A Systematic Review of Relational Databases and NoSQL Databases

Sahabdeen Aysha Asra[1] & Abdul Cader Mohamed Nafrees[2]

[1]*Department of Information Technology, South Eastern University of Sri Lanka*
[2]*Faculty of Islamic Studies and Arabic Language, South Eastern University of Sri Lanka*

[1]ashrasahabdeen005@seu.ac.lk

Abstract

The volume (petabyte to a terabyte), variety (unstructured, structured, and hybrid), and velocity of the digital world are all increasing quickly. The standard DBMS, which provides the Relational Database (RDB), guarantees the consistency of transactions and the integrity of the data. These are the standards for a good DBMS, which are applicable to numerous software programs. However, NoSQL (Not Only SQL) databases have emerged in recent years as a response to the problem after traditional databases failed to address the speed at which data was growing. These two types have different benefits and drawbacks from one another despite being used for the same tasks (creating, retrieving, updating, and managing data). This study used a systematic review technique to collect data from previously published studies. The data were collected from reputable publishers, and these data were evaluated independently to compare the RDB and NoSQL. Confidentiality, data integrity, auditing, authentication, customer interaction, Big Data technologies, Cloud technologies, complexity, performance, cost, and variety are the main differing attributes that are included in the main categories such as security, technologies, and so on. In addition, this study includes the benefits and drawbacks of RDB and NoSQL as well. Especially, RDBMS pay more attention to data consistency than NoSQL databases do. As a result, NoSQL creates a lot more vulnerability than RDBMS. NoSQL databases should use an external technique to conduct and ensure database security because they have a lot of experience with the expansion of big data and can easily handle a massive rise in data. Given the expansion and development of NoSQL database systems, its potential is vast. The majority of contemporary software and applications rely on the web, and over the coming years, the amount of data that needs to be stored is likely to rise significantly. These reasons suggest that NoSQL databases will develop significantly over the next few years, expand, and eventually find ways to address their security concerns. Furthermore, the discussion deeply analyses the comparison between RDB and NoSQL. Due to the restrictions caused by the scarcity of publications in online sources and also the lack of open access possibilities, this work was severely suppressed. Likewise, this study's principal weakness was the dearth of newly published studies. More studies on relational databases and NoSQL databases for online information systems should be done, according to our recommendation.

Keywords: RDB, NoSQL, database, Systematic Review, Information system

Introduction

The most used database is RDBMS, which stands for relational Database Management Systems. Data is stored in the form of a tuple in the form of a row. It has a number of tables, and data stored in the tables may be quickly retrieved. A non-SQL database is referred to as a NoSQL database. In contrast to relational databases, NoSQL databases do not store data in tables. It is used to store and retrieve data from databases, so it is commonly utilized to hold massive amounts of data. It has higher performance as well as enables query language (Nance et al., 2013).

The relational database management system is meant to give data in rows and columns or in a suitably formatted manner for relational databases. For the relational model of data, the database is delivered electronically. Tables are used to hold data, so each table has a unique identity. This identification, known as the primary key, is used to link tables together. NoSQL is the name given to a database that stores and retrieves data without using a primary key. Inserting data into the database does not need the use of a correct schema. As a result, modifications may be made more quickly (Battersby, 2012).

There are several distinctions between RDB and NoSQL databases. RDBMS is well-known among users since it is an ancient database that many firms rely on for data formatting. Because this database is still relatively new, there are less professionals in NoSQL. RDBMS user interface tools for data access are available on the market, allowing users to experiment with all of the RDBMS infrastructure's schema. This makes it easier to engage with the data, so consumers will have a better understanding of it. In NoSQL, the number of user interface tools for accessing as well as manipulating data is quite limited, therefore users don't have many alternatives for interacting with data (Abdullah & Zhuge, 2015). When the data is large, RDBMS (Sahatqija et al., 2018) scalability and performance suffer. Servers may not be able to handle the available load adequately, resulting in performance difficulties. It performs admirably under heavy loads. In NoSQL, scalability is excellent. When compared to RDBMS, this improves the database's efficiency. Users could effortlessly manage a large volume of data (Sundhara Kumar et al., 2017).

