Distributed Scalable Association Rule Mining over Covid-19 Data

Mahtab Shahin[1](✉), Wissem Inoubli[2], Syed Attique Shah[3], Sadok Ben Yahia[2], and Dirk Draheim[1]

[1] Information Systems Group, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia
{mahtab.shahin,dirk.draheim}@taltech.ee

[2] Software Science Department, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia
{wissem.inoubli,sadok.ben}@taltech.ee

[3] Institute of Computer Science, University of Tartu, Tartu, Estonia
syed.shah@ut.ee

Abstract. The worldwide Covid-19 widespread in 2020 has turned into a phenomenon that has shaken human life significantly. It is widely recognized that taking faster measurements is crucial for monitoring and preventing the further spread of COVID-19. The advent of distributive computing frameworks provides one efficient solution for the issue. One method uses non-clinical techniques, such as data mining tools and other artificial intelligence technologies. Spark is a widely used framework and accepted by the big data community. This research used a cross-country Covid-19 dataset to assess the performance of the Apriori and FP-growth through different components of Spark (different numbers of cores and transactions). This involves a scheme for classification and prediction by recognizing the associated rules relating to Coronavirus. This research aims to understand the difference between FP-growth and Apriori and find the ideal parameters of Spark that can improve the performance by adding nodes.

Keywords: Association rule mining · Big data · FP-growth · Spark · Apriori · Machine learning

1 Introduction

Coronavirus disease (COVID-19) belongs to a larger family of Coronaviruses (CoV). This severe illness can be deadly as it assaults our respiratory cells and causes an immune response that targets those infected cells, damages lung tissue, and might finally shut off our supply of Oxygen by clogging our airways [4]. Countries worldwide have prioritized the early and automated diagnosis of this disease to assign patients to quarantine and take further steps promptly. In some severe cases, diagnosis has taken place in specialized hospitals to more efficiently

© Springer Nature Switzerland AG 2021

track disease transmission. Diagnosis is such a rapid procedure; therefore, the high expenses of further investigations have caused financial issues harming both states and patients, especially in areas where private health systems or economic issues can restrict one's access to medical care.

Classification, grouping, regression, and correlation are all aspects of data mining [21]. Data mining provides information about previously accurate independent itemsets and their relationship in extensive databases. Frequent Itemset Mining is the process of extracting frequent itemsets from transaction databases. It is crucial to look at association rules commonly utilized in real-world applications, including web data analysis, consumer behavior research, cross-marketing, catalog design, and medical records. In addition, Association Rule Mining (ARM) is involved in biological sciences, and researches illness detection and accurate classification prediction [31]. At its most basic level, the ARM entails analyzing patterns in data, or correlation, within a dataset using data mining tools. Every if-then association, also known as association rules, is defined by If-then statements that illustrate the potential of connections between itemsets in large databases of various sorts [32]. The support and confidence parameters are used to discover links between unrelated datasets or another data source, and ARM is created by looking for recurring data patterns. Support appears to reflect the regularity with which relationships occur in the database, whereas confidence indicates how often these associations have shown to be accurate [17,18,23,24]. All itemsets that fulfill such minimum support are generated for a given dataset. Within the second step, every frequent itemset is employed to develop all potential rules from the dataset; and rules that don't satisfy specified minimum confidence are removed. The main step of association rule mining is in distinctive frequent itemsets. Many ARM algorithms are presently in use: three typical classic representatives are Apriori [2], FP-growth [7], and Eclat [8].

This paper provides a design that supported Spark and association rule mining algorithms to seek an attention-grabbing relationship between Covid-19 data set. The findings would be gainful for patients, doctors, politics, and decision-makers in health informatics. This research addresses numerous contributions to the literature:

– It shows that applying an integrative k-NN/weighted k-NN algorithms with association rule mining improves prediction efficiency.
– It shows that the weighted k-NN has the highest accuracy compared to kNN for chronic disease data.
– It finds the ideal parameters that have positive effects on Spark jobs.
– It compares FP-growth and Apriori for the performance difference and how parameter tuning affects the results.

The remainder of the paper is organized as follows. In Sect. 2, we review the related works in this field. In Sect. 3, we explained the used algorithms briefly. In Sect. 4, we describe the details of the methodology, dataset, and pre-processing part. In Sect. 5, we provide the experimental results. Finally, In Sect. 6, we conclude the paper and present possible directions for future works.

