www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC Gestão, Inovação e Tecnologias

A Systematic Literature Review of Utility Itemset Mining Algorithms for Large Datasets

Vandna Dahiya[1*]; Sandeep Dalal[2]
[1*]Research Scholar, DCSA, Maharshi Dayanand University Rohtak, India.
[1*]vandanadahiya2010@gmail.com
[2]Assistant Professor, DCSA, Maharshi Dayanand University Rohtak, India.

Abstract
Exponential growth has been measured in the size of data during the last two decades. The mining of utility itemsets from a large dataset is a challenging issue because of the diverse dimensions of data. Various itemset mining algorithms have been projected by the researchers to discover relations among the items of a database. In this paper, a systematic literature review has been presented for different algorithms, which are being used for utility itemset mining. 37 studies have been selected to answer the research questions framed for this review based on different methods of mining. These methods have been sorted into four categories with their benefits, drawbacks, performance, and scalability. It has been concluded that research efforts should be geared towards more scalable, secure, and safe methods that can operate more meritoriously on large datasets.

Key-words: Data Mining, Big Data, Utility Mining, Pattern Mining.

1. Introduction

There has been a dramatic shift in intelligent data processing with the expanded capacity of modern applications to produce and store huge quantities of data. This enormous data called as big data is composed of high velocity and variety. Also, larger the size of data, more the value for the business, and the more efficient should be the data mining tool in order to extract meaningful information. Data mining has many day today applications in various sectors like analysis of data in the health sector, computational biology, detection of cyber-crime, web mining, analysis of sentiments, decision making, weather forecasting, etc.

While there are many methods of data mining, association rule mining (ARM) is the key approach. ARM finds the relations between different items of the dataset. Utility Itemset Mining (UIM) is one of the significant areas that finds the utility data for the items among the various itemset mining techniques. It has been originated from the problem of frequent itemset mining. The itemsets with a utility value no less than a user-specified threshold value are termed as itemsets of high utility. The aim is to find all such itemsets from the database.

2. Foundations and Boundaries

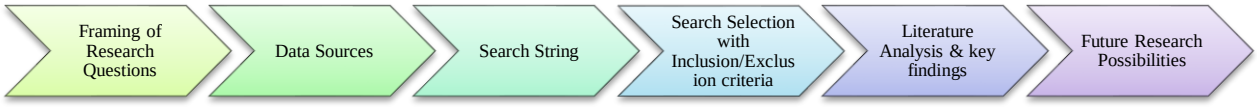
Various researchers have proposed several types of pattern mining techniques in the literature. Itemsets [1, 2], sequences [3, 4], and graphs [5, 6] are some of the patterns, which have been the focus of interest. Rakesh Agrawal et al. [7] introduced the problem of frequent itemsets mining (FIM) in 1993. An approach called the Apriori algorithm has been projected to unravel the problem of FIM. For knowledge discovery and information extraction, Apriori uses prior knowledge for mining frequent itemsets. Apriori algorithm is an iterative approach that uses k itemsets for mining (k+1) itemsets. Apriori uses the subset property, which states that any subset of frequent itemsets must also be frequent. Accordingly, there is no need for rule generation and testing. But this algorithm is very inefficient for execution time due to a lot of candidates. The second generation of mining algorithms began with the introduction of the FP-growth method [8]. This method uses only two scans of the database and itemsets are extracted from FP-tree. Earlier, these methods were having complex data structures and a huge number of projected trees; because of that reason, they were eventually dropped off. New methods with better pruning strategies and simpler data structures have been introduced. Diffsets [9], N-lists [10], Nodesets [11], DiffNodesets [12], and bit-vectors [13] are the most used structures proposed to find frequent itemsets with enhanced performances. Furthermore, because of the less complicated statistics systems of these techniques, distributed frameworks are also better blended with them.

Various parallel frameworks have been developed to efficiently mine the itemsets [14, 15, 16]. FP-tree is exponential in nature. Based on the current research the closed pattern mining [17, 18, 19, 20], maximal pattern mining [21, 22, 23, 24, 25, 26], and significant pattern mining [26, 27, 28] algorithms are the best itemset mining to solve such types of problems. In the FIM, the recurrence of an item in transactional data is always considered as 0 or 1. This implies that an item can be a part of a transaction or can’t be a part of a transaction, however, the figure of its existence can't be more than 1 in every transaction. Moreover, for frequent itemset mining methods, the same unit profit is

considered for every item of the dataset. For the real-world applications, the difference between the number of repetitions of items and the unit profit of items should be considered. The utility itemset mining came into existence with the new version of these two measurements - internal utility and external utility from [30] where a general version of the FIM was recommended. The chief challenge of this version is excessive space and time requirements.

The downward property of Apriori prunes the superset of the itemset for each anti-monotonic function. Transaction weighted utility (TWU) was the first solution to this problem. The two-phase method was proposed, which uses TWU to generate the candidates. Unlike the frequent itemset mining methods, there are several categories for utility itemset mining methods. The first category is grounded on Apriori property. The second category of algorithms is based on a compact tree-like data structure to store the database and uses a pattern growth approach. The high complexity in constructing the tree data structure leads to the emergence of the third category and fourth, which are based on list data structure and database projections. There are some hybrid methods also which combines two or more methods of different category.

Fig. 1 - Research Methodology



3. Research Methodology

Systematic literature review is carried out in a well-structured manner according to the guidelines of Kitchenham [60] and the methodology is being shown in Fig. 1.

A. Research Questions

The scope of this review is to study the available scientific literature on utility mining for big data while focussing on various methodologies and data structures used for this process. The literature search must identify the studies that can address the questions for review, which are as follows:

- What are the various techniques used for utility mining?

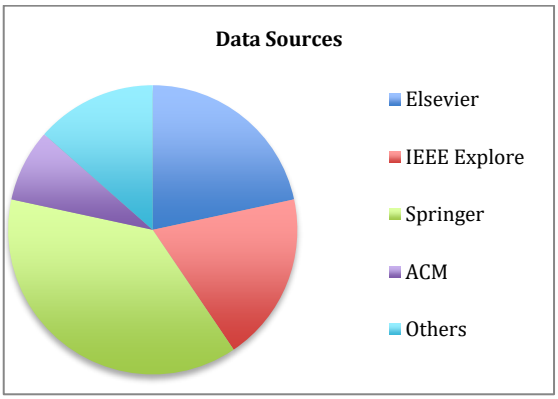
- What are the limitations and strengths of these techniques?
- Do these traditional techniques are applicable to big data as well?

