INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Comparison of Apache SparkSQL and Oracle Performance: Case Study of Data Cleansing Process

Ilma Nur Hidayati [a,*], Tien Fabrianti Kusumasari [a], Faqih Hamami [a]

[a] *School of Industrial and System Engineering, Telkom University, Bandung, 40257, Indonesia*

Corresponding author: [*]ilmanurhidayati@student.telkomuniversity.ac.id

***Abstract*— A dataset with good quality is a valuable asset for a company. The data can be processed into information to help companies improve decision-making. However, the data increased more and more over time to decrease data quality. Thus, good data management is important to keep data quality meeting company standards. One of the efforts that can be done is conducting data cleansing to clean data from errors, inaccuracies, duplication, format discrepancies, etc. Apache Spark is an engine that can analyze large amounts of data. Oracle Database is a database management system used to manage databases. Both have their own reliability and can be used to analyze SQL-shaped data. This study compared Spark and Oracle performance based on query processing time. Both were tested on queries used to perform data cleansing of millions of rows of the dataset. The research focuses on finding out Spark and Oracle's performance through quantitative analysis. The results of this study showed that there were differences in query processing times on both tools. Apache Spark is rated better because it has a relatively faster query processing time than Oracle Database. It can be concluded that Oracle is more reliable in storing complex data models than in analyzing large data. For future research, it is suggested to add other comparison aspects such as memory and CPU usage. The researchers can also consider using query optimization techniques to enrich query experiments.**

***Keywords*— Spark; Oracle; cleansing; processing time; comparison.**

Manuscript received 14 Jan. 2022; revised 17 Mar. 2022; accepted 20 Apr. 2022. Date of publication 31 May 2022.

I. INTRODUCTION

Data is a fact that can be obtained from various sources to be processed into information by individuals and organizations. For a company, data is a very valuable asset because it can be used to support its business purposes [1], [2]. However, before it can be used to support business purposes, data needs to be processed first to provide information of high quality. Data is high quality if it has properties according to its business circumstances, describes business processes, and company planning to support decision-making [3]. In addition, several indicators state that a data has good quality, namely consistency, integrity, timeliness, and usability [4], [5], [6]. Without good quality, companies cannot make good decisions because of ignorance of the conditions that occur in the company [7].

Meanwhile, data can grow more and more over time, so companies need to ensure their data quality [8]. However, the more data, the more problems occur the data. The problem can be unstructured forms of data that do not comply with the desired quality standards [5]. If the data is inaccurate, incomplete, and inconsistent, decision-making becomes not good enough or even wrong. Therefore, it is needed to conduct the stages of data pre-processing, standardization, and data cleansing to improve data quality [9].

This study discusses an issue at XYZ Ltd, a telecommunications company in Indonesia. The company provides internet and telephone services to tens of millions of customers. Everyone who subscribes to XYZ Ltd is required to have an account according to the number of services used. For example, if customer A subscribes to the internet and telephone, customer A must have two accounts. This kind of data management system cannot produce data of high quality. The data tends to have redundancy and data inconsistencies [10]. Based on these issues, the company migrated its data. Data migration was done to integrate customer accounts and reduce customer account data duplication. However, after migration, dirty data becomes increasingly. The dirty data contains customers' contract accounts that are no longer active. With this kind of data management, the database is running out of capacity faster. In addition, the company's receivables can continue to grow because inactive customer

accounts can continue to be recorded in arrears. Thus, data cleansing is needed to overcome these problems by eliminating duplicate data, noise, inaccuracy, and data discrepancies so that companies can make proper decision making [11], [12], [13], [14].

This study carried out data cleansing on tens of millions of rows of dataset results from ERP system transactions. In conducting data cleansing, it is necessary to consider the performance of the tools used, and da cleansing requires reliable speed in processing millions of data [15]. In addition, if a company has several tool options that can be used, then comparing the performance of the two tools was very useful in determining the efficiency of the data cleansing process [16].

This study conducted data cleansing using Apache Spark and Oracle Database. Apache Spark was chosen because it is a data analytics tool with excellent processing speed [17], [18]. Apache Spark is a tool used for distributed computing, particularly for real-time data analysis [19]. In addition, Spark is also very reliable for processing big data computing. Apache Spark is an interface used to perform data analytics on a large scale. Spark contains many methods that can be used to support the process. This program can be used to communicate with other databases such as Oracle. Spark also comes with easy access using python, Java, Scala, R, programming languages, and SQL support.

