia, L. P. Singer, E. J. Tollerud, M. de Val-Borro, I. Valtchanov, J. Woillez, and The Astroquery collaboration, a subset of the astropy collaboration, astroquery : An Astronomical Web-querying Package in Python, *AJ*, 2019, **157**, 98.
- 35 G. L. Team, Top 50+ Selenium Interview Questions and Answers in 2023, <https://www.mygreatlearning.com/blog/selenium-interview-questions/>, (accessed 29 March 2023).
- 36 C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, Array programming with NumPy, *Nature*, 2020, **585**, 357–362.
- 37 Nishali, What is Pandas in Python?, <https://www.scaler.com/topics/pandas/what-is-pandas-in-python/>, (accessed 23 March 2023).

- 38 What Is Pandas in Python?, <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>, (accessed 23 March 2023).
- 39 pandas - Python Data Analysis Library, <https://pandas.pydata.org/about/index.html>, (accessed 23 March 2023).
- 40 SciPy, <https://scipy.org/>, (accessed 23 March 2023).
- 41 Matplotlib — Visualization with Python, <https://matplotlib.org/>, (accessed 26 March 2023).
- 42 D. Johnson, What is TensorFlow?, <https://www.guru99.com/what-is-tensorflow.html>, (accessed 27 March 2023).
- 43 Why TensorFlow, <https://www.tensorflow.org/about>, (accessed 27 March 2023).
- 44 Case Studies and Mentions | TensorFlow, <https://www.tensorflow.org/about/case-studies>, (accessed 27 March 2023).
- 45 From Patterns in Data to Knowledge Discovery, <https://reader.elsevier.com/reader/sd/pii/S187538921500036X?token=6C5C3EAAED554C6F0E7772BD6ED0754F243EB9DA6F347F77C7DD243E246CD5E4BAC413CE727FCC41C10AD2D11168AAED&originRegion=eu-west-1&originCreation=20230515135738>, (accessed 15 May 2023).
- 46 A. Algarni, Data Mining in Education, *ijacsa*, DOI:10.14569/IJACSA.2016.070659.
- 47 I. CORPORATIVA, Discover how data mining will predict our behaviour, <https://www.iberdrola.com/innovation/data-mining-definition-examples-and-applications>, (accessed 18 May 2023).
- 48 Data Mining Specialization - Coursera, <https://www.coursera.org/specializations/data-mining>.
- 49 University of Illinois at Urbana-Champaign, <https://www.coursera.org/illinois>.
- 50 Use Chrome Developer Tools to check tags, <https://support.google.com/campaignmanager/answer/2828688?hl=en#:~:text=Right%2Dclick%20a%20page%20and,for%20the%20element%20you%20clicked.&zipy=%2Cuse-cases>.
- 51 Y. Lv, Recent Advances in Electrolytes for “Beyond Aqueous” Zinc-Ion Batteries.
- 52 PubChem, <https://pubchem.ncbi.nlm.nih.gov>.
- 53 About PubChem, <https://pubchem.ncbi.nlm.nih.gov/docs/about>.
- 54 PubChem Data Counts, <https://pubchem.ncbi.nlm.nih.gov/docs/statistics>.
- 55 M.-K. Ghali, A. Farrag, D. Won and Y. Jin, Enhancing knowledge retrieval with in-context learning and semantic search through generative AI, *Knowledge-Based Systems*, 2025, **311**, 113047.
- 56 Edmond Osei Arhin, Festus Cobena Ainoo, and Kwame Amponsah, Integration of Artificial Intelligence (AI) Into Academic Literature Reviews: An overview, *World J. Adv. Sci. Technol.*, 2025, **7**, 006–023.
- 57 M. A. Kacena, L. I. Plotkin and J. C. Fehrenbacher, The Use of Artificial Intelligence in Writing Scientific Review Articles, *Curr Osteoporos Rep*, 2024, **22**, 115–121.
- 58 OECD, *Artificial Intelligence in Science Challenges, Opportunities and the Future of Research: Challenges, Opportunities and the Future of Research*, OECD Publishing, 2023.
- 59 D. B. Resnik and M. Hosseini, The ethics of using artificial intelligence in scientific research: new guidance needed for a new tool, *AI Ethics*, 2025, **5**, 1499–1521.

