Journal of Big Data

RESEARCH Open Access

Minimum threshold determination method based on dataset characteristics in association rule mining

Erna Hikmawati*, Nur Ulfa Maulidevi and Kridanto Surendro

*Correspondence:
ernahikma21@students.itb.ac.id
School of Electrical Engineering and Informatics, Institut Teknologi, Bandung, Indonesia

Abstract

Association rule mining is a technique that is widely used in data mining. This technique is used to identify interesting relationships between sets of items in a dataset and predict associative behavior for new data. Before the rule is formed, it must be determined in advance which items will be involved or called the frequent itemset. In this step, a threshold is used to eliminate items excluded in the frequent itemset which is also known as the minimum support. Furthermore, the threshold provides an important role in determining the number of rules generated. However, setting the wrong threshold leads to the failure of the association rule mining to obtain rules. Currently, user determines the minimum support value randomly. This leads to a challenge that becomes worse for a user that is ignorant of the dataset characteristics. It causes a lot of memory and time consumption. This is because the rule formation process is repeated until it finds the desired number of rules. The value of minimum support in the adaptive support model is determined based on the average and total number of items in each transaction, as well as their support values. Furthermore, the proposed method also uses certain criteria as thresholds, therefore, the resulting rules are in accordance with user needs. The minimum support value in the proposed method is obtained from the average utility value divided by the total existing transactions. Experiments were carried out on 8 specific datasets to determine the association rules using different dataset characteristics. The trial of the proposed adaptive support method uses 2 basic algorithms in the association rule, namely Apriori and Fpgrowth. The test is carried out repeatedly to determine the highest and lowest minimum support values. The result showed that 6 out of 8 datasets produced minimum and maximum support values for the apriori and fpgrowth algorithms. This means that the value of the proposed adaptive support has the ability to generate a rule when viewed from the quality as adaptive support produces at a lift ratio value of > 1. The dataset characteristics obtained from the experimental results can be used as a factor to determine the minimum threshold value.

Keywords: Minimum threshold, Adaptive rule, Association rule

Introduction

The increase in data usage led to large quantities of data growth which were accessible from various locations everywhere at all times [1]. Data availability is a huge asset for an organization because it contains a lot of useful information. Therefore, Data Mining is a way to extract this useful information and patterns [1–6]. Furthermore, the Association rule is one of the most basic study topics widely used in Data Mining [1–5, 7–11]. It is an important data analysis method that discovers and identifies interesting relationships between sets of items in a dataset and predicts association relationships for new data [7, 8, 12].

The basic concept of the association rule is to generate rules based on items with frequent occurrence in a transaction. This includes two main processes, namely the determination of frequent itemset and the process of forming rules. A frequent itemset is a collection of items that have a higher frequency of occurrence compared to the threshold value specified in the transaction. This value is also known as the minimum support. The real-time applications of the association rule have several challenges ranging from the presence of very large, multiple, and heterogeneous data sources to difficulty in the determination the value of minimum support [7]. Furthermore, the first step in the association rule is to determine the minimum support value [13, 14]. Currently, this value is determined by the user. Currently, this value is determined by the user. This is done because the user is considered to know the most about the dataset. In addition, the user can set limits to what extent and how much of the desired output. In fact, not a few users have difficulty in determining the value of minimum support. Another difficulty is that the user is ignorant of the dataset characteristics to be searched during this process [3, 6, 11, 15–18].

The minimum support value has an important role in determining the value of several rules generated [3]. However, determining a wrong value leads to the failure of the association rule to obtain the required rule [19]. Furthermore, determining a value that is too low leads to the involvement of too many items in the rule formation process. Conversely, when a value that is too high is obtained, fewer items are involved which leads to the loss of a lot of information. The minimum support values when compared in terms of two factors namely, time and memory showed that a value that is too low will require more of these two factors when compared to a higher value [6, 11, 20]. Currently, the method used to determine the value of minimum support is based on the intuitiveness of the user. The association rule process is repeated through the change of the value of minimum support when the rule obtained does not match. However, this does not guarantee that a rule will be generated with the input minimum support value.

Decision-makers are required to make the right strategic choices which lead to the adjustment of current conditions which are full of volatility, uncertainty, complexity, and ambiguity. This is known as the VUCA (Volatility, Uncertainty, Complexity, Ambiguity) world [21]. Furthermore, the current conditions also demand the adaptive transformation of the association rules to meet the needs of the users. Decision-makers have different criteria which are used to obtain the information they need. Therefore the rule formation process should pay attention to the criteria desired by the users to obtain adaptive Rules [22].

Furthermore, to meet these needs, the determination of frequent itemset should not only be based on the frequency of occurrence of an item but also involve another criterion called item utility [10, 23–25]. Therefore, the resulting rule will be adaptive according to the criteria desired by the user.

This study which proposes a method to determine the value of minimum support based on the characteristics of the dataset and other criteria can also be used as assessments in the process of forming the rule. In the proposed method, the user is not required to determine the minimum support value at the beginning, because the minimum support value is calculated automatically based on the characteristics of the dataset. In addition, the determination of the minimum threshold value is not only based on the frequency of occurrence of items but involves other criteria that are factors in the rule formation process. With this method the rule formation process becomes more adaptive according to user needs. The proposed method is only up to the process of determining the minimum threshold for selecting frequent itemset. After the frequent itemset is formed, the rule formation process can use various existing algorithms. With this proposed method, the user is not bothered with determining the value of minimum support and must repeat the rule formation process many times because the selection of the value of minimum support is not appropriate. The main contributions of this study include:

1. A new method for determining the value of minimum support based on the characteristics of the dataset, so that the user does not need to determine the minimum support value at the beginning. This method will automatically read and calculate each item as well as its frequency in the dataset and propose the appropriate minimum threshold value.
2. The generation of a minimum support value-based not only on the frequency of occurrence of items but also based on certain criteria affecting the rule formation process such as price, profit, or item usability value. By involving certain criteria, the rule formation process becomes more adaptive according to the needs and desires of the user. Users can focus on rules based on items that have high utility.

This study is divided into several parts. Part I contains the introduction and background of this research. The related work will be described in part II, proposed methods will be explained in part III, experimental results and discussion will be explained in part IV, conclusions will be explained in part V.