2 Scrutiny of Related Work

One of the classical and well-known techniques of data mining is association rule mining [17]. Data mining determines a method to find out the relevant and gainful patterns in data [3]. This section will summarize prior works in the context of data mining techniques and association rule mining algorithms for finding frequent itemsets. Moreover, we will explain the tools that were used in this regard.

Kate and Nadig [15] proposed prediction models for breast cancer survival using the SEER dataset and machine learning approaches. They applied three different machine learning methods (naive Bayes, logistic regression, and decision tree) and discovered that the performance of the models varied greatly whenever evaluated independently at different phases. Soltani Sarvestani et al. [1] examined various research on the usage of other neural networks for accurate clinical detection of breast cancer in the largest and most active Hospital in South Iran; The idea was first assessed using publicly available statistics from throughout the world. They applied several neural network structures, and they functioned well. The PNN is the best-suited neural network model for categorizing WBCD and NHBCD data according to the overall results. This research also suggests that statistical neural networks can be utilized to aid doctors in breast cancer detection. Shukla et al. [25] proposed an unsupervised data mining creating patient cohort clusters. They applied a large dataset from the SEER program to recognize patterns associated with the survivability of breast cancer patients. These clusters, with associated patterns, were used to train the multilayer perceptron (MLP) model for enhanced patient survivability analysis. Examination of variable values in each cohort gives better insights into the survivability of a special subgroup of breast cancer patients. Wu and Zhou [27] developed two improved SVM methods to identify malignant cancer samples: support vector machine-recursive feature eliminate and support vector machine principal component analysis (SVM-PCA). Hinselmann, Schiller, Cytology, and Biopsy are four target variables that reflect the cervical cancer data. The three SVM-based methods diagnosed and categorized all four targets. They performed a comparison between these three approaches and compared the risk factor ranking result to the ground reality. The SVM-PCA technique is proven to be better than the others. Qiu et al. [20] proposed YAFIM (Yet Another Frequent Itemset Mining) and used it on real-world medical applications to discover the relationships in medicine. They concluded that the proposed method achieved 18 speedups for different benchmarks on average compared with the algorithms executed with MapReduce. It outperforms the MapReduce method about 25 times. To problem-solving of scanning the dataset in each iteration, Kumar Sethi and Ramesh [22] introduced Hybrid Frequent Itemset Mining (HFIM), which employes the vertical layout. The suggested algorithm was implemented over the Spark framework and comprised the concept of resilient distributed datasets to display in-memory processing to optimize the running time of operation. Their results showed that the HFIM performs better in terms of running time and memory consumption. Li and Sheu [19] proposed a divide-and-conquer-based scalable, highly parallelizable association rule mining heuristic (the SARL heuristic) that may reduce both time complexity and memory consumption while obtaining approximation results that are near to correct results. Comparative studies demonstrate that the suggested heuristic method outperforms algorithms by a substantial margin.

3 Preliminaries

A brief description of the algorithms used in the current study has been provided in the following.

3.1 Apriori

Agrawal et al. in 1993 proposed the AIS [2] as the first algorithm to generate all the frequent itemsets. Soon after, the developed version of AIS as the name of Apriori was introduced by Agrawal et al. Initially, association rule mining was utilized for market and sales data, where the function was to discover all the rules that would predict occurred items. This approach follows two steps [3]:

- In the *Join step*, calculate the union of two frequent itemsets of size n, assume taken A_n and B_n, which have a first $n-1$ element in common. $J_{n+1} = A_n \cup B_n$.
- In the *Prune step*, checked whether all the itemset of size n in j_{n+1} is frequent or not, and pruned those rules that do not satisfy the given condition (minimum support, confidence, and lift)

3.2 FP-Growth

Jiawei Han first introduced FP-growth in 2006 [11], where FP stands for frequent patterns. The strategy of FP-growth is based on the strategy of divide and conquer. Two scans have to do on the dataset. First, During the first scan of a database, find support for each item, and calculate a list of distributed frequent items in descending order (F-List). Second, it compresses the dataset into an FP-tree [16]. By using these steps, we can make FP-tree so that common prefixes can be provided.