- 60 E. Sachini, K. Sioumalas-Christodoulou, S. Christopoulos and N. Karampekios, AI for AI: Using AI methods for classifying AI science documents, *Quantitative Science Studies*, 2022, **3**, 1119–1132.
- 61 S. Zhang, H. Xu, Y. Jia, Y. Wen, D. Wang, L. Fu, X. Wang and C. Zhou, GEODEEPSHOVEL : A platform for building scientific database from geoscience literature with AI assistance, *Geoscience Data Journal*, 2023, **10**, 519–537.
- 62 N. D. Pahlavonovich, THE IMPORTANCE OF ARTIFICIAL INTELLIGENCE SYSTEMS IN CREATING DATABASES, *FRANCE - SCIENTIFIC REVIEW OF THE PROBLEMS AND PROSPECTS OF MODERN SCIENCE AND EDUCATION*, 2025, **1**, 4–8.
- 63 Google Scholar, https://scholar.google.com/scholar_labs/search?hl=en, (accessed 9 June 2026).

Appendix

Cathode, electrode and electrolyte combinations

	Anode	Cathode	Electrolyte	Potential range (V)	Capacity Retention (%) (per cycles)	Current density (mA g ⁻¹)	Energy Density (W h kg ⁻¹)	Capacity (mA h g ⁻¹)	Reference
	Zn	MnO ₂	ZnSO ₄	1.0–1.9	187	100			3
	Zn	a-MnO ₂ (alpha)	Zn(SO ₄) ₂ + MnSO ₄	0.8–1.9	405	150			3
	Zn	a-MnO ₂	ZnSO ₄	1.0–1.8	233	83			3
	Zn	a-MnO ₂ @ C	ZnSO ₄	1.0–1.8	272	66			3
	Zn	a-MnO ₂ /CNT	Zn(SO ₄) ₂ + MnSO ₄	1.0–1.9	665	100			3
	Zn	g-MnO ₂ (gamma)	ZnSO ₄	1.0–1.8	285	50			3
	Zn	b-MnO ₂ (beta)	ZnSO ₄	1.0–1.8	270	100			3
	Zn	d-MnO ₂ (delta)	ZnSO ₄	1.0–1.8	122	83			3
	Zn	l-MnO ₂ (lambda)	Zn(SO ₄) ₂ + MnSO ₄	0.8–1.9	122	100			3
	Zn	Todorokite MnO ₂	ZnSO ₄	0.7–2.0	98	50			3
	Zn	Mn ₂ O ₃	Zn(SO ₄) ₂ + MnSO ₄	1.0–1.9	78	100			3
	Zn	Mn ₃ O ₄	ZnSO ₄	0.8–1.9	232	200			3
	Zn	NaMnO ₂	Zn(CH ₃ COO) ₂ + CH ₃ COONa	1.0–2.0	60	120			3
	Zn	ZnHCF	ZnSO ₄	-0.2 to 1.2	73.7	300			3
	Zn	ZnHCF@MnO ₂	ZnSO ₄	1.4–1.9	118	100			3
	Zn	CuHCF	ZnSO ₄	0.0–1.1	56	20			3
	Zn	NiHCF	Zn(ClO ₄) ₂ in acetonitrile	0.5–1.8	55.6	60			3
	Zn	Zn _{0.25} V ₂ O ₅ .nH ₂ O	ZnSO ₄	0.5–1.4	282	300			3
	Zn	Ca _{0.25} V ₂ O ₅ .nH ₂ O	ZnSO ₄	0.6–1.6	340	60			3
	Zn	CaV ₆ O ₁₆ .3H ₂ O	Zn(CF ₃ SO ₃) ₂	0.2–1.6	367	50			3
	Zn	Mg _{0.34} V ₂ O ₅ .nH ₂ O	Unknown		353	100			3
	Zn	Li _x V ₂ O ₅ .nH ₂ O	ZnSO ₄	0.4–1.4	407.6	1000			3
	Zn	H ₂ V ₃ O ₈	Zn(CF ₃ SO ₃) ₂	0.2–1.6	423.8	100			3
	Zn	H ₂ V ₃ O ₈ /graphene	Zn(CF ₃ SO ₃) ₂	0.2–1.6	394	100			3
	Zn	V ₂ O ₅ .nH ₂ O/graphene	Zn(CF ₃ SO ₃) ₂	0.2–1.6	381	60			3
	Zn	V ₂ O ₅	ZnSO ₄	0.4–1.4	224	100			3
	Zn	V ₂ O ₅	LiTFSI/Zn(CF ₃ SO ₃) ₂	0.2–1.6	238	50			3
	Zn	V ₆ O ₁₃ .nH ₂ O	Zn(CF ₃ SO ₃) ₂	0.2–1.4	395	100			3
	Zn	NH ₄ V ₄ O ₁₀	ZnSO ₄	0.4–1.4	380.3	100			3
	Zn	Zn ₂ V ₂ O ₇	ZnSO ₄	0.4–1.4	248	50			3
	Zn	Zn ₃ V ₂ O ₇ (OH) ₂ .2H ₂ O	ZnSO ₄	0.2–1.8	213	50			3
	Zn	NaV ₃ O ₈ .1.5H ₂ O	ZnSO ₄ /Na ₂ SO ₄	0.3–1.3	380	50			3
	Zn	VO ₂	Zn(CF ₃ SO ₃) ₂	0.7–1.7	265	100			3
	Zn	VO ₂ (B)	Zn(CF ₃ SO ₃) ₂	0.3–1.5	357	100			3
	Zn	rGO/VO ₂	Zn(CF ₃ SO ₃) ₂	0.3–1.3	276	100			3
	Zn	VS ₂	ZnSO ₄	0.4–1.4	190.3	50			3
	Zn	Na ₃ V ₂ (PO ₄) ₃	Zn(CH ₃ COO) ₂	0.8–1.7	97	50			3
	Zn	Na ₃ V(PO ₄) ₂ F ₃ @C	Zn(CF ₃ SO ₃) ₂	0.8–1.9	75	80			3
	Zn	Mo ₆ S ₈	ZnSO ₄	0.25–1.0	79	45			3