Related work

Association rule

The concept of association rule mining was initially introduced in a research paper by Agrawal [26, 27]. This study developed a focus on different topics ranging from high utility itemset [23, 25, 28–31], Top-K [5, 11, 32], Skyline [6, 10, 33], Multicriteria [34–36] and Meta association rules [37–39]. Furthermore, the following is a formal definition of the main concept of association rule mining: $I = \{i1, i2, \ldots, im\}$ is a set of items and D is a set of transactions T, where a set of transactions T is also a set of items, therefore, $T \subseteq I$. Furthermore, provided that A is a set of items, Transaction T is said to contain A if

and only if A ⊆ T. Association rules are from A B, where A ⊆ I, B ⊆ I, and A ∩ B = Ø. In addition, rule A B has support in transaction set D provided that s% of transactions in D contain A ∪ B [4, 22, 40]. This support is shown by the occurrence frequency of items which is calculated by observing the ratio between the frequency of transactions containing itemset A divided by the total transactions. In general, it can be seen in the formula below [37]:

$$Supp(A \rightarrow B) = \frac{|\{t \in D | A \cup B \subseteq t\}|}{|D|} \tag{1}$$

Where, *Supp* is the support value; A is the antecedent of the rule in the form of itemset; B is consequent in the form of itemset; t is a transaction containing A and B; D is the total transaction.

Confidence is another threshold used in determining the rule apart from the support value. It is the ratio between the number of transactions containing items A and B divided by transactions containing item A for rule AB. The confidence value of rule A B is obtained by the formula [37]:

$$Conf(A \rightarrow B) = \frac{|\{t \in D | A \cup B \subseteq t\}|}{|\{t \in D | A \subseteq t\}|} \tag{2}$$

Where, *Conf* is Confidence; A is the antecedent of the rule in the form of itemset; B is consequent in the form of itemset; t is a transaction containing A and B; D is the total transaction.

The general framework of the association rule is to extract a rule with a support value for an item that exceeds the minimum support and confidence value. Therefore, this rule exceeds the minimum confidence value specified by the user. In this case, it can be stated that A B is included in the frequent and confidence category (strong rule) provided that Supp(AB) ≥ minsupp and Conf(AB) ≥ minconf respectively.

To evaluate rule, lift ratio can be used [35, 36, 41–48]. Lift ratio is the ratio between the support value of the rule with the antecedent and consequent support value. The higher of lift ratio value, the more interest rule or called the strong rule. A rule is called interest and strong rule if the lift ratio value > 1 because it shows that is a positive correlation between the premise and conclusion of this association rule [36, 48]. The lift ratio value can be calculated by the formula [36, 41–43, 47, 48]:

$$Lift(A \rightarrow B) = \frac{\sup(A \cup B)}{\sup(A) \times \sup(B)} = \frac{conf(A \rightarrow B)}{\sup(B)} \tag{3}$$

Where, *Lift* is Lift Ratio value; A is the antecedent of the rule in the form of itemset; B is consequent in the form of itemset; *Supp* is the support value; *Conf* is the confidence value.

Minimum support

The basic concept of the association rule requires the user to specify a minimum support value at the beginning. This value usually applies uniformly to all items, although in reality, different items may have different criteria for assessing them. Therefore, studies about multiple minimum support which state that the support value should vary for

different items have emerged [9, 23, 49, 50]. However, the implementation of this system adds a task for the user, which is to determine the minimum support for each item.

Furthermore, the difficulty of determining the minimum support value by the user has led to a new field of study, namely association rules without the use of minimum support such as Top-K [5, 11, 19, 32, 51] and Skyline [6, 10, 33, 52]. The Top-k association rule does not require the minimum support value because in this method the user is only asked to determine the k value, which is the number of rules that will be generated in the rule formation process. Therefore, it is easier for users to determine the k value because they explicitly know the number of rule results they want to obtain.

The Skyline algorithm was first proposed by Borzsony [52] and later developed by Goyal [33]. It is a point that is not dominated by other points [52]. This algorithm was combined by Jerry Chun-Wei Lin [10] and Jeng-Shyang Pan [6] to produce association rules. Furthermore, this study does not use minimum support but instead makes use of the maximum utility (utilmax) the result of each iteration of the utility list structure.

The association rules require a minimum support value to decrease the number of items used in the rule creation process, according to the results of the previous evaluation [20]. In addition, the presence of this threshold can lower the amount of time and memory required throughout the rule-making process. Therefore, regardless of the term, starting from Minimum Utility, Maximum Utility to Minimum Support, this threshold is required for the association rule process. Meanwhile, processes without the threshold will involve all the items present and require more time and memory, which will lead to rules that may not be as desired.

Literature review automate minimum support

Choosing a minimum support value is one of the most difficult aspects of applying association rules. This is because most methods presume that all database items are comparable and occur at the same frequency. However, this assumption is incorrect because some items appear frequently in the database, compared to others [53].

Furthermore, existing algorithms such as apriori and fpgrowth do not have the ability to determine the minimum support and threshold values, therefore, the user estimates these parameters intuitively. The association rule mining algorithm can generate a large number of rules, thereby causing the algorithm to experience long execution times and large memory consumption and vice versa. However, this is dependent on the threshold choice [9].

Users find it difficult to set minimum support, which led to the creation of Apriori-based mining algorithms, a frequent and attractive itemset. This causes a challenging problem due to the performance of this algorithm is highly dependent on some user-defined threshold. For example, assuming the minimum support value is too large, the database becomes empty. Small minimum support, on the other hand, leads to poor mining performance and a slew of unappealing association rules. As a result, users are being asked to identify the specifics of the database to be mined as well as the suitable threshold in an unreasonable manner. Although the minimum support was explored under the supervision of experienced miners, the results were not in accordance with users' needs [15].

Zhang [15], carried out a study with the main contribution of providing a strategy to convert fuzzy (user-defined) thresholds into actual minimum support. As a result, a strategy capable of recognizing some aspects of the database to be mined is required in order to construct a conversion function. Users must still define the real minimum support that corresponds to the database to be mined when using existing Apriori algorithms. However, without proper knowledge, it is impossible to establish the minimal support that matches to the database. Zhang proposed a computational strategy to overcome the problem of minimum support settings. This strategy differs from the existing Apriori algorithm because it allows users to define their mining requirements in a commonly used mode and automatically converts the specified threshold into actual minimum support.