The study compared the performance of Apache Spark with Oracle Database, a database that companies use as data storage. Given that Apache Spark is a data analytics tool and Oracle Database is a database management system, it is important to compare both in this case study. Below are some literature that are used in this research.

A. Data Cleansing Process

Data cleansing is a series of processes to remove anomalies and obtain a set of data that is accurate and represents its environment [20]. In general, data cleansing contains several stages, namely (1) the stage of pre-processing data, (2) the stage of data processing, and (3) the validation and verification stages [15]. Fig. 1 shows the basic principles of data cleansing implementation. It begins with analyzing data quality problems, processing them with the most advanced algorithms or tools, and ends by verifying the results [21].

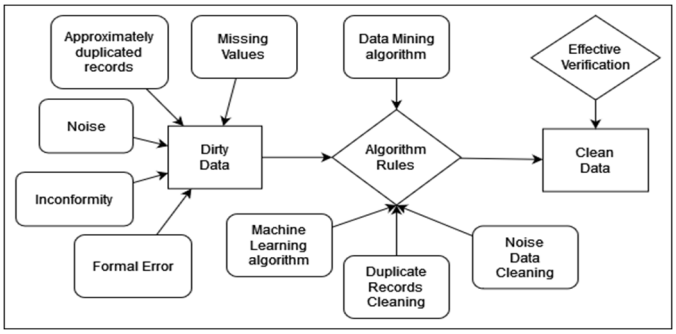

Fig. 1 Data *cleansing schema* [21]

The application of data cleansing aims to improve the quality of data by eliminating data that is not appropriate or inconsistent [11]. The first stage begins by detecting a data anomaly. Then in the second stage, it was done with the help of tools to process data anomalies based on certain rules. Finally, the third stage is to verify the consistency and accuracy of the updated data. In conducting data cleansing, it must be ascertained that the methods used do not affect the original data [15].

Tools commonly used in data cleansing are IBM Infosphere Quality Stage, DemandTools, Informatica Cloud Data Quality, and Oracle Enterprise Data Quality [22]. However, this tool requires an expensive cost. Chen and Jiang [23] used MapReduce to address lost data and incorporate information system characteristics. Sulistyo et al. [2] performed data cleansing using Pentaho Data Integration to remove the uniformity of the data format. The research shows that PDI is an open-source tool with many data processing options and can be implemented on web interfaces.

Apache Spark is a tool used for distributed computing, especially for real-time data analysis [15]. Spark is also an open-source tool and facilitates users to perform data processing functions using Spark SQL [18]. Meanwhile, Oracle Database is a database management system used to store data and provide high availability andlexible data models [24].

B. Related Work

Research on the performance of Oracle Database has been conducted by comparing NoSQL data processing with MongoDB [24]. The results of the processing time evaluation showed that in the insert, update, and delete operation, MongoDB was declared superior. The method used in this study is a quantitative analysis based on query processing time. Furthermore, research by Hazarika et al. [25] evaluated Hadoop and Spark engine performance using quantitative analysis methods. This research successfully showed that Spark has a processing time performance superior to normal and iterative queries. Next, research by Poljak et al. [26] compared MySQL, Oracle, and PostgreSQL performance in multiple indicators. In the query speed performance indicator, Oracle is better than others. Other research on performance comparisons was conducted on Apache Spark with PostgreSQL [27]. In the study, it was obtained that Apache Spark is superior because it has in-memory processing features.

There has been no similar research comparing Apache Spark and Oracle Database performance based on previous research. Therefore, this research needs to be done to determine the performance of both tools and complement previous research. This study referred to a few studies that have similarities in some parts only.

II. MATERIALS AND METHOD

In this study, there was a comparison of Apache Spark and Oracle Database about performance through data cleansing to datasets. Table 1 shows the specifications used to operate Spark SQL and Oracle SQL. Based on the table, it can be concluded that the technologies used on the server's side are different for Apache Spark and Oracle, and Oracle has higher specifications on its processor, storage, and even RAM.