3.3 k-Nearest Neighbours

According to Bank et al. [6], the general one percent of the data is futile; about one to five percent is manageable. Nevertheless, handling five to fifteen percent of missing data needs some advanced method. More than fifteen percent of missing data may significantly impact any characteristic of the data set. Missing value imputation techniques replace missing values from rows or specific classes with estimated ones, such as mean or mode values. The estimated values rely on various algorithms that return the outcome. Generally, missing values imputation often generates more effective results compared to other methods. kNN method was first proposed by Fix and Hodges [10] in 1951 and later developed by Thomas Cover [9]. It is one of the well-known imputation techniques for its ease of execution and provides fair output results. The principle of kNN is to fill missing values of the dataset according to different values of given k closest to missing items; applying distance function such as Euclidean distance function, evaluate the closeness or similarity between target instance and other instances in the data set. Then chose the top k closest instances as a candidate and determined weighted values as a replacement. The appealing advantages of this method including are:

– Appropriate for both quantitative and qualitative data.
– Avoid time consumption and computational cost, as it can make a predictive model for imputation.

3.4 Apache Spark

Apache Spark [14,26] is known as a unified framework to analyze distributed big data processing. It was originally developed in 2009 at UC Berkeley University. The popularity of Spark is its ability to in-memory calculations that enable it to make faster 100 times compared to MapReduce. Apache Spark supports four basic libraries for machine learning, associated information mining, together with SparkSQL [5], Spark Streaming, Spark MLLib [30], and GraphX [28]. Spark deployment can be in three modes: standard mode, Mesos, and Hadoop Yarn. The principle of Spark is Resilient Distributed Datasets (RDDs). An RDD is a speeded immutable set of objects across a Spark cluster. According to master/slave architecture spark cluster contains of three main components [12,13]:

– Driver Program: this component denotes the slave node in a Spark cluster. It maintains an object called Spark Context that manages running applications.
– Cluster Manager: this component can arrange the application's workflow since approved by Driver Program to workers. It also manages and controls every resource in the cluster and delivers its state to the Driver Program.
– Worker Nodes: every Worker Node denotes a container of one operation through a Spark program execution.

4 Methodology

4.1 Hardware and Software Configuration

The experiment of Hadoop and Spark were conducted on a high-performance computer by Python 3.7. It consisted of 11 nodes, and every single node was deployed with the same physical environment. Both Spark and Hadoop were configured on JDK version 8 and run the jobs on YARN. Also, HDFS is used to save intermediate data. The versions of Spark and Hadoop were 3.0.0 and 3.1.0, respectively. The details of nodes are shown in Table 1.

Table 1. System configuration.

Node type	Processor	Memory	OS	Docker version
Master	8	64	ubuntu 18.04	20.10.5
Slave	8	8		

In order to ssh the command line of the master node, we utilized PuTTY and access the HPC by its IP address.

4.2 Dataset Description

The Covid-19 data used in this experiment were taken from [29][1] – see Table 2 for details of the dataset.

Table 2. Properties of the used Covid-19 data set.

Size	# of Transaction	Time period
5.9 GB	3,048,576	December 2019 – January 2020

Among 31 attributes of this data set, we have selected 10 attributes to be included in our analysis, see Table 3, also compare with Fig. 1.

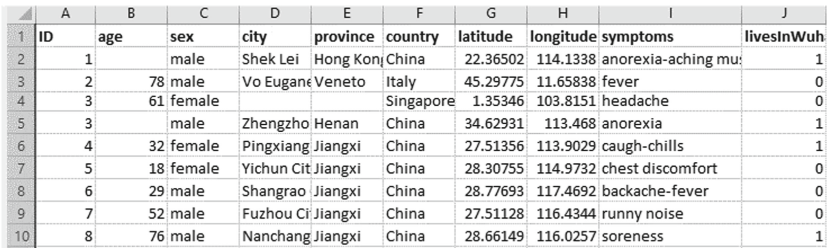
4.3 Data Pre-processing

Data preprocessing is one of the essential steps in the data mining process and is known as converting raw data into accurate data [2]. The main stages of data preprocessing are integration, cleaning, reduction, transformation, and discretization of the dataset. A preprocessing phase is developed with two goals, to optimize and speed up the process: (a) finding all sensitive transactions and determining weak rules; and (b) indexing different types of patterns affected by the sensitive transactions and the items. First, specify all sensitive transactions by scanning the whole database, by accomplishing the first purpose. Then, we removed the duplicated transactions to specify only everyday transactions instead of considering all database transactions. This process helps to reduce the size of the solutions and increase the speed of runtime. The second objective is to reduce database scanning by generating different index lists for sensitive transactions and items. Each transaction modification causes three different side effects: lost rule, hiding failure, and new rule.