Trivedi [53] carried out a study on the Semi-Apriori algorithm by integrating the average support threshold. This was followed by checking frequent items to determine the data using an automatically generated support threshold to create the itemset more frequently. This reduces time complexity as well as space complexity.

Dahbi [9] designed a method for determining an appropriate minimum threshold value for effective support. The initial contribution was that instead of using user-defined constant values, this study determined the minimum support (minsup) automatically for each data set. Meanwhile, the second made dynamic adjustments (updates) to this minsup by applying a single, standardized minimum support threshold to each level. However, not all objects in an itemset work in the same way; some were used frequently while others were used infrequently. As a result, the minsup threshold must vary depending on the item level.

Kanimozhi [3], stated that a technique with a suitable automatic support threshold at each level is one of the right choices to overcome the problems associated with the minimum support. Therefore, to achieve this task, a technique that uses the automated support system to generate the appropriate rules without losing the rules of interest was proposed based on a Confidence–Lift Measure. This approach was used to determine the initial minsup value by analyzing the itemset and its frequency. It also proposes a cumulative support threshold at the next level using items considered at the previous and current levels.

Based on previous research, most of the determination value of the minimum support is determined by the user. This becomes a problem when the user does not know the characteristics of the dataset. This causes the rule formation process to be repeated to obtain the appropriate number of rules. In this study, an automatic minimum threshold determination method is proposed based on the characteristics of the dataset, so that the user does not need to determine the value of minimum support at the beginning. In addition, if the minimum threshold value is only determined based on frequency, it is unfair for items that have other advantages, so that in this study, the determination of the minimum threshold value also involves other criteria that influence the formation of the rule. In the adaptive support method, the value of minimum support is not determined by the user. This can overcome the difficulties and problems that exist in the current association rule. In addition, in adaptive support the threshold value is not only based on frequency but also involves certain criteria that can give different weights to each item. The proposed threshold value does not need to be recalculated at each level

so that it consumes memory and time more efficiently. The adaptive support method for determining the minimum threshold can be implemented in other association rule algorithms.

The proposed model

To determine the value of minimum support in the association rule, a special method is needed to determine the value according to the characteristics of the dataset which involves certain criteria according to the desires of the user. Therefore, this study developed a method for calculating the minimum support [15], which is similar to the previously used, as shown in Fig. 1.

The adaptive support method comprises 2 types of input, namely transaction datasets and criteria values, which are used to calculate the utility of each item by multiplying their support. This method is also used to determine the frequent itemset, which also involves other predetermined criteria. From the utility results of each item, the average overall utility in the dataset is calculated and the minimum threshold value is obtained, which is further divided by the number of transactions. Furthermore, the algorithm used to determine the value of minimum support based on the characteristics of the database and certain items criteria (utility) is shown in algorithm 1.

Algorithm 1 Adaptive Support Algorithm

```
Pseudocode adaptive_support
Declare
        D = Dataset
        S = support value for each item
        X = criteria value for each item
        U = utility value for each item
        d= item in the dataset
        n= frequency of item
        |D|= the number of transactions in the dataset
        |N| = the number of items in the dataset
Begin
        Sum=sum+0
        Foreach D as d:
          S(d) = n(d) / |D|
          U(d) = S(d) * X(d)
          Sum=Sum+U(d)
        Avesup=Sum/|N|
        Min_threshold=Avesup/|D|
End
```

In the adaptive support model, the determination the value of minimum support is based on several factors which include:

1. Characteristics of the dataset

   In determining the value of minimum support, the user should know in advance, the characteristics of the dataset that will be processed for rule formation. This is because several characteristic factors of the dataset will affect the suitable minimum support value. Furthermore, for an adaptive support mode, the factor used to deter-