Literature Review

The classic DBMS provides the relational database, which assures data integrity and transactional consistency. These are the principles of a suitable DBMS for various software applications. However, as a result of the rapid expansion of data and the lack of support from traditional databases for this issue in recent years, NoSQL (Not Only SQL) databases have emerged as an alternative (Sahatqija et al., 2018). Non-relational databases, on the other hand, have exploded in popularity in recent years. These databases are usually referred to as NoSQL, which distinguishes them from standard SQL databases. The majority of them are founded on the notion that storing simple key-value pairs leads to speed (Li & Manoharan, 2015).

It is undeniable that NoSQL databases are a byproduct of the Web 2.0 era – they were only used when the developers of web services with a lot of users found that traditional relational database management systems (RDBMS) are only suitable for small but regular read/write exchanges or large batch transactions with few write access privileges, not for massive read/write workloads (as is often the case for these large scalability projects) (Tudorica & Bucur, 2011). In a database management system, the RDBMS has been the most

important model. In recent years, a new database model known as NoSQL (Not simply SQL) has emerged as a viable alternative to existing database architectures. The purpose of NoSQL is to function effectively for applications that don't use the relational database paradigm, rather than to totally replace SQL (Sundhara Kumar et al., 2017).

Especially for the NoSQL Databases a database called Mongo, which name is derived from the word "humongous," is designed for developers with reasonably large amounts of the information who also want something that requires little upkeep and is simple to use. It is a document-oriented system with records that resemble JSON objects as well as support nested feature searching. Both automated Map Reduce and sharding operations are supported. In addition to an interactive shell but also bindings for all of the other widely used languages, queries are done in JavaScript (Uyanga et al., 2021).

An n-ary relation is a subset of the Cartesian product of N domains and is used to represent all data in relational databases, which are founded on the mathematical ideas of set. The database's data is organized into relations as well as represented as tuples. The connection, which is depicted by a table, consists of a set of triples (rows) that correspond to the sequence of attributes named columns. The type of an attribute is specified by the domain, which is a collection of values with a common meaning. This data model is well-structured and quite specialized. A clearly designed schema describes the columns. Rows of connected data all have the same structure. Key value stores, graphs, and document data models are just a few of the modeling strategies used by NoSQL databases. Although NoSQL database systems occasionally use two or more data models to represent the data, NoSQL is a category that derives its name of kind from its database schema. The NoSQL data model is primarily characterized by the fact that it does not employ tables as the data's storage structure. It is also schema-less and effective at managing complex data, such as word or pdf files, photos, but also video files, among others (Obay et al., 2014).

The notion of the Internet of Things (IoT) has been around in the IT sector for a few years. IoT is concerned with the interconnection of a large number of smart devices. IoT will have applications in a variety of fields in the near future, and these applications will generate massive amounts of data (Ali et al., 2019). The difficulty of efficiently storing, transferring, and managing data sets has arisen as a result of the continual production of heterogeneous data. Traditional database systems relied on Structured Query Language (SQL) databases, which met all of the user's needs while also being simple, resilient, flexible, scalable, and fast (Rautmare & Bhalerao, 2017). With the increasing rise of network data, NoSQL databases have become extensively utilized and, to some extent, have replaced conventional databases. Presently, NoSQL databases may be used in conjunction with relational databases to compensate for their shortcomings. This database combination has significant relevance in resolving performance, scalability, and other Web2.0 difficulties (Naheman & Wei, 2013).

This study carefully discusses the major findings of previously published article results based on the research techniques explored in the methodology section below. Further, the Results and discussion section summarize all the findings and briefly summarized the major findings, limitations, and future directions in the conclusion area.

Methodology

This paper was written using a qualitative method known as systematic review, which was based on previously published research and review articles over the last five years. Where the acquired data was examined using a qualitative manner to investigate the uses, benefits, as well as drawbacks of chosen.