B. Data Sources

According to Kitchenham [60], sources of data are very important for the systematic review Fig. 2. There should not be any biasedness while deciding the sources. For that reason, Google Scholar was searched which offers a platform for all the standard databases such as

- Elsevier
- IEEE Explore
- Springer
- ACM and Others.

Fig. 2 - Major Data Sources for the Literature



C. Literature Search

The search process has been carried out as follows:

First, the search string was structured using the keywords ‘utility mining’ along with the phrases ‘large data’ and ‘big data’ to search the available literature. It returned a total of 43,000 articles, which were not feasible to review manually. The next step was to narrow down the literature work. Inclusion and exclusion criteria were framed based on the language, subject, citations, and publication years. As the interest in the Big Data paradigm started in 2005, the starting year was selected as 2005. Papers were grouped in the category of ‘relevant’ and ‘irrelevant’. In the third step, the relevant research articles were selected based on the titles, abstracts, and conclusions. Then the

180 relevant papers were grouped into four different categories based on the prime method used in them – Apriori based, tree-based, projection-based, and list-based. Articles were then manually reviewed and a final list of 37 articles was prepared for the study based on the central theme and content. These categories are presented in Table 1.

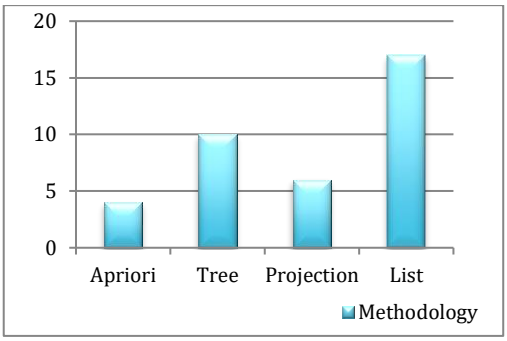
Table 1 - Distribution of Papers over the Studied Years

Year	**2005**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**
Papers	1	2	2	1	1	3	1	2	4	6	5	5	4

4. Literature Analysis

The literature analysis process must answer all the synthesized questions. Based on the literature, the selected articles for utility itemset mining have been divided into four categories based on the methods and data structures used and are explained here.

Fig. 3 - Frequency of Different Methods



A. Categorization of UIM Methods

Broadly, UIM methods are classified into four major categories - Apriori-like methods, Tree-based methods, Projection-based methods, and List-based methods as shown in Fig. 3. Also, some of the algorithms use the strategies of two or more techniques and fall into the category of hybrid methods.

A review has been carried out next based on year of proposal, number of citations, and publications as shown in Table 2.

Table 2 - Summary of UIM Methods

Categories of HUIs	Methodology	Year	No. of Citations	Author Details	Journal
Methods Based on Apriori	Two-Phase	2005	508	Y.Liu et.al *[5]*	Springer
	CHUD	2011	49	C.W. Wu et.al *[34]*	IEEE
	PHUI-Growth	2015	24	Ying Chun Lin et.al *[62]*	Springer
	PHUI-UP	2016	71	JCW Lin et.al *[53]*	Elsevier
Methods Based on Tree	CTU-PRO	2008	209	A Erwin et.al *[40]*	Springer
	IHUP	2009	586	CH Ahmed et.al *[18]*	IEEE
	DTWU	2009	8	Bay Vo et.al *[61]*	Springer
	UP-Growth	2010	390	V.S. Tseng et.al *[36]*	ACM
	UP-Growth+	2012	450	VS Tseng et.al *[55]*	IEEE
	USPAN	2012	203	J.Yin et.al *[39]*	ACM
	HUM-UT	2013	24	L. Feng et.al *[43]*	Intelligent Data Analysis
	REPT	2014	77	H Ryang et.al *[41]*	Elsevier
	HIMU	2016	15	W Gan et.al *[42]*	Springer
	MAHUSP	2017	16	M. Zihayat et.al *[44]*	Springer
Methods Based on Projection	CTU-PROL	2008	209	A Erwin et.al *[40]*	Springer
	PHUS	2014	92	G C. Lan et.al *[71]*	Elsevier
	EFIM	2015	94	S Zida et.al *[46]*	Springer
	CHN	2018	6	K Singh et.al *[47]*	IEEE
	SPHUI-Miner	2018	9	A Bai et.al [*6]*	IEEE
	EHNL	2019	3	K Singh et.al *[31]*	Elsevier
Methods Based on List	HUI-Miner	2012	421	M Liu et.al *[49]*	ACM
	HUI-list-INS	2015	25	JCW Lin et.al *[70]*	The Scientific World Journal
	HUI-list-DEL	2016	12	JCW Lin et.al *[51]*	Intelligent Data Analysis
	PHUI-Miner	2016	22	Chen and An *[37]*	Elsevier
	HUP-Miner	2015	134	S. Krishnamoorthy et.al *[28]*	Elsevier
	BigHUSP	2016	64	Zihayat et.al *[64]*	IEEE
	EFIM-par	2017	40	Tamrakar *[65]*	University Dissertation
	ULB-Miner	2018	27	QH Duong et.al *[52]*	Springer
	P-FHM+	2018	62	Sethi et.al *[66]*	Elsevier
	pEFIM	2018	36	Nguyen et.al *[67]*	Springer
	LHUI-Miner	2019	27	P Fournier et.al *[56]*	Information Science
	PHAUIM	2019	22	Sethi et.al *[68]*	Springer
	FHN	2016	42	JCW Lin et.al *[3]*	Elsevier
	HUOPM	2017	54	W Gan et.al *[54]*	IEEE
	IMHUP	2017	21	H. Ryang *[2]*	Springer
	MUHUI	2017	27	JCW Lin et.al *[54]*	Springer
	DMHUPS	2019	5	BP. Jasawal et.al *[30]*	Springer