TABLE I
TECHNOLOGY SPECIFICATION

No	Technology	Apache Spark	Oracle Database
1	*Processor*	Intel® Xeon® Platinum 8168 CPU @ 2.70 GHz 16 Core	Intel® Xeon® Platinum 8168 CPU @ 2.70 GHz 32 Core
2	RAM	40 GB	128 GB
3	Storage	544 GB	550 GB
4	OS	SUSE Linux Enterprise Server 12 SP5	SUSE Linux SUSE Linux Enterprise Server for SAP Applications
5	Software	Apache Zeppelin version 0.9.0	SQL Developer

The data object used in this research is data from customer accounts and customer payment transactions obtained from the SAP system at XYZ Ltd. It is why the table name of the dataset was unique and based on the SAP naming format. This SAP system uses Oracle Database to store every data. Table 2 displays a description of the FKKVKP table that contains several columns, a short description, and examples of value. FKKVP size is 722 MB and 30 million rows of data in total.

TABLE II
FKKVP TABLE DESCRIPTION

Columns	Description	Values
VKONT	Contract account number	1000001135
GPART	Business partner number	2000001135
LOEVM	Mark contract account for deletion	X = terminated contract account ‘ ‘ = unterminated contract account
MANDT	Client code	110
VKBEZ	Contract account name	John Doe

Based on Table 2, a LOEVM column contains the deletion mark for a specific contract account. If a contract account has X on its LOEVM, then it means that this account has been terminated after the migration data. Otherwise, if the LOEVM is empty, then it means that this account is still active. In this study, only active contract accounts were used as the object.

Next, Table 3 displays the columns contained in DFKKOP before being divided into OPENDFKKOP and PAIDDFKKOP tables. OPENDFKKOP table size is 1 GB and 90 million total data rows. Meanwhile, the PAIDDFKKOP table measures 2.3 GB.

TABLE III
DFKKOP TABLE DESCRIPTION

Columns	Description	Value
VKONT	Contract account number	1000001135
AUGST	Clearing status	9 = paid invoice ‘ ‘ = unpaid invoice
ZZNPER	Period	202201
BUDAT	Posting date in the document	20211231
BLART	Document type	IP

Based on Table 3, three columns need to be paid attention. First is AUGST column, which contains payment status. If a contract account has nine on its AUGST, then it means that this account has paid a specific invoice. Otherwise, if the AUGST is empty, the account has not paid the invoice. The second column, ZZNPER contains information about an invoice's creation period. The third column is BUDAT contains the posting date of the invoices. This study case takes contract accounts with paid invoices three months ago, September to November 2021. This case study also took account of unpaid invoices from June 2020 until November 2021. Those conditions are made based on the company business rules.

Fig. 2 displays the dataset's schema that includes DFKKOP and FKKVKP tables. The DFKKOP table contains customer billing data, including payment status, total invoice, and invoice due date. Meanwhile, FKKVKP contains customer account data. DFKKOP table was divided into two new tables, namely OPENDFKKOP and PAIDDFKKOP tables, to distinguish customers who have not paid their invoices.

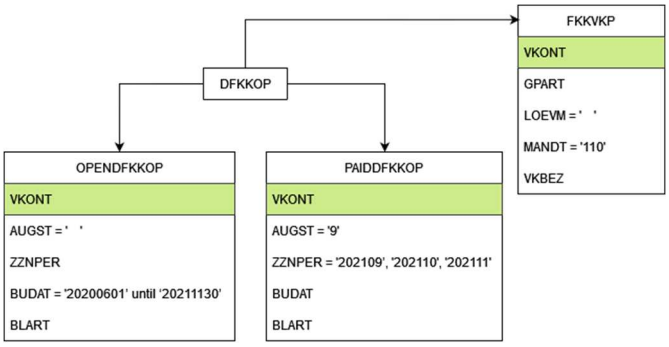

Fig. 2 Dataset schema

Based on the table schema in Fig.2, each table contains a VKONT column. This column is the contract account number used as the primary key. The VKONT column was later be used to connect FKKVKP, OPENDFKKOP, and PAIDDFKKOP tables. The OPENDFKKOP table contains contract accounts with unpaid invoices. Meanwhile, PAIDDFKKOP table contains contract accounts with paid invoices. Next, Fig. 3 displays this research process starting from the data cleansing stage using Apache Spark and Oracle.