I. Article Selection Criteria

To shortlist the downloaded publications from reputable publishers such as IEEE, Springer, Emerald, Inderscience, and Sage, the following important factors were evaluated. The following are the criteria that help to select the previous articles;

- Only full-length articles will be considered.
- Articles having a high index in citation databases.
- Open access articles
- Published in English

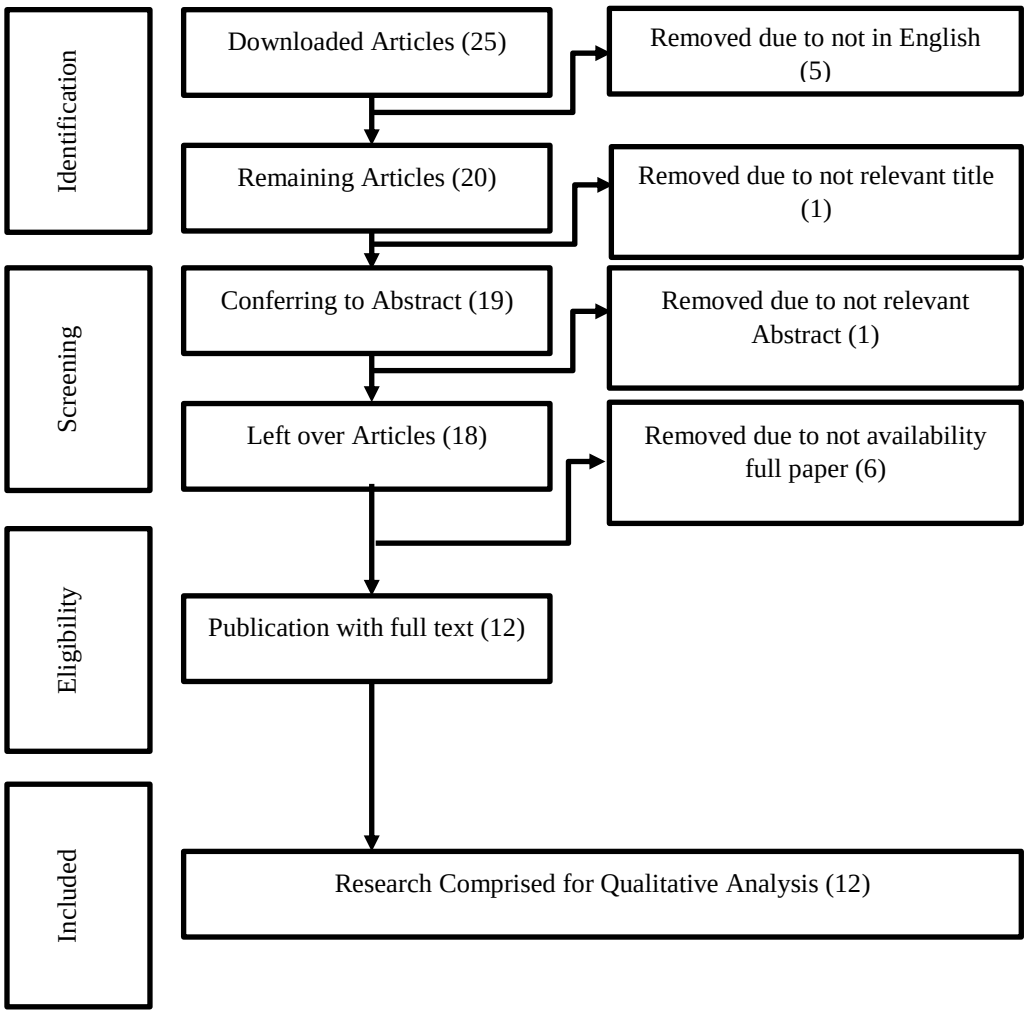

Fig 1: Classification of select the published articles

Table 1: Research Question and motivation

S. No	RQ	Motivation
1	What is define by Relational Databases and NoSQL Databases?	Defining the Relational databases and NoSQL Databases.
2	What are major difference between Relational Databases and NoSQL databases?	Identifying attributes within the category to compare the Relational databases and NoSQL Databases.
3	What are the advantages and drawbacks of Relational Databases and NoSQL Databases?	Find the advantages and disadvantages of Relational databases and NoSQL Databases.