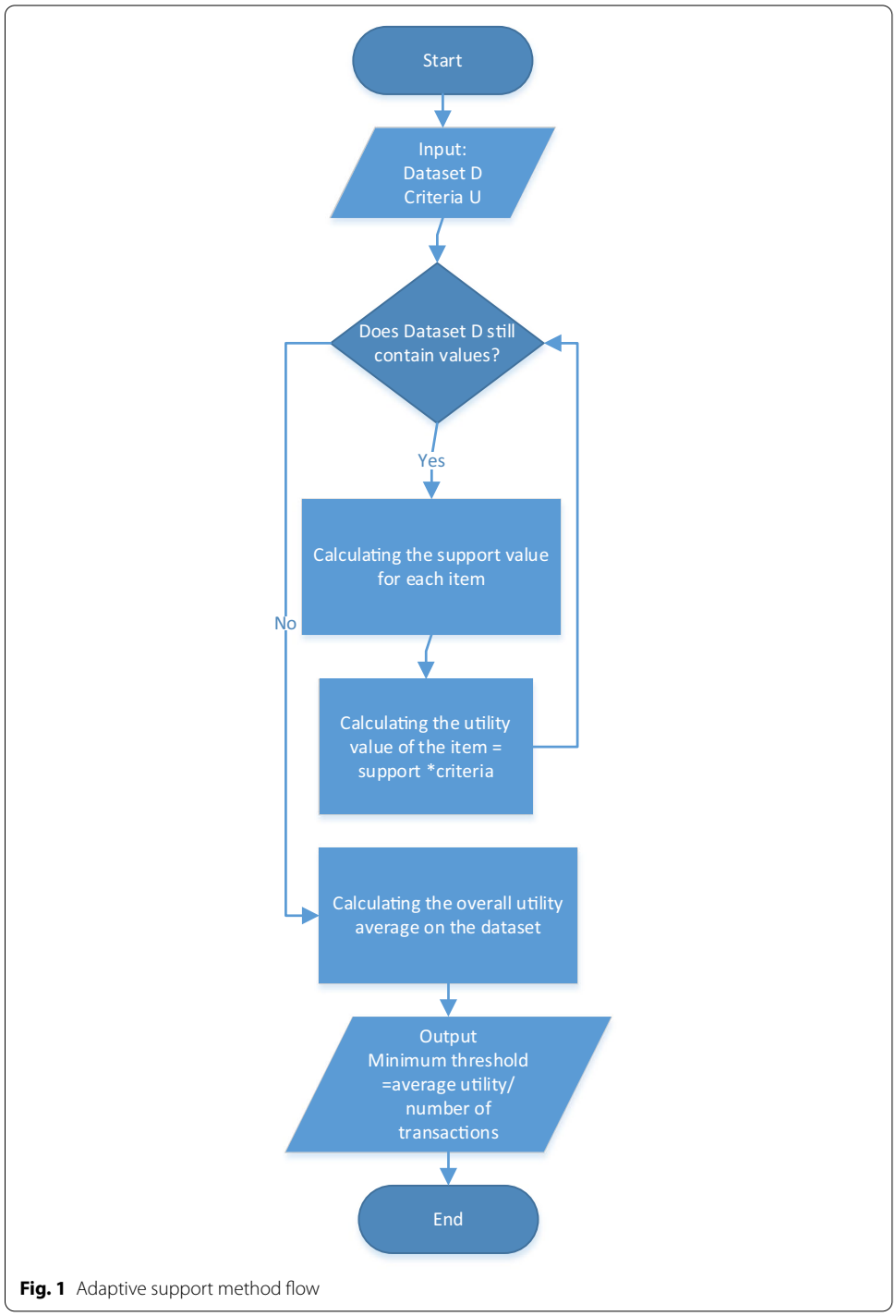

Fig. 1 Adaptive support method flow

mine the value of minimum support is shown from the number of items contained in the transaction, the number of transactions, the average number of items in each transaction, and the support value for each item.

2. Specific criteria (utility for each item)

The occurrence frequency of items is inadequate to be used as a threshold in order to produce adaptive rules. Therefore, a criterion known as items utility is required as an assessment for an item known as a frequent itemset with the right to be a part of the rule-making process. Furthermore, these criteria can be determined by the user and each item has its own criteria values. For example, a user who wants to get a rule for the most expensive item has an involved criterion such as, the item price. In addition, another example is when a user wants a rule with the biggest profit then, the criterion for the item involved is the profit from each item.

Based on both types of inputs, the calculation process is carried out according to algorithm 1 to obtain adaptive support. The calculation stages are as follows:

1. Calculation of the support value for each item in the dataset with the following formula:

$$Sup(d) = \frac{n(d)}{|D|} \tag{4}$$

2. Calculation of the utility for each item in the dataset with the following formula:

$$Util(d) = Sup(d) \times U(d) \tag{5}$$

3. Calculation of the average utility for the entire transaction with the following formula:

$$ave\sup = \frac{\sum Util(d)}{|N|} \tag{6}$$

4. Calculation of the minimum threshold value used for the rule formation process which is the average utility value divided by the total existing transactions through the following formula:

$$\min\sup = \frac{ave\sup}{|D|} \tag{7}$$

5. Performance of the rule formation process with existing methods through the apriori algorithm, fpgrowth, or other algorithms

Where, *Sup(d)* = support value for an item; *n(d)* = number of occurrences for an item; $|D|$ = total transaction; $|N|$ = total item; *U(d)* = utility value for an item; *Util(d)* = utility and support value for an item; *Avesup* = Average utility of the item; *Minsup* = minimum threshold value (item density level).

The determination of the minimum threshold value can be calculated automatically based on the characteristics of the dataset from the proposed adaptive support method. Furthermore, the user does not need to determine the value of minimum support at the beginning or repeat the experiment several times to determine the appropriate value of minimum support.

Table 1 Description of the dataset

No	Dataset	Description
1	Chess [55]	Chess game dataset
2	Mushrooms [55]	This data set contains descriptions of hypothetical samples belonging to 23 species of Agaricus and Lepiota gilled mushrooms (pp. 500–525). Each species is labeled as either definitely edible, definitely poisonous, or maybe edible but not recommended. This last category was merged with the toxic category. There is no easy criteria for determining the edibility of a mushroom, according to the Guide; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy
3	Skin [55]	The skin dataset is created by sampling B,G,R values from face photos of people of various ages (young, middle, and old), races (white, black, and asian), and genders from the FERET and PAL databases at random
4	Accidents [55]	data on traffic accidents that has been anonymised
5	Connect [55]	This database contains all legal 8-ply connect-4 positions in which neither player has yet won and the following move is not forced
6	Retail [55]	Transactions with a Belgian retailer that has yet to be named
7	RecordLink [55]	Individual data such as first and last names, gender, date of birth, and postal code were acquired over the course of several years by iterative insertions
8	Test	Example transaction dataset

Table 2 Characteristics of the dataset

No	Dataset	Number of transactions	Number of items	Average number of items on each transaction
1	Chess [55]	3196	75	37
2	Mushrooms [55]	8416	119	23
3	Skin [55]	245,057	11	4
4	Accidents [55]	340,183	468	33.8
5	Connect [55]	67,557	129	43
6	Retail [55]	88,162	16,470	10.30
7	RecordLink [55]	574,913	29	10
8	Test	3	5	3

Experiment and result

Data experiment

The special dataset for the association rule case used in this study was obtained from SPMF, a Java-based open-source software and data mining library [54]. It is a pattern mining program that is released under the GPL v3 license. SPMF is also linked to 224 data mining algorithms, including itemset mining, sequential pattern rule mining, associated rule mining, sequence prediction, periodic pattern mining, high utility pattern mining, time-series mining, clustering, and classification. Furthermore, it produces larger dataset used to evaluate and compare algorithm performance. Description of the dataset can be seen in Table 1.

The dataset characteristics used can be seen in Table 2.