[1] https://github.com/beoutbreakprepared/nCoV2019.

Table 3. Selected attributes

Attribute	Description
ID	Identify document for each reported case
Age	Age of the reported case
Gender	Male/female
City	Name of the reported city
Province	Name of the reported province
Country	Name of the reported country
Latitude	The latitude of the specific location
Longitude	The longitude of the specific location
Symptoms	List of reported symptoms in the case' description
Lives in Wuhan	0 the person does not live in Wuhan
	1 the person lives in Wuhan



	A	B	C	D	E	F	G	H	I	J
1	ID	age	sex	city	province	country	latitude	longitude	symptoms	livesInWuh
2	1		male	Shek Lei	Hong Kon	China	22.36502	114.1338	anorexia-aching mu	1
3	2	78	male	Vo Eugane	Veneto	Italy	45.29775	11.65838	fever	0
4	3	61	female			Singapore	1.35346	103.8151	headache	0
5	3		male	Zhengzho	Henan	China	34.62931	113.468	anorexia	1
6	4	32	female	Pingxiang	Jiangxi	China	27.51356	113.9029	caugh-chills	1
7	5	18	female	Yichun Cit	Jiangxi	China	28.30755	114.9732	chest discomfort	0
8	6	29	male	Shangrao	Jiangxi	China	28.77693	117.4692	backache-fever	0
9	7	52	male	Fuzhou Ci	Jiangxi	China	27.51128	116.4344	runny noise	0
10	8	76	male	Nanchang	Jiangxi	China	28.66149	116.0257	soreness	1

Fig. 1. First ten rows of the dataset

Filling Missing Values with k-Nearest Neighbours. The imputation of the missing values process comprises two main steps: The first step selects the set of attributes to the features with missing values as the target. Let the Covid-19 dataset be represented as a patient information expression matrix C with m columns and n rows corresponding to transactions and attributes, respectively. To impute the missing values of transaction X_c, c $\in \{1, \ldots,$ n$\}$ and attributes X_c, $i \in \{1, \ldots,$ m$\}$, it is to find k other transactions, each with a known value for attribute i and its features being the most similar to that of items.

$$d_{ij} = dist(x_i, x_j) = \sqrt{\sum_{p=1}^{n} (d_{ip} - d_{j_p})^2} \tag{1}$$

Where $dist$ $(x_i,\ x_j)$ denotes the Euclidean distance between transaction x_i and x_j. n is the number of items, and x_{ip} is the p^{th} of transaction x_i. The second step includes predicting the missing value using the observed values belonging to the selected item of transactions. At this stage, an average of values in experiment i from the k closest transactions is then used to estimate the missing value in transactions x_i. Can determine the estimated $\widetilde{x_{ip}}$ value of the missing x_{ip} as below:

$$\widetilde{x_{ip}} = \frac{\sum_{\forall x_a \in N_g} x_{ai}}{k} \tag{2}$$

In the above equations, x_i is the set of k nearest neighbors of transaction x_i. Moreover, in the weighted variation, the contribution of each transaction $x_a \in N_g$ is weighted by the similarity of its explanation to that of the transaction. Accordingly, higher weights are defined as a more similar transaction. A weighted average of values from k nearest transactions is then used to assess the missing value in the target transaction. This weight computation is as follow:

$$\widetilde{x_{ip}} = \sum_{\forall x_a \in N_g} x_{ai} w_i \tag{3}$$

where;

$$w_i = \frac{\frac{1}{d_i}}{\sum_k^{i=1} \frac{1}{d_i'}} \tag{4}$$

Table 4. Split-Validation results for the pre-processing

K#	KNN	WKNN
1	80.1	89.2
2	75	78.2
3	79.6	86.6
4	67.3	78.4
5	66.2	97.9
6	59.9	89.2
7	60.4	83.2

5 Results and Discussion

To implement the framework, we applied FP-growth and Apriori algorithms of the MLlib machine learning library. Then, the deployment and execution of the recommended system are done over a distributed computing environment formed of a different number of clusters, nodes, and transactions by Spark resources management. This section provides how to evaluate the performance of the Spark with three different experiments. We utilized the running time to present the efficiency because it can show the difference and efficiency directly (Fig. 2 and Table 4).