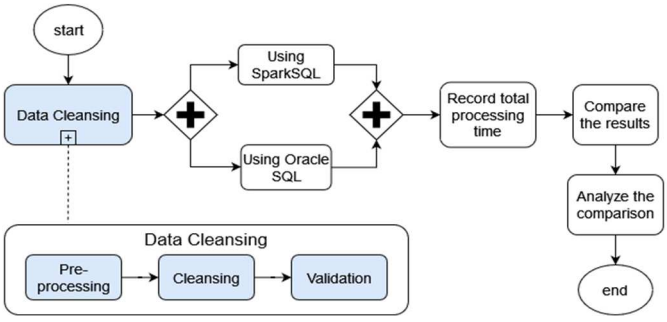

Fig. 3 Research methodology

The stages of data cleansing are selected based on research that has been done before Ridzuan and Zainon [15], namely pre-processing stage, processing stage, and validation stage. In processing using Spark, first, the data was retrieved from Oracle Database and then stored in Spark Warehouse so that there are three main tables, namely FKKVKP, OPENDFKKOP, and PAIDDFKKOP tables. The tool used to process data with Apache Spark is Apache Zeppelin. To retrieve the data, JDBC drivers connect Spark to Oracle.

After completing the process of data cleansing, this research obtained a new table containing the VKONT

column. The VKONT column contains a contract account ID that meets these conditions below:

- Has not been terminated during the migration process
- Does have unpaid invoices in the range of June 2020 until November 2021
- Does have paid invoices in the range of September to November 2021

Therefore, it is concluded that this research does not focus on the cleansing process but the performance of the tools. This research tested the performance of Spark SQL and Oracle SQL using queries at the data cleansing stage and queries at the validation stage. Here is the query that was used as an experimental object.

A. Data Cleansing Phase

After the data is successfully stored in Spark Warehouse, the next stage is to do data cleansing. Table 4 displays the query used to perform data cleansing. The result of the query is a data set of customer accounts whose contract account meets the business rules.

TABLE IV
DATA CLEANSING STEPS

No	Query	Description
Q1	`CREATE TABLE accounts AS SELECT * from fkkvkp WHERE loevm=' '`	Create a new table that contains unterminated customer account data
Q2	`CREATE TABLE opendfkkop AS SELECT * FROM dfkkop WHERE augst=' 'AND (budat BETWEEN '20200601' AND '20211130')`	Create a new table containing unpaid customer invoice data over the past 18 months
Q3	`CREATE TABLE paiddfkkop AS SELECT * FROM dfkkop WHERE (zznper='202109' OR zznper='202110' OR zznper='202111') AND blart='IP'`	Create a new table that contains customer invoice data that has been paid in the past three months
Q4	`CREATE TABLE acc_results AS SELECT vkont FROM accounts WHERE vkont IN (SELECT vkont FROM opendfkkop) OR vkont IN (SELECT vkont FROM paiddfkkop)`	Set up a new table that contains customer account IDs from FKKVKP table that meet the criteria
Q5	`CREATE TABLE active_acc AS SELECT DISTINCT * FROM acc_results`	Delete duplication on customer account ID data

Once the data cleansing stage is complete, the next step is the validation stage to ensure the cleansing results have followed the business rules from the company side.

B. Validation Phase

Table 5 shows the query used to validate the results of the cleansing stage. The next step is recording the total processing time of the query by both Spark SQL and Oracle SQL. From the recording results, an analysis was done to compare those two tools. This comparison was conducted on queries from the cleansing steps and validation steps. The results are explained in the next section using tables and charts.

TABLE V
VALIDATION STEPS

No	Query	Description
Q1	`SELECT vkont FROM active_acc GROUP BY vkont HAVING COUNT(vkont)>1`	Validate that there is no duplicate data
Q2	`SELECT vkont FROM active_acc WHERE vkont IN (SELECT vkont FROM paiddfkkop WHERE zznper < '202109')`	Check if there are active customer accounts that paid invoices beyond 3 months ago
Q3	`SELECT vkont FROM active_acc WHERE vkont in (SELECT vkont FROM opendfkkop WHERE augst='9' OR budat < '20200601')`	Check if there are active customer accounts that haven't paid invoices more than 18 months ago

III. RESULTS AND DISCUSSION

This section reviews the comparison results of the query processing in the aspect of time performance on both tools. The tables are shown in this section use defined queries from the previous section, and table 6 shows the query processing time on both tools. At this stage, Apache Spark has always been superior to Oracle Database. However, there is a significant difference in Q3, and Oracle's high load can cause this at the time of the experiment conducted.