	Zn	ZnMo ₆ S ₈	ZnSO ₄	0.25–1.0	57	128			3
	Zn	Calix[4]quinone	Zn(CF ₃ SO ₃) ₂	0.2–1.8	335	20			3
	Zn	1,4-Bis(diphenylamino)benzene	LiTFSI/Zn(CF ₃ SO ₃) ₂	0.6–2.0	125	26			3
	Zn	P-NiCo ₂ O ₄ -x	KOH + Zn(CH ₃ COO) ₂	1.4–2.0	309.2	6000			3
	Zn	MoS ₂ -x nanosheets	Zinc trifluoromethanesulfonate	0.2–1.2	128.23	100			3
	LiTi ₂ (PO ₄) ₃	LiFePO ₄	Li ₂ SO ₄ (1 M)		90% (1000)	6.0 C		55	7
	VO ₂	LiFePO ₄ /C	LiNO ₃ (Saturated)		94% (50)	C/3		106	7
	LiV ₃ O ₈	LiFePO ₄ /C	LiNO ₃ (9 M)		99% (100)	10.0 C		90	7
	Activated carbon	LiFePO ₄	Li ₂ SO ₄ (0.5 M)		63% (10)	5.0 C		124	7
	Activated carbon	LiFePO ₄	Li ₂ SO ₄ (0.5 M)		45%	20.0 C		58	7
	VO ₂	LiMn ₂ O ₄	LiNO ₃ (Saturated)		83% (42)	0.2 C		120	7
	Activated carbon	LiMn ₂ O ₄	Li ₂ SO ₄ (0.5 M)		93% (10,000)	90.0 C		118	7
	TiP ₂ O ₇	LiMn ₂ O ₄	LiNO ₃ (5 M)		85% (10)	0.2 C		42	7
	Activated carbon	LiMn ₂ O ₄	Li ₂ SO ₄ (0.5 M)		99% (200)	4.5 C		110	7
	Activated carbon	LiMn ₂ O ₄	Li ₂ SO ₄ (0.5 M)		94% (1200)	4.5 C		110	7
	Li _x V ₂ O ₅ -Ppy	LiMn ₂ O ₄	LiNO ₃ (5 M)		82% (60)	0.2 C		40	7
	Ppy	LiCoO ₂	Li ₂ SO ₄ (Saturated)		63% (120)	0.1 C		104	7
	LiTi ₂ (PO ₄) ₃	LiMn _{0.05} Ni _{0.05} Fe _{0.9} PO ₄	Li ₂ SO ₄ (Saturated)		53% (50)	0.2 C		87	7
				Average operating voltage					
	Mo ₆ S ₈	LiMn ₂ O ₄	LiTFSI (21 m)	1.4	78 % after 100 cycles	19		141	9
	Mo ₆ S ₈	LiFePO ₄	LiTFSI (21 m)	0.9	60 % after 1000 cycles	130		100	9
	C/TiO ₂	LiMn ₂ O ₄	LiTFSI (21 m) + LiOTf (7 m)	1.65	78 % after 1000 cycles	75		144	9
	Li ₄ Ti ₅ O ₁₂	Li _{0.5} Mn _{1.5} O ₄	Li(TFSI) _{0.7} (BETI) _{0.3} · 2H ₂ O	3.1	63 % after 100 cycles	1000		120	9
	Li ₄ Ti ₅ O ₁₂	LiCoO ₂	Unknown	2.4	75 % after 200 cycles	1370		120	9
	Li ₄ Ti ₅ O ₁₂	LiCoO ₂	Li(PTFSI) _{0.6} (TFSI) _{0.4} · 1.15H ₂ O	2.4	77 % after 100 cycles	36		100	9
	Hollow MoS ₃ nano	LiMn ₂ O ₄	LiTFSI (21 m)	1.35	70 % after 1000 cycles	100		127	9
					94 % after 1000 cycles	1000		65.1	9
	Mo ₆ S ₈	LiMn ₂ O ₄	LiTFSI (21 m)	2.3			84		
	Mo ₆ S ₈	LiFePO ₄	LiTFSI (21 m)	1.3			47		14
	Mo ₆ S ₈	Li _{0.5} Mn _{1.5} O ₄	LiTFSI (21 m)	2.9			80		14
			LiTFSI (21 m) + 1M HTFSI(0.1%)	3			126		14

```

proxies = {"http": "http://10.10.1.10:3128", "http": "http://10.10.1.10:1080", "http": "http://95.66.151.101:8080"}
headers = {'User-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.3 Safari/605.1.15'}
url= 'https://scholar.google.com/scholar?as_q=aqueous+battery+review&as_epq=&as_eq=non-aqueous&as_occt=any&as_sauthors=&as_publication=&as_ylo'
response = requests.get(url,headers=headers, proxies=proxies)
soup=BeautifulSoup(response.content, 'lxml')
for x in range (10, 100, 10): #loops through url in order to search through all page numbers and repeats the above code in a for loop to get the same
    url = 'https://scholar.google.com/scholar?start='+ str(x) + '&q=aqueous+battery&hl=en&as_sdt=0,5' #loops through url
    response = requests.get(url,headers=headers)
    soup=BeautifulSoup(response.content, 'lxml')