1) Apriori-Based Methods

Apriori-based methods are grounded on Apriori property, which is also coined as downward closure property. Accordingly, “All nonempty subsets of a frequent itemset must also be frequent”. Various Apriori-based algorithms have been proposed like Two-Phase (2005), FUP (2011), CHUD (2011), and PHUI-UP (2016), for itemset mining as shown in Table 3. In this conventional itemset mining process, prior knowledge is used to mine the itemsets, that’s why the name is Apriori. The

level-wise search approach is used for example, d-itemset information is used to mine (d+1) itemset. Therefore, this approach is very slow as there are huge segments of candidate itemsets that should be investigated. As noted, the utility of an itemset could be equivalent to, greater, or less than that of its subsets and supersets. So, we cannot use this anti-monotone or downward closure property to reduce the search space. The utility itemset mining algorithm, consisting of two phases and based on the Apriori approach has been proposed in [32]. This approach depends on a utility class model i.e Transaction Weighted Utilization (TWU) model. Following the definition of TWU, the first phase is used to generate the candidates and find all the itemsets with utility less than the user-specified utility by testing. In the second phase, the actual utility value is obtained by scanning the revised database after the TWU model is applied. PHUI-Growth [62] has been recommended for large datasets, which extends the famous FP-Growth with the horizontal representation of data. The algorithm uses MapReduce frame and does not require copying the data at all the sites. The techniques of HUIs based on Apriori methods have been discussed in table 3 with the dataset, their advantages, and disadvantages.

Briefly, based on the literature, it can be said that these UIM methods, which are based on Apriori property use to test and generate techniques. These methods have the advantage of removing some of the unwanted candidate itemsets and thus improving the overall efficiency of the algorithms as can be seen in Two-phase [33] and CHUD [34] and lowers the time for updating the database as in FUP [35]. Large memory consumption and calculations due to the generation of large set of candidates are the shortcomings such as CHUD [34], and FUP [35].

Table 3 Apriori-Based Methods

Algorithm	**Type of Utility Itemsets**	**Dataset**	**Advantages**	**Disadvantages**
Two-Phase (2005)	Complete itemsets of HUIs	T20I6D1000, Chain-store	It generates complete set of utility itemsets and less figure of candidate itemsets	Repeated scans of database are conducted to search the candidate levels
FUP (2011)	High average utility itemsets	Chain-store	Takes less time in reprocessing and data updation	Huge number of candidates
CHUD (2011)	Closed high utility itemsets	Mushroom, Foodmart, BMSWebView1, T10I8D200K	Number of HUIs are reduced	High requirements of memory and time due to transaction merging
PHUI-Growth (2015)	Complete set of HUIs	Retail, Chainstore	A new method of pruning the search space-DLU-MR is used that provides scalability.	Synchronization is slow among the nodes.
PHUI-UP(2016)	Potential high utility itemsets	T10I4D100K, Foodmart, Accident, Retail	Enhanced the performance for uncertain databases	Database is scanned multiple times

2) Tree-Based Methods

Though, algorithms based on Apriori mine UIs capably, suffer from several hitches such as the large number of candidate-generation, repeated database scans, and slow speed due to the complex calculations. To remove these deficiencies, calculations based on HUIM trees are projected with CTU-PRO (2008), UP-Growth (2010), UP-Growth+, Uspan (2012), REPT (2014), HIMU (2016), IHUP (2009), HUM-UT (2013) and MAHUSP (2017) tree-based algorithms as shown in table 4. These algorithms composed of three main phases: 1) tree construction; 2) candidate generation; 3) HUIs identification from these generated candidates. A new data structure for tree-based algorithms called as UP tree is being used in UP-growth [36] and UP-growth+ [38], which is a compact structure and reduces the number of candidates efficiently. UP tree only needs two scans of databases to complete the whole process of mining. A dense utility pattern tree is used for mining UIs by traversing the tree from bottom to top in CTU-PRO [33]. TWU concept is applied for pruning the search space in CTU-PRO. Various novel tree-based algorithms have been proposed to further improve the performance. USpan [39] develops a lexicographic quantitative sequence tree (LQS-tree) by exploiting a sequence-weighted utility (SWU) and a sequence weighted downward closure attribute (SWDC). DHAUIM [30] practices a novel data structure - IDUL prefix tree with a recursive process to maintain the itemsets. To deal with the high utility average pattern, the researcher have proposed two algorithms names MAUGrowth [40], and MAUTree for mining of rare patterns. DTWU-Mining [61] extends the TWU-Mining with a data split strategy for large datasets. The communication cost is low among nodes but does not provide any fault tolerance. Top-k mining approach was proposed without specifying any threshold to find the top k itemsets. REPT [41] is one such technique, where a set of effective rules for top-K HUIs with a less amount of generated candidates has been proposed. By minimizing the threshold, search space is reduced through these techniques. Although these trees have generally smaller structures, they are not minimal and take huge storage space. The functioning of such algorithms hangs on the amount of conditional tree and the expense of traversing each conditional tree throughout the complete mining process. Tree-based techniques are presented with their advantages and disadvantages in Table 4.

Table 4 - Tree-Based Methods

Algorithms	Tree Name	Tree Structure	Data Set	Advantages	Disadvantages
CTU-PRO (2008)	CUP-Tree-(Compressed Utility Pattern)	For every item, there is a node that contains an id, array for TWU values and a pointer to associations an item has	Retail modified, Changed BMSPOS, T10N5D100K, T5N5DXM	Better functioning on spare databases	Generate lots of candidates
IHUP (2009)	IHUPL-Tree (Incremental High utility model lexicographical tree)	Every node encloses the node name, TWU, transaction frequency	Mushroom, Retail, Kosarak, Chain-store	Fewer nodes in the tree	Takes more time in identifying the actual patterns and generates a huge number of candidates
DTWU-Mining	Vertical WIT tree structure	Data Split Strategy	Accidents, Chess, Mushroom	Low communication cost	No fault tolerance
UP Growth (2010)	UP-Tree (Utility Pattern Tree)	Every node encloses an item name, support count, the parent node, utility value and links to other nodes of the same name	BMS-datasets Web-View-1, Chess, T1016D100K	Decreases the number of candidates, as tree is more compact and prevailing	Performs better when the minimum utility value is low
UP-Growth+ (2012)	UP-Tree	Every node encloses an item name, support count, the parent node, utility value and links to other nodes of the same name	Accidents Chess; Chain store, Food Mart	Compact structure of tree, less number of candidates	High time and space requirements
Uspan (2012)	LQS tree (lexicographic Quantitative Sequence tree)	Every node encloses a sequence, the node's child is an I-Concatenated or S Concatenated. Children of nodes are listed in an incremental and alphabetical order	Online purchase transactions, Mobile communication transactions	Low utility itemsets for large scale data are easily identified	Very complex utility matrix with high storage costs
HUM-UT (2013)	UT-Tree structure	Both inner-node and tail-node contains node name, a pointer to the parent node and children node. Tail-node contains utility list	Retail, T10I4D100K	A more stable algorithm which doesn't require additional database scans	Takes more time in processing the tree
REPT (2014)	UP-Tree-(Utility Pattern Tree)	Every node encloses an item name, support count, the parent node, utility value and links to other nodes of the same name	Accident, Chain-stores, Mushrooms, Retail	Reduced number of candidate generation and less search space	Runtime increases
HIMU (2016)	(MIU)-Tree	Every node encloses a name, node-link and a pointer	Foodmart, Mushroom	No repeated scans of database	Requires more time in processing the tree
MAHUSP (2017)	MAS-Tree-(Memory Adaptive-high utility-sequential tree)	Each node contains a node name, node utility, and node Rsu-rest utility	Kosarak, ChainStore, D10KC10T3S412N1K, D100KC8T3S4I2N10K	Adjustable with the available memory	Time-consuming