Results and Discussion

SQL databases are relational databases that feature a table-based data structure and must adhere to a rigid, established schema. Document databases, graph databases, key-value pairs, and broad stores are all examples of NoSQL databases or non-relational databases. NoSQL databases should not need a preset schema, allowing you to interact with "unstructured data" more freely. Relational databases are vertically scalable, however, they are normally more costly, whereas NoSQL databases are much more cost-effective due to their horizontal scalability characteristics (Gupta et al., 2018). NoSQL databases were introduced to overcome the restrictions of relational database technology. When compared with relational databases, NoSQL databases are more scalable and provide higher performance. In addition, the flexibility and an extensive selection of data models speed up the development when compared to the relational model, especially in cloud computing (Radoev, 2017).

Database administrators tend to purchase larger servers as the database load will increase. RDBMS will not be scaled out simply on service clusters. But NoSQL databases are designed to grow new nodes on scaling up, which expressively moderates the cost of service hardware (Abdullah & Zhuge, 2015). The design and query languages of NoSQL differ extensively between many NoSQL products as well as traditional databases. Another downside if NoSQL databases are backups. Some NoSQL databases provide tools for the backup process but they are not insufficient to guarantee the accuracy of data (Obay et al., 2014).

Table 2: Comparison between RDB and NoSQL Database

Reference	Category	Attributes	Relational Databases	NoSQL Databases
(Obay et al., 2014)	Security	Confidentiality	Relational databases frequently accomplish data secrecy since they utilize encryption	Data confidentiality cannot be established since data is typically stored in plain view.

			methods to store encrypted data.	
		Data Integrity	Relational databases utilize ACID principles to verify that database transaction are consistently performed and that data integration.	Since eventual consistency is one of the BASE fundamental principles, data integrity in NoSQL databases is not always attained.
		Auditing	Offer methods for auditing that permit writing to syslog or xml files in the database. Most relational databases also offer more sophisticated auditing, such as Oracle Fine Grained Auditing, which permits audit in specific situations.	NoSQL databases generally don't offer auditing. Some databases that offer auditing have problems.
		Authentication	Every relational database has an authentication method built in, and you may utilize any of those mechanisms.	Most NoSQL databases don't include an authentication or authorization mechanism by default, but they can employ an outside way to do this task.
		customer interaction	Relational databases use SSL and encryption to enable secure client connection methods.	The majority of NoSQL databases don't really offer safe client communication protocols.
(Palanisamy & Suvithavani, 2020)	Technologies	Big Data	Big data management in RDBMS is a really serious problem. So because table structure causes a significant degree of difficulty.	"Not just SQL" databases developed techniques to enhance the display of storing as well as retrieving information in order to handle the huge amount of data.
		Cloud	Relational databases do not provide complete content data search and	Since all of the characteristics of NoSQL databases are highly

		are difficult to grow beyond a certain point. Therefore, it is not suitable for cloud settings	desirable for cloud databases, "Not just SQL" databases are the best response for cloud databases. Additionally, semi-structured as well as unstructured data is handled by cloud databases.
(Uyanga et Others al., 2021)	Consistency	Has a firm adherence to the rigid schema	Has inconsistent use of a schema-free approach.
	Complex ity	In situations when user-storage data is difficult to translate into tables, build complicated data.	hold both less complicated unstructured and semi-structured data
	Performance	Sluggish because they need a lot more time to comprehend information.	Usually offers higher query performance
	Cost	costly method for storing data	Inexpensive due to open source as well as low-cost upgrade
	Variety	closed as well as open source platforms	Primarily open source NoSQL

According to the table there are security, technology and other categories are included to define the attributes. NoSQL databases may be the ideal option when we want to expand on the options that rdbms can provide us, for instance, in situations where scalability, availability, and performance are crucial. NoSQL databases can be the ideal solution when a software application does not require or prioritize data integrity. Relational databases' complexity can be avoided with NoSQL. The platform's size is another factor that needs to be considered. If changing the data model at any time is required without having an impact on the system as well as application's functionality, a NoSQL database is the solution. Such databases were created in order to enhance and grow the technology systems while creating a flexible also scalable business environment. Relational databases should be the solution if data integrity and consistency are important requirements for software applications. The security of transactions and other sensitive acts must be ensured on some platforms since they frequently handle sensitive data. For instance, the only trustworthy alternative is RDB since sensitive data like accounting records; bank transactions and finance record require confidentiality as well as consistency of data as well as the ability to complete multiple tasks in a single atomic operation. The relational database should be used when working with data that must connect various collections into a single outcome. In addition to security, consistency and availability should be taken into consideration as a crucial factor.