```

```

71
72 nonkeywords=[] #creates a new variable so that this list can contain the non relevant article
73
74 i=0
75 while i < len(title_list): #this loop takes out an article that has redox in the title
76     title_split = title_list[i].split(" ")
77     #print(title_split)
78
79     if 'redox' in title_split:
80         #print ("yes")
81         nonkeywords.append(title_list[i])
82         title_list.remove(title_list[i])
83     i= i+1
84
85 j=0
86 while j < len(title_list): #this loop does the same as previous but instead excludes flow
87     title_split = title_list[j].split(" ")
88     # print(title_split)
89
90     if 'flow' in title_split:
91         #print ("yes")
92         nonkeywords.append(title_list[j])
93         title_list.remove(title_list[j])
94     j= j+1
95
96 k=0
97 while k < len(title_list): #this loop does the same as previous but instead excludes air
98     title_split = title_list[k].split(" ")
99     #print(title_split)
100
101     if 'air' in title_split:
102         #print ("yes")
103         nonkeywords.append(title_list[k])
104         title_list.remove(title_list[k])
105     k= k+1
106
107
108 #print (title_list)
109
110 print(str(len(nonkeywords)) + " articles are not relevant to your research") #says the amount
111
112 print(str((len(title_list))) + " articles are relevant to your research")
113
114 #print(len(title_list))
115 #print(len(abstract_list)-(len(nonkeywords)))
116 #print(len(url_list)-(len(nonkeywords)))
117
118

```

Appendix 3

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon Dec 6 11:02:51 2021
5
6  @author: ia
7  """
8
9  import os
10 import csv
11 import pandas as pd
12 import numpy as np
13 import matplotlib.pyplot as plt
14
15 f = open('/Users/ia/Documents/Documents - Irem's MacBook Pro/Masters/Resources/manuald2.csv',
16 all_data = f.read()
17 print(all_data)
18 ls = all_data.split('\n')
19 anode = []
20 cathode = []
21 anode
22 for l in ls:
23     bits = l.split(',')
24     anode.append(str(bits[0]))
25     cathode.append(str(bits[1]))
26 zn= anode.count("Zn")
27 print ("There are", zn, "zinc anodes")
28 c=len(anode)
29 li = c - zn
30 print ("There are", li, "lithium anodes")
31 anode = ['Lithium', 'Zinc']
32 counts = [li,zn]
33
34 plt.bar(anode,counts)
35 plt.title('The amount of times different anode materials appear in the data ')
36 plt.xlabel('Anode material')
37 plt.ylabel('How many times it has appeared')
38 plt.show()
39

```

```
document
97 Toward an aqueous solar battery: Direct electr... ... https://epub.ub.uni-muenchen.de/67501/1/
CNx_So...
98 Progress in aqueous rechargeable sodium-ion ba... ... https://onlinelibrary.wiley.com/doi/abs/
10.100...
99 An aqueous rechargeable chloride ion battery ... https://www.sciencedirect.com/science/
article/...

[100 rows x 3 columns]
3 articles are not relevant to your research
97 articles are relevant to your research
```

Appendix 5