To summarizing, the performance on larger datasets concerning tree-based methods is better than that of Apriori based methods for dense datasets as well as sparse datasets with the algorithm such as CTU-PRO [40], and UP-Growth [36]. The number of candidates are less with reduced scanning of databases. The major disadvantages of these algorithms are that they take too much time and storage space in generating and storing the conditional tree. Also, the large number of conditional tree generation makes the mining process inefficient as in the case of the CTU-PRO algorithm [40]

and IHUP [8]. Memory requirements are high to check and maintain the conditional tree as in the UP-growth algorithm [36]. The construction of the tree structure requires more time, for example, REPT [41], HIMU [42], HUM-UT [43], and MAHUSP [44].

3) Projection-Based Methods

To overawe the disadvantages of earlier algorithms, projection-based algorithms have been projected to enhance the mining procedure. CTU-PROL (2008), PHUS (2014), EFIM (2017), CHN (2018), SPHUI-Miner (2018), and EHNL (2019) are various such methods used for utility itemset mining as shown in table 5. The idea behind projection-based algorithms is to reuse the processed database as a smaller sub-database by mapping. Itemset or sub-sequence is grown for every mapping with sub-database [45]. In case of insufficient main memory, parallel projections are used for large datasets with disk storage. CTU-PROL [40] generates smaller datasets from the large dataset, which can be accommodated in the main memory. These smaller datasets are then used as parallel projections and mining is performed independently on them. CTU-PROL practices the TWU to clip the search space. Various projection-based algorithms are shown in Table 5.

Table 5 - Projection-Based Methods

Algorithms Name	**Specific Methods**	**Datasets**	**Advantages**	**Disadvantages**
CTU-PROL.(2008)	Parallel projection	Modified Retails, Modified BMSPOS, T10N5D100K, T5N5DXM	The actual utility of itemsets is found without further analysis of database.	A huge number of candidates and high resources are required.
PHUS.(2014)	Prefix-based projection	S8T6I4N4KD200K	Good output in both trimming processes.	A huge number of candidates.
EFIM.(2017)	Projection and merging	Accident, Mushroom BMS, Connect, Chess	Memory requirements are less and complexity is linear in accordance with the items.	Time and memory requirement is very high in recursive projection.
CHN.(2018)	Projection and merging	Accidents, Chess, Mushroom, Pumsb, BMSPOS, kosarak, T40I10D100K	Enhanced performance for dense and sparse datasets.	Redundant candidates with TWU.
SPHUI.Miner (2018)	Projection and merging	Webdocs, Chess, Mushroom, Foodmart, Chainstore,	Fast mining because of reduced scanning time of database.	Less efficient for large database.
EHNL.(2019)	Projection and merging	Accidents, Chess, Mushroom, T40I10D100K	Dataset mapping and transactional consolidation strategies lessen the scanning charges.	Time-consuming with mapping of every itemset.

The projection-based techniques do not require multiple scanning of the databases, which reduce overall cost of the algorithms such as CTU-PROL [40], EFIM [46], CHN calculations [47], and EHNL [31]. Pruning strategies are well developed for projection-based algorithms that increase the efficiency of the algorithms and the subsequence that is obtained for the upper limit of the

sequence is having more accuracy. The major drawback of projection-methods is the generation of redundant candidates such as in CTU-PROL [40] calculations, PHUs [71], CHN [47], and EHNL [31].

4) List-Based Methods

The researchers have explored list-based methods after tree-based methods in HUPM. The steps for mining are 1) construction of utility list for each itemset by scanning the data; 2) filtering the database once again, to adjust the modifications in the utility list; 3) search space is reduced by deleting the itemset with a value less than minutil. A list-based method computes the utility of the itemset, maintains the information about used itemsets, and further reduces the search space and time. HUI-Miner (2012), HUI-list-INS-(2015), HUI-list-DEL-(2016), HUP-Miner-(2015), ULB-Miner-(2018), LHUI-Miner-(2019), FHN-(2016), HUOPM-(2017), IMHUP-(2017), MUHUI-(2017), and DMHUPS-(2019) are various type of UI mining methods shown in table 6, which fall into the categories of list-based methods. HUI-Miner uses a novel structure, utility-list to save the utility information related to the itemset and search space reduction. HUI-Miner finds HUIs form the built-in utility list by not generating the candidates and hence avoid any calculations. The list-structure, datasets, advantages, and their disadvantage are shown in Table 6.