The study environment used was a laptop with an Intel Core i7-8550U CPU @ 1.80 GHz 1.99GHz, 16 GB of Installed memory (RAM), and a 500GB SSD Hard drive. The tool used for the simulation is using the SPMF [54] software that has been described previously

Table 3 Adaptive support calculation results

Dataset	Density rate (average occurrence of each item in all transactions)	Minimum support (density/number of transactions)	Number of rules with lift ratio > 1
Chess	1577.413	49.35%	320,577,653
Mushrooms	1626.445	19.32%	19,011,494
Skin	8165.67	33.36%	2
Accidents	36,992.1812	11.59%	–
Connect	22,519	33.33%	–
Retail	74.61	0.08%	33,170
RecordLink	213,691.1154	37.16%	9892
Test	1.8	60%	4

Table 4 Number of Rules for each minimum support value with apriori algorithm

Minimum Support value	Number of Rules							
	Chess	Mushroom	Skin	Accidents	Connect	Retail	RecordLink	Test
100%	0	0	0	0	0	0	0	0
90%	10742	22	0	180	3640704	0	180	0
80%	552564	52	0	1432		0	180	0
70%	8111370	180	0	8226		0	1310	0
60%	83864464	266	0	54170		0	3268	4
50%	880936478	1248	0	375774		0	5906	4
40%		5904	2	2764708		0	14012	4
30%		78888	8			2	59754	32
20%		19174370	84			2	86252	32
10%		393890460	166			8	208204	32
5%			256			32	231656	32
0%						35166		

Result

The proposed adaptive support method was carried out on 8 datasets according to Table 1. In this study, other criteria used to determine the minimum threshold were the same for all items, namely valued 1. Furthermore, the results from all 6 of the 8 datasets obtained the appropriate minimum support value and rules with lift ratio >1. The results of this adaptive support calculation process are shown in Table 3.

The trial of the proposed adaptive support method used two basic algorithms in the association rule, namely Apriori and Fpgrowth [10]. The Apriori algorithm used a level-wise approach and generated candidate items for each level [26]. Meanwhile, the fpgrowth algorithm used the pattern growth method but did not generate candidates for each level. After the elimination of itemset with minimum support, the next step was to obtain a Frequent Pattern-Tree [55]. The trial was carried out from the highest minimum support value at 100% and then repeated while a reduction of this value occurred. All minimum support values are not tested on all datasets because the number of rules is different. The more rules that are generated, the more difficult it will be in the decision-making process. For example, in the connect dataset with a minimum support value of 70%, it produces a rule 392,469,141 using the fpgrowth algorithm and consumes a lot of resources (runtime and memory) so the test is stopped. Tables 4, 5, 6, 7 shows the test results of each dataset for each minimum support value using the a priori and fpgrowth algorithms. The blue color in the table is the minimum support value proposed by the adaptive support method.

Table 5 Number of rules for each minimum support value with the fpgrowth algorithm with lift ratio > 1

Minimum Support value	Number of Rules							
	Chess	Mushroom	Skin	Accidents	Connect	Retail	RecordLink	Test
100%	0	0	0	0	0	0	0	0
90%	2,564	22	0	154	341,469	0	110	0
80%	189,334	45	0	841	24,086,200	0	110	0
70%	3,232,975	104	0	5,036	392,469,141	0	966	0
60%	32,975,301	129	0	30,945		0	2,134	4
50%	280,026,979	515	0	209,256		0	3,070	4
40%		3,046	2	1,574,428		0	5,611	4
30%		58,930	8	14,910,124		2	29,580	25
20%		18,982,160	76	209,815,664		2	50,750	25
10%		370,981,205	156			8	121,129	25
5%			209			30	130,524	25
0%			474			33170	1,640,695	25

Table 6 Runtime for each minimum support value with apriori algorithm

Minimum Support Value	Runtime (MS)							
	Chess	Mushroom	Skin	Accidents	Connect	Retail	RecordLink	Test
100%	11	12	0	0	0	78	0	0
90%	29	16	0	1	4820	78	1	0
80%	469	16	0	2		78	0	0
70%	8340	16	0	14		78	1	0
60%	109,935	20	0	58		120	8	0
50%	880,936,478	31	0	306		94	20	0
40%		94	0	3341		93	12	0
30%		531	0			117	85	0
20%		10,030	0			109	131	0
10%		800,330	1			94	184	0
5%			4			140	216	1
0%						1002		

Table 7 Runtime for each minimum support value with fpgrowth algorithm with lift ratio > 1

Minimum Support Value	Runtime (MS)							
	Chess	Mushroom	Skin	Accidents	Connect	Retail	RecordLink	Test
100%	0	0	0	2	0	0	0	0
90%	4	0	0	9	546	0	0	0
80%	367	0	0	22	45,541	0	0	0
70%	6125	0	0	35	996,275	0	4	0
60%	121,658	0	0	159		0	7	0
50%	1,431,849	3	0	816		0	7	0
40%		4	0	6297		0	7	0
30%		73	0	64,794		0	44	0
20%		37,757	0	1,058,700		0	77	0
10%		1,213,180	0			0	168	0
5%			0			0	175	0
0%			4				2683	0

Based on Tables 4–7, it was shown that for each different dataset, the same minimum support value led to a varied number of rules. This was because of the dataset characteristics consisting of the number of items, the average item in each transaction, and the different density values. Furthermore, the trials that have been carried out showed that the higher minimum support value led to a smaller number of rules and runtimes and vice versa. The minimum support value generated by the adaptive rule method when compared with the results of trials using the fpgrowth and apriori algorithms is shown in Fig. 2.

Figure 2 shows that 6 out of 8 datasets or about 75% of the total experimental data obtained a minimum support value between the minimum and maximum values for each algorithm. This means that the value of the proposed adaptive support can generate a rule and if viewed from the quality, adaptive support produces a rule that has a lift ratio value > 1. In addition, with the proposed adaptive support value, users no longer need to guess the appropriate minimum support value that can generate rules. So that there is no longer a possibility that with the specified minimum support value, it does not get a rule, or the number of rules is 0 and must repeat the experiment with another minimum support value. From the experiments that have been carried out, it can be seen that this method is suitable for data sets that have a number of items less than 150 items and are not too dense. with the proposed algorithm can save time for execution because there is no need to guess the appropriate minimum support value. Figure 3 shows that the smaller the minimum support value, the more rules are generated. The red line in Fig. 3 shows the proposed minimum support value based on the adaptive support method.

Figure 3 shows no significant difference between the apriori and fpgrowth algorithms related to the number of rules generated for each minimum support value. Therefore, this study aims to determine the value of the proposed minimum support using the adaptive support method (red line). The result showed that the proposed minimum support value produces a number of limited rules.