We must first fully understand the requirements of the system, the limits the application must adhere to, as well as the potential problems that may arise during development before choosing the technologies and also working environment for our software program. Varied software applications have various demands and requirements, which makes it simple to identify the proper technologies that we should pick based on these differences. There are certain crucial DBMS elements that can be helpful in this approach.

For modern applications with much more complicated, continuously changing data sets that require a flexible data model that does not need to be built instantly, NoSQL is a preferable solution. Most NoSQL database programmers or companies are drawn to the agile capabilities that allow them to get to market faster and make upgrades faster. NoSQL databases, unlike traditional SQL-based relational databases, can store as well as handle data in real-time. While SQL databases have some unique use cases, NoSQL databases have numerous characteristics that SQL databases cannot handle without incurring significant expenses as well as making crucial tradeoffs in terms of performance, flexibility, as well as other factors (Rautmare & Bhalerao, 2017). Despite its rapid development, NOSQL currently trails behind relational databases in terms of user numbers. The fundamental reason for this is because SQL is more known to users, whereas NOSQL databases do not have a standard query language. It will undoubtedly be a game changer if a common query language for NOSQL is established. The majority of NoSQL databases are general-purpose, whereas SQL databases are better suited to specialized purposes (Sahatqija et al., 2018). As a result, if you need to store a large number of different sorts of variables and utilize all of the SQL query capabilities, you won't be able to locate a SQL database that is acceptable for your needs. This specificity is also why a document schema is specified in a SQL database (Naheman & Wei, 2013). Data structure is one of the most critical aspects of any system, and while NoSQL systems may produce any design, SQL databases are more specific. There is often a distinction between the entities that can be stored in a SQL database and those which can be stored in a NoSQL database. In general, NoSQL databases are better for storing huge volumes of data in tables with many columns or records, whereas SQL databases are better for storing data in tables with multiple columns and records. Only the amount of memory available to the machine limits the number of documents that may be kept in a SQL database (Battersby, 2012). Furthermore, you will need to utilize SQL syntax if you need to execute complex searches to get information that you can't achieve with a simple inquiry. You'll also need a SQL explorer to view the database structure of the NoSQL system. Some NoSQL databases include strong queries that may be used to run more complex queries. These searches, however, do not apply to the data they are holding (Sahatqija et al., 2018).

NoSQL databases are classified into several categories based on how they store data. Some NoSQL databases maintain data in the file, while others merely keep data in JSON or XML data fields, yet others keep all data in JSON. These are only examples; you must ensure that your NoSQL database is appropriate for the task at hand. Moreover, when considering consistency of the data RDBMS takes more consideration on it than NoSQL databases. Therefore, NoSQL leads to much insecurity compared with RDBMS. If you enter the same data all over again, it will take it without any errors while the relational database gives errors for duplicate entries (Obay et al., 2014).

Conclusion

The term NoSQL (Not Only SQL) refers to a class of non-relational database solutions. Non-relational data stores might be a better name for the items in this area than NoSQL databases. Graph-oriented storage systems are the most similar to regular databases in terms of properties, and they have the ability to grow into full databases (Radoev, 2017). NoSQL databases are classified into several categories based on how they store data. A few NoSQL databases maintain data in files, while others merely keep data in JSON or XML data fields, while yet others keep all data in JSON. These are only examples; you must ensure that your NoSQL database is appropriate for the task at hand.

In overall Furthermore, when it comes to data consistency, RDBMS pay more attention to it than NoSQL databases. As a result, as contrasted to RDBMS, NoSQL creates a lot of vulnerability. It will accept the same data without problems if you enter it again, however a relational database will give you an alert if you submit the same data again. NoSQL databases, in general, have a lot of experience with the expansion of big data and can easily handle a massive rise in data, but there aren't many security concerns, so they should use an external technique to conduct and ensure database security. Its possibilities are great given the growth and advancement of NoSQL database systems. The majority of modern software and applications are dependent on the web, as well as the amount of data that needs to be stored is likely to increase significantly over the next few years. These factors lead us to believe that NoSQL databases will experience significant expansion and advancement over the coming years and will eventually find solutions to their security issues. Relational and NoSQL databases can be configured together as well as work seamlessly together in a mixed system that uses both types of technologies. Additionally, they can combine the advantages of each technological advancement to produce a system that is more efficient.