The notion of remaining utility and utility list were first introduced in HUI-Miner [49] for the vertical representations of dataset. Numerous procedures use utility-list data structures along with HUI-list-INS [70] and HUI-list-DEL [51]. Utility list structure can reduce the memory resources by following the merging procedures as shown in ULB-Miner [52], IMHUP [53], HUP-Miner [28], and MUHUI [54]. By expanding list-based methods, the researchers presented another perception for utility mining, for example, HUOPM [57]. PHUI-Miner [37] extends the HUI-Miner for large datasets by implementing in parallel. This is a Spark-based algorithm and provides approximate results by using compression and sampling techniques. BigHUSP [64] is based on Uspan, which is another MapReduce based technique for big data and uses utility matrix representation of data. The method is effective for mining HUIs from large sets but the phases of Mapreduce are four. EFIM-par [65] has been proposed that extends the famous EFIM and is a parallel implementation based on the Spark framework. The algorithm is very efficient for large datasets and an improvement can be done by dividing the workload into a more effective manner. P-FHM+ [66] has been introduced with a desirable length of itemsets to be mined. This algorithm is a parallel implementation of FHM+. But this algorithm suffers from inefficient load distribution. Another Spark-based algorithm for large data

sets is pEFIM [67] that uses multi-core processor-based architecture. But this algorithm uses lots of space due to threads. PHAUIM [68] is another list-based algorithm that is an extension to HAUIM for average utility itemsets. In brief, most of the list-based mining algorithms provide high-speed mining and better performance over dense and spare databases. The utility-list and other-list structures face the problem of a complicated construction process.

Table 6 - List Based Algorithms

Algorithms	List Name	List Structure	Datasets	Advantages	Disadvantages
HUI-Miner (2012)	Utility List	Each node contains tid, iutil and rutil	Chain, Chess, Kosarak, Accidents, Mushroom, Retail, T1014D100K, T40110D100K	Residual and list utility of data is introduced	Construction of list is very time consuming.
HUI-list-INS (2015)	Utility List	Each node contains tid, iutil and rutil	Foodmart, Retail, Chess, T1014D100K	Large number of patterns are generated with low memory consumption	Long runtime
HUP-Miner (2015)	Partition Utility list	Each node contains pk partition, Rup (Remaining utility of itemsets.	Chain, Retail, Mushroom, Chess, T1014D100K, T2016D100K, T40I10D100K, Kosarak	Connection time is reduced	High memory requirements
HUI-list-DEL (2016)	Utility List	Each node contains tid, iutil and rutil	Foodmart, Retail, Mushroom, T1014D100K	Suitable for real applications as no multiple data scans	Long runtime.
FHN (2016)	Positive and Negative Utility list	Each node contains tid, til, nutil and rputil	Mushroom, Accidents, BMS-POS, T10I4D100K, Retail, Chess	Doesn't perform multiple database scans and generate candidates	Performance degrades on sparse dataset
PHUI-Miner (2016)	Utility List	Vertical list	Efficient pruning of search space because of use of the compression and sampling methods	Approximate Results	
BigHUSP (2016)	Utility List	Vertical Utility Matrix	Effective pruning strategeies, less intermediate candidates	Multiple mapreduce phases	
HUOPM (2017)	Utility Occupation list	Each node contains tid, uo, and ruo	BMSPOS2, Chess, T10I4D100K, T40I10D100K, Retail, Mushroom	Provides new research perspectives	Filters valid itemsets at times
IMHUP (2017)	Index utility list	Each node contains iitem, iutil and iindex	Accidents, Chainstore, Chess, Connect Retail, T10I4D100K, T10I4D200K	Without generating any candidate key, joining time is reduced between utility lists	Upper bound is not tight enough.
MUHUI (2017)	Probability utility list	Each node contains tid, prob, iu and ru	Foodmart, Accident, Retail, T10I4D100K.	Provides scalability	Poor Performance on Dense dataset
EFIM-par (2017)	Utility List	Data Split strategy	Connect, Chess	Based on EFIM, method is very effective for generating the candidates	Poor method of task distribution
ULB-Miner (2018)	Utility list buffer	Each node contains tid, iutil, rutil and Sul	Connect, Foodmart, kosarak, Chainstore	Reduced utility lists and small memory consumption.	Construction of utility list is more complicated.
P-FHM+ (2018)	Utility List	Split data strategy with vertical utiltiy list	Chess, Retail, Accidents	Length constraints on HUIs is possible and mining of desired length HUIs is done	Load distribution is inefficient
pEFIM (2018)	Utility list	Multi-core processor-based architecture	Foodmart, Connect, Retail	Static Load balancing is used for parallel working	More threads require more memory as they need their own private space
PHAUIM (2019)	Average utility list	Average utility list	Accidents, Mushroom, Retail	Average utility is considered and search space splitting is efficient	More runtime with average utility
LHUI-Minner (2019)	Local Utility List	Each node contains iutil Periods, util Periods	Mushroom, Retail, Kosarak, E-commerce	Low utilization of memory	Dynamically adjusting the parameters is challenging

B. Limitations of the Review

The analysis in this paper is based on the understanding of authors for some of the selected articles from various standard databases such as Elsevier, ACM, and IEEE. It might be possible to miss out on some of the relevant articles due to various reasons such as

- Non-availability of the article.
- Source language other than English.
- Narrow search string.
- The article could not be included due to some exclusion criteria etc.

These are some of the limitations of this study.

5. Conclusion and Future Scope

Utility itemset mining is a significant chore for mining data. Various methods and procedures have been presented so far for this purpose. Algorithms related to UIM have been presented, which are based on the Apriori property, tree and list data structures and based on projection methods. A systematic review has been done for the structures, advantages, and disadvantages of the algorithms. Design of UIM algorithms for big data is a provocating task as the data is having huge volume, velocity and variety. Many redundant patterns are also there in this data. Based on the analysis, the research efforts can be directed in following directions:

- In candidate generation and processing, UIM consumes a lot of memory and execution time. Efforts to minimise these two factors for realistic implementation should be planned in the future.
- For UIM with big data, the most important research issue is scalability. Other issues are fast computations, security, and privacy.
- UIM is widely used for single data streams, there is a need of processing the multiple data streams in parallel for big data.

References

M. K. Sohrabi and V. Ghods, Top-Down Vertical Itemset Mining, *in Proceedings: 6th International Conference of Graphic and Image Processing (ICGIP 2014)* 2014, pp. 94431V–94431V7.