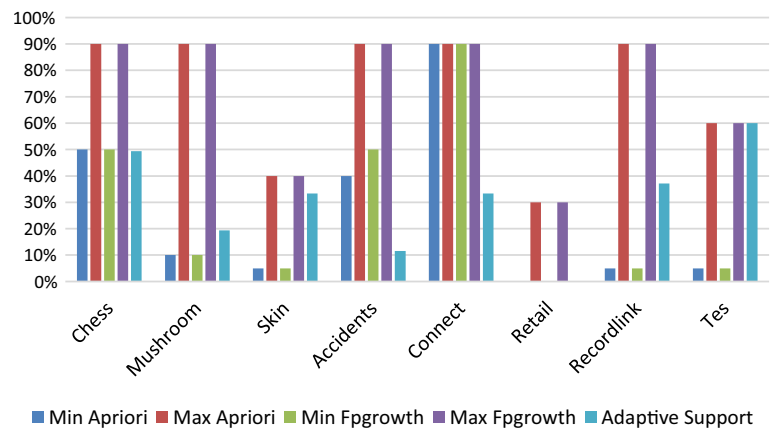

Fig. 2 Comparison of minimum support values using Apriori, Fpgrowth, and Adaptive support

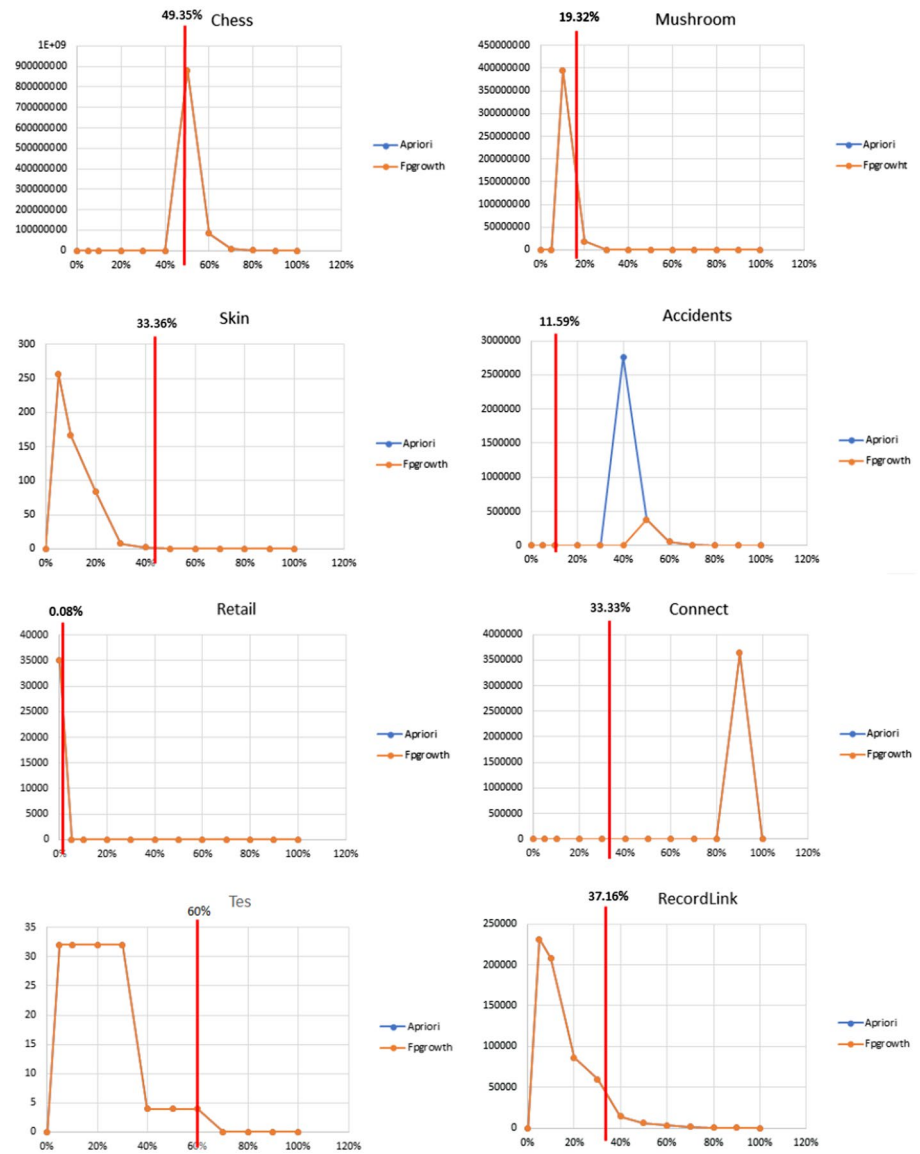

Fig. 3 Test results of the number of rules for each minimum support value

Conclusion and future work

Based on the dataset's characteristics, the minimal support value can be established. The number of transactions, items, and the average number of item frequencies in each transaction are all aspects of the dataset that can be utilized to compute the minimal support value (item density). Datasets with high-density levels require a larger minimum support value compared to those with low-density. Therefore, based on this process, an adaptive support model capable of determining the value of minimum support based on the characteristics of the dataset and certain criteria is proposed. From the experimental results obtained from 8 and 6 datasets, a rule with a lift ratio value > 1 based on the minimum support value is determined. Association rules provide many benefits in real life, one of which is the recommender system. In addition,

association rules can be implemented in classification, information enhancement, promo design, cross selling design, customer loyalty improvement and segmentation. These rules can also be implemented for decision support and recommendation systems.

The minimum support value is important and has a big influence on the rule formation process in the association rule. The errors obtained when determining this resulting rule were not desired. Furthermore, the determination of this value was difficult especially if the user is ignorant of the dataset characteristics as a factor which was used as a reference in determining the value of minimum support. This is because datasets with a high level of density required a larger minimum support value when compared to datasets with low density. This density level is the average value of the occurrence frequency of each item in the dataset. Therefore, an adaptive support model which determined the minimum support value based on the characteristics of the dataset was proposed. In addition, certain criteria leading to adaptive rules following the desires of the user were included.

From the study results on 8 datasets, 6 datasets obtained a rule based on the minimum support value suggested by the adaptive support model. In the future, the author will integrate the multicriteria system model [56] and this adaptive support method to obtain a completely adaptive rule model [22]. In addition, in future research, the author will conduct experiments using datasets that have utility and will use more diverse evaluation methods.