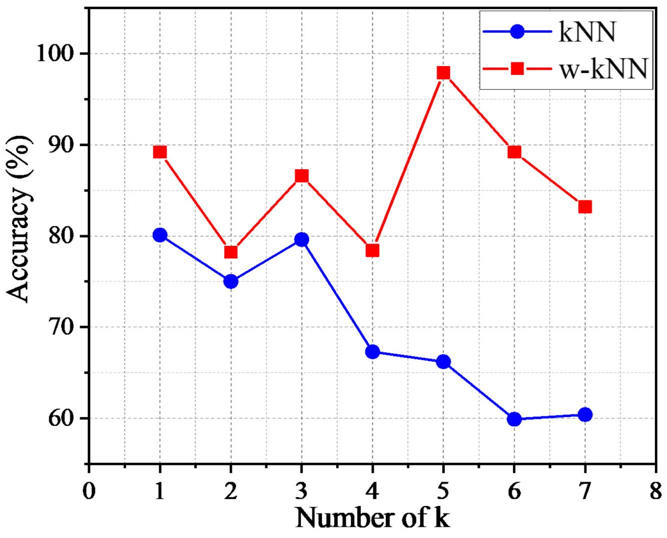


Fig. 2. Split-Validation results of kNN and wkNN

5.1 Core Utilization

This parameter determines the parallel computing ability for each executor. The executer-core is employed for configuring the number of CPU cores for each executor. As one CPU core can execute many tasks simultaneously, the more CPU cores assigned to the executors, the faster the Spark job. Experiments have been done for both algorithms in the same configuration to analyze the performance of Apriori and FP-growth. As shown in Fig. 3, we have applied the experiments in different numbers of core, from one to eight.

5.2 Node Utilization

Spark splits the work into multiple execute tasks on worker nodes. Thus Spark processors data in less time. Figure 4 shows that the running time strongly decreases as far as the number of nodes increases. Besides, we can find that the FP-growth curve is still sharper for all nodes than Apriori, which means FP-growth is more efficient than Apriori. The average running time of FP-growth and Apriori can be shown from Table 5.

5.3 Number of Transactions

We determined a comparative analysis between running time on FP-growth and Apriori to show the scalability. For that, we increased the number of transactions to take a sufficient database size. Later, we measured the speed of the

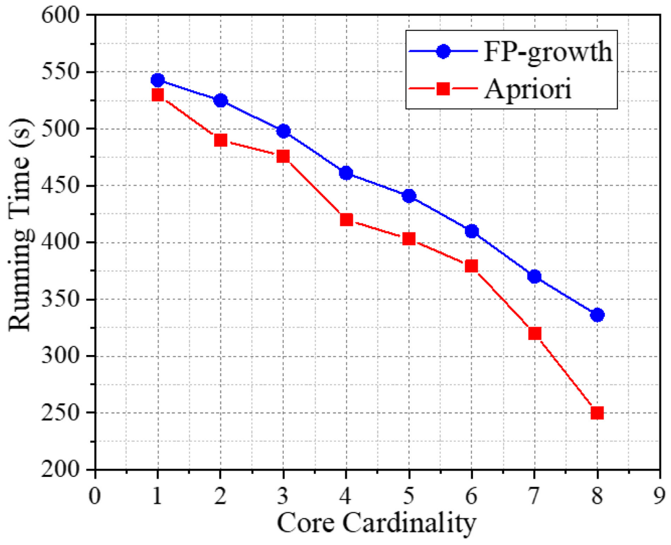


Fig. 3. Results of the number of cores

Table 5. Running time of Apriori and Fp-growth based on the variation of the number of nodes

Node #	Algorithms (s)	
	Apriori	FP-growth
1	553	2.1
3	437	11.6
5	297	281
7	238	230
9	119	116
11	59	41

FP-growth algorithm using the MLlib library compared to the Apriori in the same environment. The results are outlined in Fig. 5, which represents the line chart of execution times of different algorithms. As shown in the line chart, running the association rules with FP-growth is faster than Apriori. For example, it takes about 600 s to process 3 million transactions when Apriori takes 650 s. Furthermore, the FP-growth algorithm of Spark Mllib accomplishes good scalability because of the distributed computing on cluster nodes. As a result of the above analysis, FP-growth provided the most suitable environment to implement the Covid-19 dataset.