H. Ryang, and Y. Until "Indexed List-Based High Utility Pattern Mining With Utility Upper-Bound Reduction And Pattern Combination Techniques", *Knowledge and Information Systems* 51, number 2, pp. 627-659, 2017, Available from: https://doi.org/10.1007/s10115-016-0989-x

J. C. W. Lin, Philippe Fournier-Viger, and Wensheng Gan. "FHN: An Efficient Algorithm For Mining High-Utility Itemsets With Negative Unit Profits", *Knowledge-Based Systems* 111, pp. 283-298, 2016. https://doi.org/10.1016/j.knosys.2016.08.022

B. Huynh, B. Vo, and V. Snasel, "An Efficient Parallel Method For Mining Frequent Closed Sequential Patterns", *IEEE Access* 5, pp. 17392–17402, 2017, https://ieeexplore.ieee.org/document/8012369

Y. Liu, W.-K. Liao and A. Choudhary, A Two-Phase Algorithm For Fast Discovery Of High Utility Itemsets, *In Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining,* 2015

A. Bai, S. Parag Deshpande, and Meera Dhabi. "Selective Database Projections Based Approach For Mining High-Utility Itemsets", IEEE Access 6, pp. 14389-14409, 2018, https://ieeexplore.ieee.org/document/8246497

R. Agrawal, T. Imieli and A. Swami, Mining Association Rules Between Sets Of Items In Large Databases, *in Proc. ACM SIGMOD International Conference on Management of Data,* Washington, D.C., USA, pp. 207–216, 1993

J. Han, J. Pei and Y. Yin, Mining Frequent Patterns Without Candidate Generation, *in Proc. ACM SIGMOD International Conference on Management of Data,* SIGMOD Record 29 (2000) 1–12

M. J. Zaki and K. Gouda, Fast Vertical Mining Using Diff Sets, *In Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, D.C., USA,) pp. 326–335, 2003

Z. H. Deng and S. L. Lv, "Prepost+: An Efficient N-Lists-Based Algorithm For Mining Frequent Itemsets Via Children–Parent Equivalence Pruning", *Expert Systems with Applications* 42, 5424–5432, 2015, Available from: https://doi.org/10.1016/j.eswa.2015.03.004

Z. H. Deng and S. L. Lv, "Fast Mining Frequent Itemsets Using Nodesets", *Expert Systems with Applications 42*, 4505–4512, 2014, Available from: https://doi.org/10.1016/j.eswa.2014.01.025

Z. H. Deng, "Diffnodesets: An Efficient Structure For Fast Mining Frequent Itemsets", *Applied Soft Computing 41*, 214–223, 2016, Available from: https://doi.org/10.1016/j.asoc.2016.01.010

Dahiya, O., Solanki, K., Dalal, S., and Dhankhar. A., "Regression Testing: Analysis of its Techniques for Test Effectiveness", *International Journal Of Advanced Trends In Computer Science And Engineering*, volume 9, No. 1, pp. 737-744, 2020, http://www.warse.org/IJATCSE/static/pdf/file/ijatcse74842019.pdf

M. K. Sohrabi and A. A. Barforoush, "Parallel Frequent Itemset Mining Using Systolic Arrays", *Knowledge-Based Systems*, 462–471, 2013, https://doi.org/10.1016/j.knosys.2012.09.005

M. K. Sohrabi, "A Gossip-Based Information Fusion Protocol For Distributed Frequent Itemset Mining", *Enterprise Information Systems* 12, 659–673, 2018, https://doi.org/10.1080/17517575.2017.1405286

M. K. Sohrabi and N. Taheri, "A Hadoop-Based Parallel Mining Of Frequent Itemsets Using N-Lists", *Journal of the Chinese Institute of Engineers* 41, 229–238, 2018, https://doi.org/10.1080/02533839.2018.1454853

A. Tran, T. Truong and B. Le, “Simultaneous Mining Of Frequent Closed Itemsets And Their Generators: Foundation And Algorithm”, *Engineering Applications of Artificial Intelligence* 36, 64–80, 2018, https://doi.org/10.1016/j.engappai.2014.07.004

CF Ahmed, Syed Khairuzzaman Tanveer, Byeong-Soo Jeong, and Young-Koo Lee. "Efficient Tree Structures For High Utility Pattern Mining In Incremental Databases", *IEEE Transactions on Knowledge and Data Engineering* 21, number 12, 1708-1721, 2019, https://ieeexplore.ieee.org/document/4782959

D. Kim and U. Yun, ‘‘Efficient mining of high utility pattern with considering of rarity and length,’’ *International Journal of Speech Technology,* volume 45, no. 1, pp. 152–173, July 2016. https://doi.org/10.1007/s10489-015-0750-2

A. Y. Rodríguez-Gonz alez, F. Lezama, C. A. Iglesias-Alvarez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and E. M. Cote, “Closed Frequent Similar Pattern Mining: Reducing The Number Of Frequent Similar Patterns Without Information Loss”, *Expert Systems with Applications* 96, 271–283, 2018, https://doi.org/10.1016/j.eswa.2017.12.018

U. Yun and G. Lee, “Incremental Mining Of Weighted Maximal Frequent Itemsets From Dynamic Databases”, *Expert Systems with Applications* 54, 304–327, 2016, https://doi.org/10.1016/j.eswa.2016.01.049

J. Sahoo, A. K. Das, and A. Goswami, “An Effective Association Rule Mining Scheme Using A New Generic Basis”, *Knowledge and Information Systems* 43,127–156, 2015, https://doi.org/10.1007/s10115-014-0732-4

Z. Liu, L. Hu, C. Wu, Y. Ding, Q. Wen and J. Zhao, “A Novel Process-Based Association Rule Approach Through Maximal Frequent Itemsets For Big Data Processing”, *Future Generation Computer Systems* 81, 414–424, 2014, https://doi.org/10.1016/j.future.2017.08.017

M. R. Karim, M. Cochez, O. D. Beyan, C. F. Ahmed and S. Decker, “Mining Maximal Frequent Patterns In Transactional Databases And Dynamic Data Streams: A Spark-Based Approach”, *Information Sciences* 432, pp. 278–300, 2018, https://doi.org/10.1016/j.ins.2017.11.064

H. Fishy and M. H. N. Shahraki, “Incremental Mining Maximal Frequent Patterns From Univariate Uncertain Data”, *Knowledge-Based Systems* 152, pp. 40–50, 2018, https://doi.org/10.1016/j.knosys.2018.04.001