Acknowledgements
We would like to thank Institut Teknologi Bandung, Pasim National University and LPDP (Indonesia Endowment Fund for Education), Ministry of Finance, Republic Indonesia for supporting this research.

Authors' contributions
The author confirms the sole responsibility for this manuscript fully as a sole author for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation. The author read and approved the final manuscript.

Funding
I sincerely thank LPDP (Indonesia Endowment Fund for Education), Ministry of Finance, Republic Indonesia for providing me with the financial support for this research. This research was funded by LPDP (Indonesia Endowment Fund for Education), Ministry of Finance, Republic Indonesia.

Availability of data and materials
The original data used for this study is available in: http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The author reports no potential conflict of interest.

Received: 17 July 2021 Accepted: 8 November 2021
Published online: 25 November 2021

References

1. Luna JM, Fournier-Viger P, Ventura S. Frequent itemset mining: a 25 years review. Wiley Interdiscip Rev Data Min Knowl Discov. 2019. https://doi.org/10.1002/widm.1329.

2. Prajapati DJ, Garg S, Chauhan NC. Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment. Future Computing Inform J. 2017;2:19–30.
3. Selvi CSK, Tamilarasi A. An automated association rule mining technique with cumulative support thresholds. Int J Open Probl Comput Math. 2009;2:12.
4. Zhang C, Zhang S. Association rule mining: models and algorithms. Berlin: Springer; 2002.
5. Ryang H, Yun U. Top-k high utility pattern mining with effective threshold raising strategies. Knowl Based Syst. 2015;76:109–26.
6. Pan JS, Lin JC-W, Yang L, Fournier-Viger P, Hong T-P. Efficiently mining of skyline frequent-utility patterns. Intell Data Anal. 2017;21:1407–23.
7. Zhang S, Wu X. Fundamentals of association rules in data mining and knowledge discovery: fundamentals of association rules. Wiley Interdiscip Rev Data Min Knowl Discov. 2011;1:97–116.
8. Weng C-H, Chen Y-L. Mining fuzzy association rules from uncertain data. Knowl Inf Syst. 2010;23:129–52.
9. Dahbi A, Balouki Y, Gadi T. Using multiple minimum support to auto-adjust the threshold of support in apriori algorithm. In: Abraham A, Haqiq A, Muda AK, Gandhi N, editors. Proceedings of the ninth international conference on soft computing and pattern recognition (SoCPaR 2017). Cham: Springer International Publishing; 2018. p. 111–9. https://doi.org/10.1007/978-3-319-76357-6_11.
10. Lin JC-W, Yang L, Fournier-Viger P, Hong T-P. Mining of skyline patterns by considering both frequent and utility constraints. Eng Appl Artif Intell. 2019;77:229–38.
11. Duong Q-H, Liao B, Fournier-Viger P, Dam T-L. An efficient algorithm for mining the top- k high utility itemsets, using novel threshold raising and pruning strategies. Knowl Based Syst. 2016;104:106–22.
12. Vu L, Alaghband G. An efficient approach for mining association rules from sparse and dense databases. 2014 World Congress on Computer Applications and Information Systems (WCCAIS). Hammamet, Tunisia: IEEE; pp. 1–8. 2021. http://ieeexplore.ieee.org/document/6916550/. Accessed 20 Jun 2021.
13. Boley M, Grosskreutz H. Approximating the number of frequent sets in dense data. Knowl Inf Syst. 2009;21:65–89.
14. Wazir S, Beg MMS, Ahmad T. Comprehensive mining of frequent itemsets for a combination of certain and uncertain databases. Int J Inf Technol. 2020;12:1205–16.
15. Zhang S, Wu X, Zhang C, Lu J. Computing the minimum-support for mining frequent patterns. Knowl Inf Syst. 2008;15:233–57.
16. Alias S, Razali MN, Tan Soo Fun, Sainin MS. Sequential pattern mining using personalized minimum support threshold with minimum items. 2011 International Conference on Research and Innovation in Information Systems. Kuala Lumpur, Malaysia: IEEE; pp. 1–6. 2011. http://ieeexplore.ieee.org/document/6125688/. Accessed 20 Jun 2021.
17. Ghafari SM, Tjortjis C. Association rules mining by improving the imperialism competitive algorithm (ARMICA). In: Iliadis L, Maglogiannis I, editors. Artificial intelligence applications and innovations. Cham: Springer International Publishing; 2016. p. 242–54. https://doi.org/10.1007/978-3-319-44944-9_21.
18. Lin W-Y, Tseng M-C. Automated support specification for efficient mining of interesting association rules. J Inf Sci. 2006;32:238–50.
19. Salam A, Khayal MSH. Mining top—k frequent patterns without minimum support threshold. Knowl Inf Syst. 2012;30:57–86.
20. Hikmawati E, Surendro K. How to determine minimum support in association rule. In Proceedings of the 2020 9th International Conference on Software and Computer Applications. Langkawi Malaysia: ACM; pp. 6–10. 2020. https://doi.org/10.1145/3384544.3384563
21. Giones F, Brem A, Berger A. Strategic decisions in turbulent times: lessons from the energy industry. Bus Horiz. 2019;62:215–25.
22 Hikmawati E, Maulidevi NU, Surendro K. Adaptive rule: a novel framework for recommender system. ICT Express. 2020. https://doi.org/10.1016/j.icte.2020.06.001.
23. Krishnamoorthy S. Efficient mining of high utility itemsets with multiple minimum utility thresholds. Eng Appl Artif Intell. 2018;69:112–26.
24. Liu M, Qu J. Mining high utility itemsets without candidate generation. In: Proceedings of the 21st ACM international conference on Information and knowledge management—CIKM'12. Maui, Hawaii, USA: ACM Press; pp. 55. 2012. http://dl.acm.org/citation.cfm?doid=2396761.2396773. Accessed 26 Sept 2019.
25. Nguyen LTT, Nguyen P, Nguyen TDD, Vo B, Fournier-Viger P, Tseng VS. Mining high-utility itemsets in dynamic profit databases. Knowl Based Syst. 2019;175:130–44.
26. Agrawal R. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases. San Francisco, CA, USA; 1994. pp. 487–99.
27. Agrawal R, Imielinski T, Swami A, Road H, Jose S. Mining Association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. Washington, DC, USA; 1993. pp. 207–61.
28. Yao H, Hamilton HJ. Mining itemset utilities from transaction databases. Data Knowl Eng. 2006;59:603–26.
29. Ahmed CF, Tanbeer SK, Jeong BS, Lee YK. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Trans Knowl Data Eng. 2009;21:1708–21.
30. Lin C-W, Hong T-P, Lu W-H. An effective tree structure for mining high utility itemsets. Expert Syst Appl. 2011;38:7419–24.
31. Tseng VS, Shie B-E, Wu C-W, Yu PS. Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Trans Knowl Data Eng. 2013;25:1772–86.
32 Fournier-Viger P, Wu C-W, Tseng VS. Mining Top-K association rules. In: Kosseim L, Inkpen D, editors. Advances in artificial intelligence. Berlin: Springer; 2012. p. 61–73. https://doi.org/10.1007/978-3-642-30353-1_6.
33. Goyal V, Sureka A, Patel D. Efficient Skyline Itemsets Mining. In: Proceedings of the Eighth International C* Conference on Computer Science and Software Engineering—C3S2E '15. Yokohama, Japan: ACM Press. pp. 119–24. 2008. http://dl.acm.org/citation.cfm?doid=2790798.2790816. Accessed 26 Sept 2019.
34. Choi DH, Ahn BS, Kim SH. Prioritization of association rules in data mining: multiple criteria decision approach. Expert Syst Appl. 2005;29:867–78.

