Advanced Literature Search

Fionn McGoldrick

I. Introduction

This document is a sample solution for Tutorial Exercise 5. Its brief is available as a PDF on our Moodle page.

Conducting effective literature searches is a fundamental research skill that requires mastery of advanced query techniques. This exercise demonstrates how to use sophisticated search queries in three major academic databases: IEEE Xplore, ACM Digital Library, and Scopus. These techniques include Boolean operators, proximity searches, wildcard characters, field-specific filters, and strategic exclusions to refine results. The exercise also illustrates strategies for finding both secondary (reviews and surveys) and primary sources, depending on your research objectives.

Each section presents a specific research topic and shows how to progressively refine search queries through iterative improvements, demonstrating how researchers narrow from hundreds or thousands of results to a manageable set of highly relevant papers. The exercises use real query syntax that you can copy and execute directly in each database.

II. IEEE Xplore

This research focuses on security architectures and frameworks for Internet of Things deployments in smart city applications, specifically targeting survey and review papers while excluding healthcare and honeypot-related cybersecurity research. The topic "IoT Security Architectures for Smart Cities" narrows the broad "Internet of Things" domain to security concerns in urban infrastructure contexts.

A. Initial Broad Search

This initial query searches for papers with either "Internet of Things" or its abbreviation "IoT" in the document title, combined with "smart cit*" (where the wildcard * captures "city" and "cities"):

```
(("Document Title":"Internet of Things")
OR ("Document Title":"IoT"))
AND ("Document Title":"smart cit*")
```

The use of field-specific searches (e.g. `"Document Title":`) ensures that these core concepts appear in the title rather than just anywhere in the paper, which significantly improves relevance. However, this search returns approximately 974 results covering all aspects of IoT in smart cities, including applications, protocols, and infrastructure, making it too broad for a focused security review.

B. Adding Review Focus

The second query adds a critical refinement by restricting results to survey and review papers:

```
(("Document Title":"Internet of Things")
OR ("Document Title":"IoT"))
AND ("Document Title":"smart cit*")
AND (("Document Title":"Literature Review")
OR ("Document Title":"survey")
OR ("Document Title":"review"))
```

By requiring "Literature Review", "survey", or "review" to appear in the document title, we filter for secondary sources that synthesise existing research rather than primary research papers reporting original studies. This is particularly valuable for literature review work, as these papers provide comprehensive overviews of a research area and often identify key themes, gaps, and future directions. This single refinement dramatically reduces the results from 974 to approximately 44 papers, demonstrating the power of field-specific filters. The results now focus on papers that specifically aim to review the IoT smart city landscape, but still cover all aspects (applications, protocols, security, etc.) without a specific focus area.

C. Final Refined Search with Security Focus and Exclusions

The final query adds two sophisticated refinement techniques that narrow the focus to security architectures while excluding unwanted subdomains:

```
(("Document Title":"Internet of Things")
OR ("Document Title":"IoT"))
AND ("Document Title":"smart cit*")
AND (("Document Title":"Literature Review")
OR ("Document Title":"survey")
OR ("Document Title":"review"))
AND ("security" NEAR/10 "architecture")
AND (NOT ("Document Title":"Honeypot*"))
AND (NOT ("Document Title":"Health*"))
```

First, the proximity operator (NEAR/10) requires that "security" and "architecture" appear within 10 words of each other anywhere in the document. This ensures the paper addresses security architectural concerns rather than mentioning these terms in unrelated contexts or separate sections. The proximity constraint is crucial because papers might mention security in one paragraph and architecture in another, unrelated paragraph, which would not indicate a focus on security architectures.

Second, strategic exclusions remove unwanted research directions using NOT operators with wildcards. Papers with "Honeypot*" in the title (where * captures variations like "Honeypots" or "Honeypotting") are excluded because honeypot research focuses on decoy systems for attracting attackers rather than production security architectures for smart cities. Similarly, "Health*" is excluded to remove healthcare IoT

papers (covering “Health”, “Healthcare”, “Healthtech”), which have fundamentally different security requirements and