TABLE VI
COMPARISON ON DATA CLEANSING STEPS

Query	Step	Oracle Database (in second)	Apache Spark (in second)
Q1	Create table accounts	11	8
Q2	Create table opendfkkop	68	48
Q3	Create table acc_result	1848	42
Q4	Create table active_acc	9	5

Fig. 4 shows a comparison graph of Apache Spark and Oracle Database at the cleansing stage. Based on the comparison results at the cleansing stage, it can be concluded that the difference in the processing time of Apache Spark with Oracle Database is 1:18. In Q3, there was a significant difference. A fairly complex query can cause this because it contains two subqueries, which is why Oracle's computational process becomes longer.

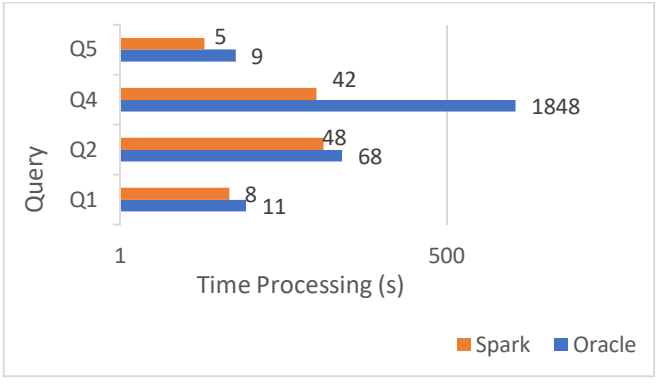

Fig. 4 Comparison graph on cleansing stage

It was discovered that there was a significant difference in Q3. Thus, the experiment is applied to the same query, but the number of data rows is equal. Data of FKKVKP, OPENDFKKOP, and PAIDDFKKOP table is made the same. Table 7 shows the comparison results using Q3 at the cleansing stage. With the same amount of data and the same query, from the comparison results, Spark SQL proved superior to Oracle SQL.

TABLE VII
COMPARISON ON Q3 WITH THE SAME DATA ROWS

Rows Count (FKKVKP, OPENDFKKOP, PAIDDFKKOP)	*Query*	Oracle SQL (s)	Spark SQL (s)
1000	`CREATE TABLE TA_VKONT AS SELECT VKONT FROM FKKVKP WHERE VKONT IN (SELECT VKONT FROM OPENDFKKOP) OR VKONT IN (SELECT VKONT FROM PAIDDFKKOP)`	0.13	0.1
10000		15	0.1
100000		1773	1

Next, Table 8 displays the processing time at the validation stage using Oracle SQL. At this stage, the experiment was conducted five times with various time processing results.

TABLE VIII
VALIDATION STEPS ON ORACLE SQL

No	Query	Experiment (sec)					Mean (sec)
		1	2	3	4	5	
Q1	Check duplicate data	3	3	3	3	3	3
Q2	Check customer accounts that have no open invoice	10	8	7	7	7	7.8
Q3	Check customer accounts that still have an open invoice	15	13	12	12	12	12.8

Table 9 shows the processing time using Spark SQL at the validation stage. This experiment was also done five times with various results. After each experiment is recorded, the next is to calculate the average of each stage.

TABLE IX
VALIDATION STEPS ON SPARK SQL

No	Query	Experiment (sec)					Mean (sec)
		1	2	3	4	5	
Q1	Check duplicate data	2	2	1	1	1	1.4
Q2	Check customer accounts that have no open invoice	15	2	1	1	1	4
Q3	Check customer accounts that still have an open invoice	19	13	12	11	11	13.2

Fig. 5 shows the Apache Spark and Oracle database performance comparison graph at the validation stage. A slightly different amount of time on the third query can be seen. At this stage, it can be concluded that in Q3, Spark performed worse than Oracle. Spark needs time to load the dataset on the first experiment. Once the data is called already, the processing time will be faster. Therefore, the processing time decreased little by little in each iteration.