35. Ait-Mlouk A, Gharnati F, Agouti T. An improved approach for association rule mining using a multi-criteria decision support system: a case study in road safety. Eur Transp Res Rev. 2017. https://doi.org/10.1007/s12544-017-0257-5.
36 El Mazouri FZ, Abounaima MC, Zenkouar K. Data mining combined to the multicriteria decision analysis for the improvement of road safety: case of France. J Big Data. 2019. https://doi.org/10.1186/s40537-018-0165-0.
37. Ruiz MD, Gómez-Romero J, Molina-Solana M, Campaña JR, Martin-Bautista MJ. Meta-association rules for mining interesting associations in multiple datasets. Appl Soft Comput. 2016;49:212–23.
38. Ruiz MD, Gómez-Romero J, Molina-Solana M, Ros M, Martin-Bautista MJ. Information fusion from multiple databases using meta-association rules. Int J Approx Reason. 2017;80:185–98.
39. Xiong J, Liu Z. Fuzzy meta association rules based on hierarchy theory based analysis of epidemic incidence of hand, foot and mouth disease in children. Future Gener Comput Syst. 2019;91:574–8.
40. Kantardzic M. Data mining concepts, models, methods, and algorithms. 2nd ed. Hoboken: Wiley; 2011.
41. Alam TM, Shaukat K, Hameed IA, Khan WA, Sarwar MU, Iqbal F, et al. A novel framework for prognostic factors identification of malignant mesothelioma through association rule mining. Biomed Signal Process Control. 2021;68: 102726.
42. Hong J, Tamakloe R, Park D. Application of association rules mining algorithm for hazardous materials transportation crashes on expressway. Accid Anal Prev. 2020;142: 105497.
43. Kim YS, Yum B-J. Recommender system based on click stream data using association rule mining. Expert Syst Appl. 2011;38:13320–7.
44. Lakshmi KS, Vadivu G. Extracting association rules from medical health records using multi-criteria decision analysis. Procedia Comput Sci. 2017;115:290–5.
45. Mirhashemi SH, Mirzaei F. Extracting association rules from changes in aquifer drawdown in irrigation areas of Qazvin plain, Iran. GroundwSustain Dev. 2021;12: 100495.
46. Mukhopadhyay A, Maulik U, Bandyopadhyay S, Coello CAC. A survey of multiobjective evolutionary algorithms for data mining: part I. IEEE Trans Evol Comput. 2014;18:4–19.
47. Telikani A, Gandomi AH, Shahbahrami A. A survey of evolutionary computation for association rule mining. Inf Sci. 2020;524:318–52.
48. Tseng M-C, Lin W-Y. Efficient mining of generalized association rules with non-uniform minimum support. Data Knowl Eng. 2007;62:41–64.
49 Gan W, Lin JC-W, Fournier-Viger P, Chao H-C, Zhan J. Mining of frequent patterns with multiple minimum supports. Eng Appl Artif Intell. 2017;60:83–96.
50. Wu CW, Shie B-E, Tseng VS, Yu PS. Mining top-K high utility itemsets. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '12. Beijing, China: ACM Press. pp. 78. 2012. http://dl.acm.org/citation.cfm?doid=2339530.2339546. Accessed 26 Sept 2019.
51. Lee Y-C, Hong T-P, Lin W-Y. Mining association rules with multiple minimum supports using maximum constraints. Int J Approx Reason. 2005;40:44–54.
52. Borzsony S, Kossmann D, Stocker K. The Skyline operator. In: Proceedings 17th International Conference on Data Engineering. Heidelberg, Germany: IEEE Comput. Soc. pp. 421–30. 2001. http://ieeexplore.ieee.org/document/914855/. Accessed 26 Sept 2019.
53. Trivedi J, Patel B. An automated support threshold based on apriori algorithm for frequent itemsets. Int J Adv Res Innovative Ideas Educ. 2017;3(6):446-52.
54. Fournier-Viger P, Lin CW, Gomariz A, Gueniche T, Soltani A, Deng Z, et al. The SPMF Open-Source Data Mining Library Version 2. In: Proc 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III. Springer LNCS 9853; pp. 36–40. 2016. http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php
55. Pei JHJ. Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. Dallas, Texas, USA; 2000. p. 1–12.
56. Hikmawati E, Maulidevi NU, Surendro K. Multi-criteria recommender system model for lockdown decision of Covid-19. In 2021 10th International Conference on Software and Computer Applications (ICSCA 2021). New York, NY, USA: Association for Computing Machinery; 2021. pp. 39–44. https://doi.org/10.1145/3457784.3457790

Publisher's Note