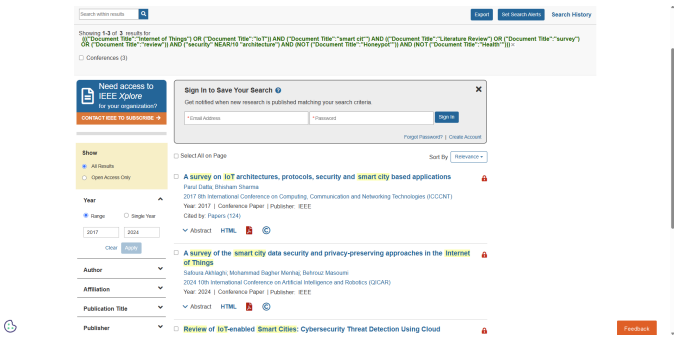

Fig. 1. IEEE Xplore search results for the refined query on IoT security architectures for smart cities. The results show survey and review papers addressing security architectural frameworks while excluding healthcare and honeypot research.

regulatory contexts (e.g., HIPAA compliance) than those of smart city infrastructure. These combined refinements reduce results from 44 to approximately 3 highly targeted papers, demonstrating how proximity operators and exclusions can achieve laser-focused results for specific research questions. Figure 3 shows the search results from the most refined query. The retrieved papers specifically address security architecture concerns in IoT smart city deployments through survey and review methodologies, such as [1], demonstrating how systematic use of field filters, proximity operators, wildcards, and exclusions produces highly targeted results.

III. ACM DIGITAL LIBRARY:

This research investigates the application of machine learning and artificial intelligence techniques to automate various aspects of software testing, including test case generation, test automation, and defect prediction. The topic “Machine Learning Techniques for Automated Software Testing” is highly relevant to modern software development practices, where AI-driven approaches increasingly augment traditional testing methodologies to improve efficiency and test coverage.

A. Initial Broad Search

This initial query casts a wide net by combining multiple alternative terms for machine learning and AI techniques with various testing-related terms:

```
(”machine learning” OR ”deep learning” OR
”artificial intelligence” OR AI OR ML)
AND (testing OR ”test automation” OR
”software testing”)
```

The use of OR operators within parentheses creates synonym groups, ensuring we capture papers regardless of which terminology authors prefer. Including both full terms (e.g., “machine learning”) and common abbreviations (e.g., ML, AI) is important because different research communities use different conventions. The query uses quotation marks around multi-word phrases to ensure they are searched as units. This broad search returns approximately 99,999 results, covering all aspects of AI/ML in testing contexts, including papers where testing is mentioned only tangentially or where ML is a minor component. This initial search is intentionally comprehensive to understand the full scope of available literature before applying refinements.

B. Adding Field Restrictions and Software Context

The second query adds field-specific restrictions to improve precision dramatically:

```
Title:(”machine learning” OR ”deep learning”
OR AI OR ML) AND (test* OR testing OR
software)
```

By requiring ML/AI terms to appear in the Title: field, we ensure these techniques are the primary focus of the paper rather than just mentioned in passing. Papers with ML/AI in the title are fundamentally about these approaches, not papers that merely reference them in related work sections. The wildcard test* captures variations including “test”, “tests”, “testing”, and “tester”, providing flexibility without sacrificing precision. Adding the general term “software” ensures we maintain a software engineering context rather than ML/AI applications in other domains, such as medical diagnosis or financial prediction. This refinement reduces results from 99,999 to approximately 15,754 papers, a dramatic reduction that demonstrates the power of field-specific searching. The remaining papers genuinely focus on ML/AI as the main contribution to software testing problems.

C. Final Refined Search with Proximity and Exclusions

The final query applies two additional powerful refinement techniques that transform an already focused set of papers into a highly targeted collection suitable for in-depth review:

```
Title:(”machine learning” OR ”deep learning”
OR AI OR ML) AND Title:(test*) AND
(”test generation”~10 OR
”test automation”~10 OR
”defect prediction”~10) AND NOT
(survey OR review OR blockchain OR IoT)
```

First, proximity operators (ACM’s tilde notation ~10) require that specific concept pairs appear within 10 words of each other, indicating these terms are discussed together as unified concepts rather than mentioned separately. For example, ”test generation”~10 ensures papers specifically address the generation of test cases using ML, not papers that separately mention testing in one section and generation in another unrelated context. Similarly, ”test automation”~10 and ”defect prediction”~10 focus results on specific testing applications rather than general testing discussions. This proximity requirement also adds a Title: restriction on test*, ensuring test-related concepts appear prominently in the paper’s title

that synthesises existing work (which would be relevant for a different research question but not for understanding