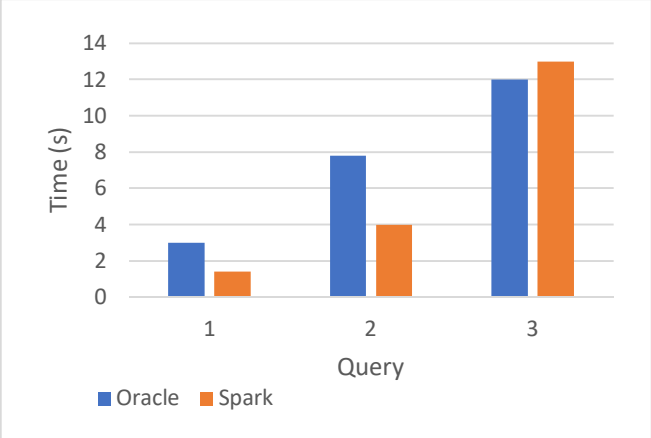

Fig. 5 Comparison graph on validation stage

Based on the comparison results of query processing time at the cleansing and validation stages, it can be concluded that Apache Spark has superior performance. This is because Spark supports parallel processing, where data is formed into an RDD and distributed to each cluster to process the data in a smaller form. Other than that, Spark has an in-memory processing method, so the data calling process is faster. However, data must first be loaded and stored in Spark Warehouse a few times, which causes Spark to perform a little worse in Q3.

Meanwhile, Oracle SQL uses a disk-based data processing method to process data. Data must be invoked first from the disk. This causes data processing time to be slower for RDBMS in general. Although Oracle Database uses more advanced technology specifications, Apache Spark processing remains better. Table 10 shows a comparison of these two tools in several aspects. Both Spark and Oracle have their respective advantages. Therefore, reliable or not, these two tools are determined by the needs of a company is using it.

TABLE X
ORACLE AND SPARK COMPARISON

No	Oracle Database	Apache Spark
1	An RDBMS used to manage relational databases	An engine used to perform data analysis
2	Use SQL-based query	Use SQL-based queries through APIs, namely Spark SQL module
3	Closed-source and paid licenses	Open-source and unpaid licenses
4	Data is stored on disk for processing	Spark uses in-memory processing so that computation process is faster [25]
5	Supports all SQL statements	Supports all SQL statements except UPDATE

When compared to previous research [24], [25], [26], this study uses the same method of quantitative comparative analysis. Previous research has proven that Apache Spark and Oracle Database are superior to their comparison tools. However, there has been no research comparing the performance of these two tools. Therefore, it is expected that

the results of this study can be a source of reference for future research.

IV. Conclusion

This study displays the process of cleaning data from dirty data to obtain the reliability aspect of data quality. Data cleansing is done using Spark SQL and Oracle. It is known that the two tools have different architecture and processing techniques. Therefore, based on the results obtained, it confirmed that the performance of Apache Spark in terms of query processing time is superior to Oracle Database. However, both are not similar tools. Thus, it can be said that Oracle Database is more suitable for use as a data management tool in large sizes. Meanwhile, Apache Spark is more suitable for use as a data analysis tool. Therefore, to do data analytics, data should be first loaded in Apache Spark from Oracle Database and then moved into Spark Warehouse using JDBC connector.

Given the limitations of the material and methodology used in this study, then for future research, it is recommended that a comparison can be made in several more aspects, such as memory usage and CPU usage. Other than that, future research can consider adding query optimization procedures into the testing queries. Thus, the comparison is not only done with similar queries but also with more diverse queries.

References

[1] I. Taleb and M. A. Serhani, "Big Data Pre-Processing: Closing the Data Quality Enforcement Loop," in *IEEE International Congress on Big Data (BigData Congress)*, Honolulu, 2017.

[2] H. A. Sulistyo, T. F. Kusumasari and E. N. Alam, "Implementation of Data Cleansing Null Method for Data Quality Management Dashboard using Pentaho Data Integration," in *International Conference on Information and Communications Technology*, Yogyakarta, 2020.

[3] F. Boufarez, A. B. Salem, M. Rehab and S. Correia, "Similar Data Elimination: MFB Algorithm," *2013 International Conference on Control, Decision and Information Technologies (CoDIT),* p. 289, 2013.

[4] I. Taleb, M. A. Serhani and R. Dssouli, "Big Data Quality Assessment Model for Unstructured Data," in *International Conference on Innovations in Information Technology (IIT)*, Al Ain, 2018.

[5] A. Juneja and N. N. Das, "Big Data Quality Framework: Pre-Processing Data in Weather Monitoring Application," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con)*, Faridabad, 2019.