This study was highly suppressed by the limitations due to very few publications available in the online sources and fewer options provided for open access. Similarly, less number of recently published articles was another main limitation of this study. We suggested conducting more research studies on relational databases and NoSQL databases for online information systems.

References

Abdullah, A., & Zhuge, Q. (2015). From Relational Databases to NoSQL Databases: Performance Evaluation. *Research Journal of Applied Sciences, Engineering, and Technology*, *11*(4), 434–439. https://doi.org/10.19026/rjaset.11.1799

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian Journal of Research in Computer Science*, *October*, 1–10. https://doi.org/10.9734/ajrcos/2019/v4i230108

Battersby, S. (2012). How a 3000-year-old code unmasked a stellar cannibal. *New Scientist*, *216*(2895), 43–45. https://doi.org/10.1016/S0262-4079(12)63205-9

Gupta, A., Tyagi, S., Panwar, N., Sachdeva, S., & Saxena, U. (2018). NoSQL databases: Critical analysis and comparison. *2017 International Conference on Computing and Communication Technologies for Smart Nation, IC3TSN 2017, 2017-Octob*, 293–299. https://doi.org/10.1109/IC3TSN.2017.8284494

Li, Y., & Manoharan, S. (2015). A performance comparison of SQL and NoSQL databases A performance comparison of SQL and NoSQL databases. *978-1-4799-1501-9/13- 2013 Ieee*, *November*, 15–19. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6625441

Naheman, W., & Wei, J. (2013). Review of NoSQL databases and performance testing on HBase. *Proceedings - 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013*, 2304–2309. https://doi.org/10.1109/MEC.2013.6885425

Nance, C., Losser, T., Iype, R., & Harmon, G. (2013). Association for Information Systems AIS Electronic Library (AISeL) NOSQL VS RDBMS -WHY THERE IS ROOM FOR BOTH NOSQL VS RDBMS -WHY THERE IS ROOM FOR BOTH. *Proceedings of the Southern Association for Information Systems Conference, Savannah, GA, USA*, 111–116. http://aisel.aisnet.org/sais2013%5Cnhttp://aisel.aisnet.org/sais2013/27

Obay, M. A. M., Altrafi, G., & Ismail, M. O. (2014). Relational vs . NoSQL Databases : A Survey. *International Journal of Computer and Information Technology*, *03*(03), 2279–2764.

Palanisamy, S., & Suvithavani, P. (2020). A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*. https://doi.org/10.1109/ICCCI48352.2020.9104047

Radoev, M. (2017). A Comparison between Characteristics of NoSQL Databases and Traditional Databases. *Computer Science and Information Technology*, *5*(5), 149–153. https://doi.org/10.13189/csit.2017.050501

Rautmare, S., & Bhalerao, D. M. (2017). MySQL and NoSQL database comparison for IoT application. *2016 IEEE International Conference on Advances in Computer Applications, ICACA 2016*, 235–238. https://doi.org/10.1109/ICACA.2016.7887957

Sahatqija, K., Ajdari, J., Zenuni, X., Raufi, B., & Ismaili, F. (2018). Comparison between relational and NOSQL databases. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 216–221. https://doi.org/10.23919/MIPRO.2018.8400041

Sundhara Kumar, K. B., Srividya, & Mohanavalli, S. (2017). A performance comparison of

document oriented NoSQL databases. *International Conference on Computer, Communication, and Signal Processing: Special Focus on IoT, ICCCSP 2017*. https://doi.org/10.1109/ICCCSP.2017.7944071

Tudorica, B. G., & Bucur, C. (2011). A comparison between several NoSQL databases with comments and notes. *Proceedings - RoEduNet IEEE International Conference*. https://doi.org/10.1109/RoEduNet.2011.5993686

Uyanga, S., Munkhtsetseg, N., Batbayar, S., & Bat-Ulzii, S. (2021). A Comparative Study of NoSQL and Relational Database. *Smart Innovation, Systems and Technologies*, *212*(1), 116–122. https://doi.org/10.1007/978-981-33-6757-9_16