M. K. Vanahalli and N. Patil, “An Efficient Parallel Row Enumerated Algorithm For Mining Frequent Colossal Closed Itemsets From High Dimensional Datasets”, *Information Sciences,* 2018, https://doi.org/10.1016/j.ins.2018.08.009

M. K. Sohrabi and A. A. Barforoush, “Efficient Colossal Pattern Mining In High Dimensional Datasets”, *Knowledge-Based Systems* 33, pp. 41–52, 2012, https://doi.org/10.1016/j.jksuci.2020.04.008

S. Krishnamoorthy, ‘‘Pruning Strategies For Mining High Utility Itemsets,’’ *Expert System and Applications,* volume 42, number 5, pp. 2371–2381, 2015, https://doi.org/10.1016/j.eswa.2014.11.001

Dhankhar, K. Solanki, A. Rathee and Ashish, “Predicting Student’s Performance by using Classification Methods,” *International Journal of advanced trends in computer science and engineering,* volume 8, number 4, 2019, DOI: 10.26438/ijcse/v6i7.4348

Jaysawal, Bijay Prasad, and Jen-Wei Huang. "DMHUPS: Discovering Multiple High Utility Patterns Simultaneously." *Knowledge and Information Systems* 59, number 2, 337-359, 2019, https://doi.org/10.1007/s10115-018-1207-9

K. Singh, A. Kumar, S. S. Singh, H. K. Shakya, and B. Biswas, ‘‘EHNL: An Efficient Algorithm For Mining High Utility Itemsets With Negative Utility Value And Length Constraints,’’ *Information Science*, volume 484, pp. 44–70, 2019, https://doi.org/10.1016/j.ins.2019.01.056

T. Le and B. Vo, “An N-List-Based Algorithm For Mining Frequent Closed Patterns”, *Expert Systems with Applications* 42, 6648–6657, 2015, https://doi.org/10.1016/j.eswa.2015.04.048

B. Vo, S. Pham, T. Le, and Z. H. Deng, “A Novel Approach For Mining Maximal Frequent Patterns”, *Expert Systems with Applications* 73, 178–186, 2017, https://doi.org/10.1016/j.eswa.2016.12.023

C. W. Wu, P. Fournier-Viger, P. S. Yu, and V. S. Tseng, ‘‘Efficient Mining of A Concise And Lossless Representation Of High Utility Itemsets,’’ *In Proc. IEEE 11th International Conference on Data Mining*, pp. 824–833, 2011, https://ieeexplore.ieee.org/document/6137287

T. P. Hong, C. H. Lee, and S. L. Wang, ‘‘An Incremental Mining Algorithm For High Average-Utility Itemsets,’’ *In Proc. 10th Int. Symp. Pervez. Syst., Algorithms, Netw.*, pp. 421–425, 2010, https://ieeexplore.ieee.org/document/5381569

V. S. Tseng and C. W. Wu, UP-growth: An Efficient Algorithm For High Utility Itemset Mining, in Proceedings ACM SIGKDD *International Conference on Knowledge Discovery Data Mining*, Washington, DC, USA, pp. 253–262, 2010.

Chen Y., A. An, “Approximate Parallel High Utility Itemset Mining”, *Big Data Research*, Volume 6, pp 26-42, 2016, https://doi.org/10.1016/j.bdr.2016.07.001

V. S. Tseng, B. E. Shie, C. W. Wu, and P. S. Yu, ‘‘Efficient Algorithms For Mining High Utility Itemsets From Transactional Databases,’’ *IEEE Transactions, Knowledge and Data Engineering.*, volume 25, number 8, pp. 1772–1786, Aug. 2013, https://ieeexplore.ieee.org/document/6171188

J. Yin, Z. Zhang, and L. Cao, USpan: An Efficient Algorithm For Mining High Utility Sequential Patterns, in Proc. 18th ACM SIGKDD *International Conference on Knowledge and Discovery Data Mining*, pp. 660–668, 2012.

A. Erwin, R. P. Gopalan, and N. R. Achuthan, Efficient Mining Of High Utility Itemsets From Large Datasets, In Proc. 12th Pacific–*Asia Conference Advanced Knowledge Discovery and Data Mining*, pp. 554–561, 2008

H. Ryang and U. Yun, ‘‘Top-k High Utility Pattern Mining With Effective Threshold Raising Strategies,’’ *Knowledge-Based Systems*, volume 76, pp. 109–126, 2015, https://doi.org/10.1016/j.knosys.2014.12.010

W. Gan, J. C. W. Lin, P. Fournier-Viger, and H.-C. Chao, ‘‘More Efficient Algorithms For Mining High-Utility Itemsets With Multiple Minimum Utility Thresholds,’’ *in Database and Expert Systems Applications (Lecture Notes in Computer Science)*, volume 9827, pp. 71–87, 2016, https://doi.org/10.1007/978-3-319-44403-1_5

L. Feng, L. Wang, and B. Jin, ‘‘UT-tree: Efficient Mining Of High Utility Itemsets From Data Streams,’’ *Intell. Data Anal.*, volume 17, number 4, pp. 585–602, 2013, https://dl.acm.org/doi/10.5555/2595577.2595580

M. Zihayat, Y. Chen, and A. An, ‘‘Memory-Adaptive High Utility Sequential Pattern Mining Over Data Streams,’’ *Machine Learning*, volume 106, number 6, pp. 799–836, 2017, https://doi.org/10.1007/s10994-016-5617-1

W. S. Gan, J. C. W. Lin, and Fournier-Viger, ‘‘A Survey of Utility-Oriented Pattern Mining,’’ *J. of Latex Class Files*, volume 6, number 1, 2018, https://ieeexplore.ieee.org/document/8845637

S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, EFIM: A Highly Efficient Algorithm For High-Utility Itemset Mining, *in Proceeding International Conference on Artificial Intelligence,* pp. 530–546, 2015

K. Singh, S. S. Singh, A. Kumar, H. K. Shakya, and B. Biswas, ‘‘CHN: An Efficient Algorithm For Mining Closed High Utility Itemsets With Negative Utility,’’ *IEEE Transactions Knowledge Data Engineering.*, 2018, https://ieeexplore.ieee.org/document/8540872

Sandeep Dalal, Vandna Dahiya, “A Novel Technique - Absolute High Utility Itemset Mining (AHUIM) Algorithm for Big Data”, *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE),* volume 9, Issue 5, pp 7451-7460, 2020, http://www.warse.org/IJATCSE/static/pdf/file/ijatcse78952020.pdf

M. Liu and J. Qu, Mining High Utility Itemsets Without Candidate Generation, in Proc.21st ACM *International Conference on Information and Knowledge Management* (CIKM), HI, USA, pp. 55–64, 2012.