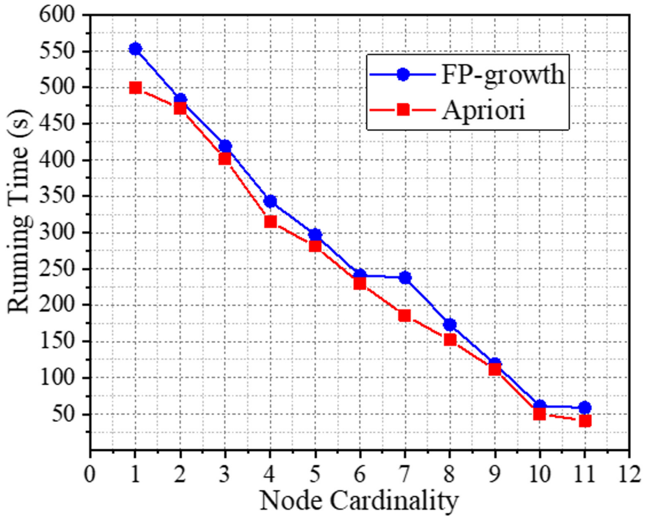


Fig. 4. Results of the number of nodes

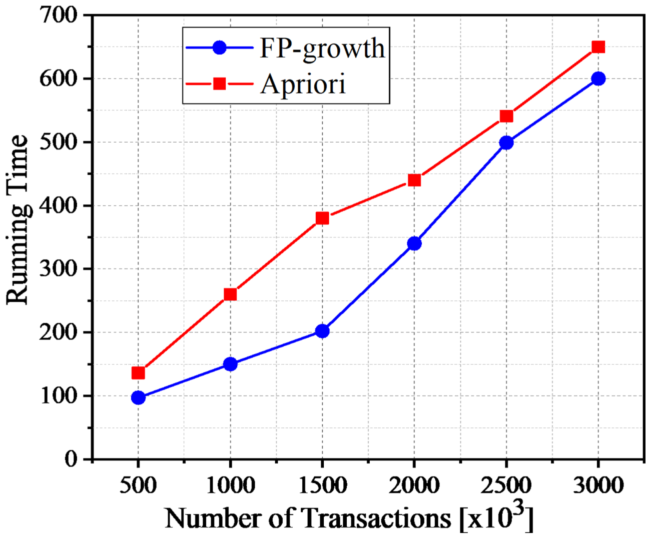


Fig. 5. The optimal performance for Apriori and FP-growth with different input splits

6 Conclusion

This paper aims to design a distributed framework for finding frequent itemsets of the Covid-19 dataset. We applied the Spark framework to expedite the parallel processing of data and decrease the calculation cost. Overall, the experiment results show that the performance of FP-growth is always superior to the Apriori algorithm. It is because Apriori requires scans of the database multiple times for generating candidate sets to generate frequent items; in contrast, FP-growth scans of the dataset only twice. Moreover, Apriori needs more time and large memory space. Due to the minimum support threshold reduction, the number and exponentially increase the length of frequent itemsets.

The most relevant future work that can stem from the research is discovering association rules from the Covid-19 data set. Furthermore, we want to expand this approach and extract the symptom patterns from the dataset. Hence, it is worth investigating the quality of the results produced by Apriori and FP-growth.

Acknowledgements. This work has been conducted in the project "ICT programme" which was supported by the European Union through the European Social Fund.

References

1. Abdelghani, B., Guven, E.: Predicting breast cancer survivability using data mining techniques. In: SIAM International Conference on Data Mining (2006)
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
3. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of 20th International Conference on Very Large Data Bases, VLDB, vol. 1215, pp. 487–499. Citeseer (1994)
4. Anwar, H., Khan, Q.U.: Pathology and therapeutics of COVID-19: a review. Int. J. Med. Stud. **8**(2), 113–120 (2020)
5. Armbrust, M., et al.: Spark SQL: Relational data processing in spark. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1383–1394 (2015)
6. Banks, D., House, L., McMorris, F.R., Arabie, P., Gaul, W.A.: Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, 15–18 July 2004. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17103-1
7. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: a case study. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 254–260 (1999)
8. Chen, Y., Li, F., Fan, J.: Mining association rules in big data with NGEP. Clust. Comput. **18**(2), 577–585 (2015)
9. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
10. Fix, E., Hodges, J.L.: Discriminatory analysis. nonparametric discrimination: consistency properties. Int. Stat. Rev./Revue Int. Stat. **57**(3), 238–247 (1989)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. ACM SIGMOD Rec. **29**(2), 1–12 (2000)

12. Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., Nguifo, E.: A comparative study on streaming frameworks for big data. In: VLDB 2018–44th International Conference on Very Large Data Bases: Workshop LADaS-Latin American Data Science, pp. 1–8 (2018)
13. Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., Nguifo, E.M.: An experimental survey on big data frameworks. Futur. Gener. Comput. Syst. **86**, 546–564 (2018)
14. Inoubli, W., Aridhi, S., Mezni, H., Mondher, M., Nguifo, E.: A distributed algorithm for large-scale graph clustering (2019)
15. Kate, R.J., Nadig, R.: Stage-specific predictive models for breast cancer survivability. Int. J. Med. Inf. **97**, 304–311 (2017)
16. Kaur, G., Aggarwal, S.: Performance analysis of association rule mining algorithms. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(8), 856–58 (2013)
17. Kaushik, M., Sharma, R., Peious, S.A., Shahin, M., Ben Yahia, S., Draheim, D.: On the potential of numerical association rule mining. In: Dang, T.K., Küng, J., Takizawa, M., Chung, T.M. (eds.) FDSE 2020. CCIS, vol. 1306, pp. 3–20. Springer, Singapore (2020). https://doi.org/10.1007/978-981-33-4370-2_1
18. Kaushik, M., Sharma, R., Peious, S.A., Shahin, M., Yahia, S.B., Draheim, D.: A systematic assessment of numerical association rule mining methods. SN Comput. Sci. **2**(5), 1–13 (2021)
19. Li, H., Sheu, P.C.-Y.: A scalable association rule learning heuristic for large datasets. J. Big Data **8**(1), 1–32 (2021). https://doi.org/10.1186/s40537-021-00473-3
20. Qiu, H., Gu, R., Yuan, C., Huang, Y.: YAFIM: a parallel frequent itemset mining algorithm with spark. In: 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, pp. 1664–1671. IEEE (2014)
21. Rasheed, J., et al.: A survey on artificial intelligence approaches in supporting frontline workers and decision makers for the COVID-19 pandemic. Chaos Solit. Fractals **141**, 110337 (2020). https://doi.org/10.1016/j.chaos.2020.110337. https://www.sciencedirect.com/science/article/pii/S0960077920307323
22. Senthilkumar, A., Hari Prasad, D.: An efficient FP-growth based association rule mining algorithm using hadoop MapReduce. Indian J. Sci. Technol. **13**(34), 3561–3571 (2020)
23. Shahin, M., et al.: Big data analytic in association rule mining: A systematic literature review. In: Proceedings of the International Conference on Big Data Engineering and Technology (2021). (in press)
24. Shahin, M., et al.: Cluster-based association rule mining for an intersection accident dataset. In: Proceedings of the IEEE International Conference on Computing, Electronic and Electrical Engineering (ICECUBE) (2021)
25. Shukla, N., Hagenbuchner, M., Win, K.T., Yang, J.: Breast cancer data analysis for survivability studies and prediction. Comput. Methods Program. Biomed. **155**, 199–208 (2018)
26. Spark, A.: Unified analytics engine for big data (2018). Accessed 5 Feb 2019
27. Wu, W., Zhou, H.: Data-driven diagnosis of cervical cancer with support vector machine-based approaches. IEEE Access **5**, 25189–25195 (2017)
28. Xin, R.S., Gonzalez, J.E., Franklin, M.J., Stoica, I.: GraphX: a resilient distributed graph system on spark. In: First International Workshop on Graph Data Management Experiences and Systems, pp. 1–6 (2013)
29. Xu, B., et al.: Epidemiological data from the COVID-19 outbreak, real-time case information. Sci. Data **7**(1), 1–6 (2020)
30. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I., et al.: Spark: cluster computing with working sets. HotCloud **10**(10–10), 95 (2010)

31. Zaki, M.J.: Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. **12**(3), 372–390 (2000)
32. Zhang, S., Webb, G.I.: Further pruning for efficient association rule discovery. In: Stumptner, M., Corbett, D., Brooks, M. (eds.) AI 2001. LNCS (LNAI), vol. 2256, pp. 605–618. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45656-2_52