[6] T. Hongxun, W. Honggang and Z. Kun, "Data Quality Assessment for On-line Monitoring and Measuring System of Power Quality Based on Big Data and Data Provenance Theory," in *3rd IEEE International Conference on Cloud Computing and Big Data Analysis*, Chengdu, 2018.

[7] S. Loetpipatwanich and P. Vichitthamaros, "Sakdas: A Python Package for Data Profiling and Data Quality Auditing," in *1st International Conference on Big Data Analytics and Practices*, Bangkok, 2020.

[8] S. R. Amethyst, T. F. Kusumasari and M. A. Hasibuan, "Data Pattern Single Column Analysis for Data Profiling using an Open Source Platform," in *IOP Conference Series Materials Science and Engineering*, 2018.

[9] S. Juddoo and C. George, "A Qualitative Assessment of Machine Learning Support for Detecting Data Completeness and Accuracy Issues to Improve Data Analytics in Big Data for the Healthcare Industry," in *3rd International Conference on Emerging Trends in Electrical, Electronic, and Communications Engineering*, Balaclava, 2020.

[10] F. Haneem, N. Kama, R. Ali and S. Basri, "Resolving data duplication, inaccuracy and inconsistency issues using Master Data Management," 2017.

[11] E. Rahm and H. H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Bulletin of the Technical Committee on Data Engineering,* vol. 23, pp. 3-13, 2000.

[12] T. F. Kusumasari and Fitria, "Data Profiling for Data Quality Improvement with Openrefine," in *International Conference on Information Technology Systems and Innovation* , Bali, 2016.

[13] V. Kumar and C. Khosla, "Data Cleaning-A Thorough Analysis and Survey on Unstructured Data," in *8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, 2018.

[14] I. F. Ilyas and X. Chu, Data Cleaning, Association for Computing Machinery, 2019.

[15] F. Ridzuan and W. M. N. W. Zainon, "A Review on Data Cleansing Methods for Big Data," in *The Fifth Information Systems International Conference 2019*, 2019.

[16] J. Yin, J. Zhang, D. Li, T. Wang and K. Jing, "Big data cleaning model of smart grid based on Tensor Tucker decomposition," in *International Conference on Big Data & Artificial Intelligence & Software Engineering* , Bangkok, 2020.

[17] A. Wakde, P. Shende, S. Waydande, S. Uttarwar and G. Deshmukh, "Comparative Analysis of Hadoop Tools and Spark Technology," in *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, 2018.

[18] X. Li and W. Zhou, "Performance Comparison of Hive, Impala and Spark SQL," in *International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, 2015.

[19] G. Gousios, "Big Data Software Analytics with Apache Spark," in *International Conference on Software Engineering: Companion Proceedings*, 2018.

[20] H. Müller and J.-C. Freytag, "Problems, methods, and challenges in comprehensive data cleansing," Humboldt University Berlin, 2003.

[21] C. Li, Y. Hou and Z. Yu, "Research on data cleaning technology based on instance level," *Journal of Physics,* pp. 1-4, 2019.

[22] M. Smallcombe, "Top Data Cleansing Tools for 2022," integrate.io, 12 January 2021. [Online]. Available: https://www.integrate.io/blog/top-10-data-cleansing-tools/. [Accessed 6 January 2022].

[23] F. Chen and L. Jiang, "A parallel algorithm for data cleansing in incomplete information systems using MapReduce," in *International Conference on Computational Intelligence and Security*, Kunming, 2014.

[24] S. Padhy and G. M. M. Kumaran, "A Quantitative Performance Analysis between Mongodb and Oracle NoSQL," in *International Conference on Computing for Sustainable Global Development*, New Delhi, 2019.

[25] A. V. Hazarika, G. J. S. R. Ram and E. Jain, "Performance Comparision of Hadoop and Spark Engine," in *International conference on I-SMAC*, Palladam, 2017.

[26] R. Poljak, P. Poscic and D. Jaksic, "Comparative Analysis of the Selected Relational Database Management Systems," in *International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, 2017.

[27] J. Powers, "Apache Spark Performance Compared to a Traditional Relational Database using Open Source Big Data Health Software," *Project Paper for CSE8803 Big Data Analytics for Health Care,* pp. 1-5, 24 April 2016.