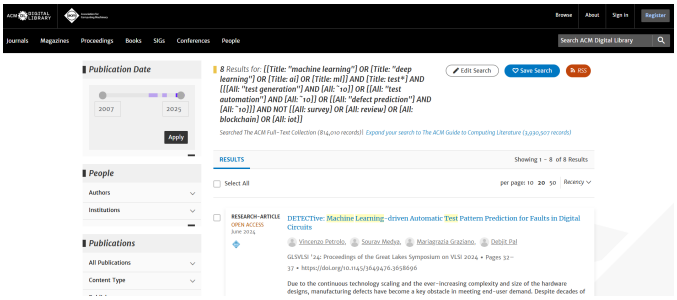

Fig. 2. ACM Digital Library search results for the refined query on machine learning techniques for automated software testing. The results show primary research papers applying ML/AI specifically to test generation, test automation, or defect prediction, excluding surveys and specialised application domains.

specific ML techniques for testing). Additionally, "blockchain" and "IoT" are excluded because papers focused on these specialised domains have different constraints, architectures, and concerns than general software testing applications. These combined refinements reduce results from 15,754 to approximately 8 highly targeted papers, a substantial reduction that demonstrates how proximity searching, combined with strategic exclusions, produces laser-focused results. The final set comprises papers that specifically apply ML/AI techniques to concrete software testing challenges with primary research contributions.

As shown in Figure 2, the refined search returns papers that specifically address machine learning applications to concrete software testing challenges, such as [3]. The progression from over 99,999 results to just 8 demonstrates how systematic application of field restrictions, proximity operators, and strategic exclusions transforms an overwhelming literature landscape into a manageable, highly relevant set of papers suitable for in-depth review.

IV. Scopus

This research investigates agile methodologies and practices specifically adapted for geographically distributed software development teams, focusing on primary research papers (empirical studies, case studies, experiments) that report original findings about methodological practices. The topic "Agile Software Development Methodologies for Distributed Teams" addresses the challenges and solutions to agile practices when teams cannot co-locate, an increasingly relevant concern in modern software engineering, particularly in the postpandemic era, when remote work has become prevalent.

A. Initial Broad Search

This initial Scopus query uses the TITLE-ABS-KEY field code to search within titles, abstracts, and author-specified keywords simultaneously:

```
TITLE-ABS-KEY(agile AND
"distributed team*" AND
"software development")
```

The query combines three core concepts: agile methodologies, distributed teams (with wildcard team* capturing both "team" and "teams"), and software development context. Scopus syntax differs significantly from IEEE Xplore and ACM, requiring specific field codes and formatting conventions. Scopus requires explicit Boolean operators (AND) to combine multiple search terms within field codes, and uses quotation marks to search for exact phrases. This broad search returns approximately 152 results covering various aspects of distributed agile development, including case studies, surveys, experience reports, theoretical papers, and tool demonstrations. While this provides a comprehensive view of the literature landscape, it is too diverse and includes too many secondary sources for a focused literature review based primarily on original research.

B. Adding Field Restrictions

The second query adds field-specific restrictions to improve precision:

```
TITLE(agile) AND
TITLE-ABS-KEY("distributed team*") AND
TITLE-ABS-KEY("software development")
```

By requiring "agile" to appear specifically in the TITLE field rather than just anywhere in the title-abstract-keywords combination, we ensure that agile methodologies are the primary focus of the paper rather than a secondary topic or mentioned only in passing. Papers with "agile" in the title are fundamentally about agile practices, not papers that merely reference agile briefly in their related work or background sections. The distributed teams and software development terms remain in TITLE-ABS-KEY to maintain reasonable recall while ensuring these concepts are prominent. This refinement reduces the results from 152 to approximately 91 papers, a significant reduction that demonstrates how field-specific restrictions can effectively narrow results by ensuring that core concepts appear in prominent document locations. The remaining papers centre on agile as their main contribution, though they still include both primary and secondary research and may not all focus specifically on methodological practices.

C. Final Refined Search with Proximity, Temporal Filters, and Exclusions

The final query applies multiple sophisticated refinement techniques that showcase Scopus's advanced capabilities and transform a moderately focused set into a highly targeted collection of recent primary research:

```
TITLE(agile) AND
TITLE-ABS-KEY("distributed" W/5 "team*") AND
TITLE-ABS-KEY("software" W/3 "development")
AND TITLE-ABS-KEY(method* OR practice*) AND
PUBYEAR > 2019 AND NOT TITLE(blockchain OR
review OR survey OR "systematic literature"
OR "literature review")
```

First, proximity operators (Scopus' W/n notation) and wildcards work together to ensure genuine conceptual relationships. The W/5 operator means "distributed" and "team*" must appear within 5 words of each other (in any order), indicating papers discussing distributed teams as a unified concept rather than mentioning these terms separately in different contexts. Similarly, W/3 ensures "software" and "development" appear within 3 words, maintaining the software engineering context. The wildcards method* and practice* capture variations such as "methodology", "methodologies", "methods", "practice", "practices", and "practitioner", ensuring that papers address methodological or practical aspects rather than just theoretical discussions.