Vandna Dahiya, Sandeep Dalal, “Big Data Mining: Current Status and Future Prospects”, *International Journal of Advance Science and Technology*, volume 29, number. 3, pp. 4659-4670, 2020, *http://sersc.org/journals/index.php/IJAST/article/view/5681*

J. C. W. Lin, W. Gan, and T.-P. Hong, ‘‘A Fast Maintenance Algorithm Of The Discovered High-Utility Itemsets With Transaction Deletion,’’ *Intelligent Data Analysis*, volume 20, number 4, pp. 891–913, 2016, http://doi.org.10.3233/IDA-160837

Q.-H. Duong, P. Fournier-Viger, H. Ramampiaro, K. Nørvåg, and T.-L. Dam, ‘‘Efficient High Utility Itemset Mining Using Buffered Utility Lists,’’ *International Journal of Speech Technology*, volume 48, number 7, pp. 1859–1877, 2018, http://doi.org/10.1007/s10489-017-1057-2

J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, ‘‘Efficient Algorithms For Mining High- Utility Itemsets In Uncertain Databases,’’ *Knowledge Based Systems*, vol. 96, pp. 171–187, 2016, https://doi.org/10.1016/j.knosys.2015.12.019

J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, ‘‘Efficiently Mining Uncertain High-Utility Itemsets,’’ *Soft Computing*, volume 21, number 11, pp. 2801–2820, 2017, https://doi.org/10.1007/s00500-016-2159-1

V. S. Tseng, B. E. Shie, C. W. Wu, and P. S. Yu, ‘‘Efficient Algorithms For Mining High Utility Itemsets From Transactional Databases,’’ *IEEE Transactions Knowledge and Data Engineering*, volume 25, number 8, pp. 1772–1786, 2013, https://ieeexplore.ieee.org/document/6171188

P. Fournier-Viger, Y. Zhang, J. Chun-Wei Lin, H. Fujita, and Y. S. Koh, ‘‘Mining Local And Peak High Utility Itemsets,’’ *Information Science*, volume 481, pp. 344–367, 2019, https://doi.org/10.1016/j.ins.2018.12.070

W. Gan, J. C. W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, ‘‘HUOPM: High-Utility Occupancy Pattern Mining,’’ *IEEE Transactions Cybernetworking*, volume 50, number 3, pp. 1195–1208, 2020, https://ieeexplore.ieee.org/document/8645787

T. Hashem, M. R. Karim, M. Samiullah, and C. F. Ahmed, “An Efficient Dynamic Superset Bit-Vector Approach For Mining Frequent Closed Itemsets And Their Lattice Structure”, *Expert Systems with Applications* 67, pp. 252–271, 2017, https://doi.org/10.1016/j.eswa.2016.09.023

F. A. M. Zaki and N. F. Zulkurnain, “RARE: Mining Colossal Closed Itemset In High Dimensional Data”, *Knowledge-Based Systems* 161, pp. 1–11, 2018, https://doi.org/10.1016/j/knosys.2018.07.025

Kitchenham, B., “Procedures for Performing Systematic Reviews”, Keele, UK, Keele University, 33, pp. 1-26, 2004.

Bay Vo, Huy Nguyen, Tu Bac Le, Parallel Method for Mining High Utility Itemsets from Vertically Partitioned Distributed Datasets, *International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, Springer, pp 251-260, 2009.

Ying Chun Lin, Cheng-Wei Wu, Vincent s. Tseng, Mining High Utility Itemsets in Big Data, *Pacific Asia Conference on Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, Springer, pp 649-661, 2015.

Sandeep Dalal, Vandna Dahiya, “Big Data Preprocessing: Needs and Methods”, International Journal of Engineering Trends and Technology, volume 68, Issue 10, pp 100-104, 2020, *http://ijettjournal.org/archive/ijett-v68i10p217*

Zihayat M., Hut Z. Z., An A., Hut Y., Distributed and Parallel High Utility Sequential Pattern Mining, *International Conference on Big Data*, IEEE, pp 853-862, 2016.

A. Tamrakar, *“High Utility Itemsets Identification in Big Data,”* Doctoral dissertation, University of Nevada, Las Vegas, 2017.

K.K. Sethi, D. Ramesh and D.R. Edla, “P-FHM+: Parallel High Utility Itemset Mining Algorithm For Big Data Processing.” *Procedia computer science*, 132, pp. 918 927, 2018, https://doi.org/10.1016/j.procs.2018.05.107

Nguyen, T.D., Nguyen, L.T., & Vo, B., A Parallel Algorithm For Mining High Utility Itemsets. In *International Conference on Information Systems Architecture and Technology*, Springer, Cham, 286-295, 2018.

Sethi, K.K., Ramesh, D., & Sreenu, M., Parallel High Average-Utility Itemset Mining Using Better Search Space Division Approach. In *International Conference on Distributed Computing and Internet Technology*, Springer, pp 108-124, 2019.

Vandna Dahiya, Sandeep Dalal, “Parallel Approaches of Utility Mining for Big Data,” Webology, volume 17, issue 2, pp. 31- 43, 2020, https://www.webology.org/abstract.php?id=293

J. C. W. Lin, W. Gan, T.-P. Hong, and B. Zhang, ‘‘An incremental high-utility mining algorithm with transaction insertion,’’ *Science World Journal*, volume 2015, pp. 1–15, Feb. 2015, https://doi.org/10.1155/2015/161564

G.-C. Lan, T.-P. Hong, V. S. Tseng, and S.-L. Wang, ‘‘Applying the maximum utility measure in high utility sequential pattern mining,’’ *Expert Systems and Applications*, volume 41, number 11, pp. 5071–5081, Sep. 2014, https://doi.org/10.1016/j.eswa.2014.02.022

Chen Y., A. An, “Approximate Parallel High Utility Itemset Mining”, *Big Data Research*, volume 6, pp 26-42, 2016, https://doi.org/10.1016/j.bdr.2016.07.001