Second, temporal filters and strategic exclusions combine to focus on recent primary research while removing unwanted paper types and specialised domains. The temporal filter (`PUBYEAR > 2019`) restricts results to publications from 2020 onwards, ensuring the review reflects current practices, which is particularly important in rapidly evolving fields like distributed agile development, where practices have changed significantly, especially following the COVID-19 pandemic's impact on remote work. Strategic NOT exclusions in the title remove multiple unwanted categories: "review", "survey", "systematic literature", and "literature review" are excluded to focus on primary research papers (case studies, empirical studies, experiments) rather than secondary literature, aligning with the principle that literature reviews should be based primarily on original research contributions. Additionally, "blockchain" is excluded to remove papers where agile is discussed only in the context of this specialised domain. These combined refinements reduce the results from 91 to approximately 29 highly targeted recent primary research papers, a noticeable reduction that demonstrates how proximity operators, temporal filters, and strategic exclusions together produce laser-focused results suitable for a comprehensive yet manageable literature review.

Figure 3 presents the search results from Scopus for the most refined query. The retrieved papers, such as [2], specifically report original research on agile methodological concerns in distributed team contexts, demonstrating how database-specific syntax (W/n proximity, TITLE vs TITLEABS-KEY distinctions), field restrictions, wildcards, temporal filters (PUBYEAR), and strategic exclusions (AND NOT) produce highly focused results centered on primary research suitable for evidence-based literature reviews.

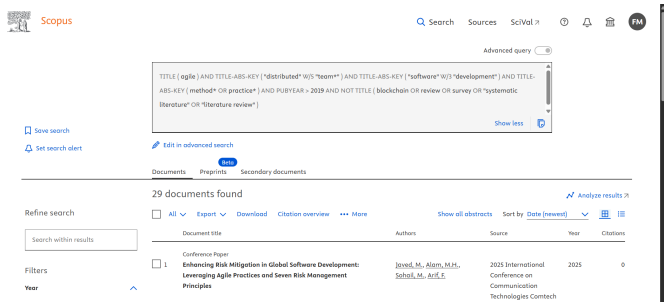

Fig. 3. Scopus search results for the refined query on agile methodologies for distributed software development teams. The results show recent primary research papers addressing methodological practices for geographically dispersed agile teams, excluding secondary literature and specialised domains.

V. Conclusion

This exercise illustrated how to conduct effective literature searches using advanced query techniques across three major academic databases. The critical lessons are:

- Use proximity operators to ensure related concepts appear near each other, indicating genuine relationships rather than coincidental mentions.
- Use wildcards to capture word variations and improve recall without sacrificing precision.
- Target specific fields to ensure core concepts appear in prominent locations, significantly improving relevance.
- Use exclusions strategically (NOT) to remove unwanted subdomains or contexts that would otherwise dilute results with irrelevant papers.
- Include synonyms and related terms to avoid missing relevant papers that use different vocabulary or terminology.
- Combine Boolean operators strategically using AND, OR, and NOT to express complex search requirements.
- Adapt queries to each database's unique syntax requirements and field codes.
- Apply publication year restrictions, as currency is important for rapidly evolving topics.
- Start broadly to understand the literature landscape, then systematically narrow through multiple query iterations, aiming for one to a few dozen highly relevant papers for a focused literature review.

References

[1] Safoura Akhlaghi, Mohammad Bagher Menhaj, and Behrouz Masoumi. "A survey of the smart city data security and privacy-preserving approaches in the Internet of Things". In: *2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)*. 2024, pp. 379–385. DOI: 10.1109/QICAR61538.2024.10496658.

[2] Maria Javed et al. "Enhancing Risk Mitigation in Global Software Development: Leveraging Agile Practices and Seven Risk Management Principles". In: *2025 International Conference on Communication Technologies (ComTech)* (2025), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:279454584.

[3] Muralidhar Yalla and Asha Sunil. "AI-Driven Conversational Bot Test Automation Using Industry Specific Data Cartridges". In: *2020 IEEE/ACM 15th International Conference on Automation of Software Test (AST)*. 2020, pp. 105–107.