A SURVEY ON FREQUENT PATTERN MINING IN UNCERTAIN BIG DATA G Divya Zion #1 , B K Tripathy *2 #1

Research · July 2021

DOI: 10.13140/RG.2.2.24931.07208

2 authors:

B.K. Tripathy
VIT University
736 PUBLICATIONS **4,269** CITATIONS
SEE PROFILE

Divya Zion
4 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Design Engineering

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

A SURVEY ON FREQUENT PATTERN MINING IN UNCERTAIN BIG DATA

G Divya Zion [#1], B K Tripathy [*2]

[#1]Research Scholar, School of Computer Science & Engineering, Vellore Institute of Technology Vellore, Tamilnadu, INDIA

[*2]Dean & Senior Professor, School of Information Technology and Engineering, Vellore Institute of Technology Vellore, Tamilnadu, INDIA

[1]ziondivya@gmail.com, gdivya.zion2016@vitstudent.ac.in, [2]tripathybk@vit.ac.in

Abstract - Pattern mining is a significant way to mine data in attaining fascinating correlation statistics within the dataset. In that respect, different versions of pattern mining techniques are considered, specified as frequent itemset mining, sequence pattern mining, and structure mining. Frequent Pattern mining is a rising analytical task which is performed on data statistics using Big data, Machine Learning, Data Mining tools, and drives to bring out knowledge supported with domain target. Lately, researchers hold up close to mining frequent pattern across uncertain transaction dataset. Researchers get hold of to mining itemsets on uncertain big data that have got voluminous attention supported with Apache Hadoop and Spark framework. This paper attempts to present a comprehensive outline using specific approaches to perform pattern mining on uncertain data in the Big Data aspects. At first, we look into the trouble attached with pattern mining techniques related to Apache Hadoop, Apache Spark, parallel and distributed processing and examine them in the big data view. Frequent itemsets mining in uncertain data has been analyzed and this composition reasons out with open issues and further enhancement of present approaches.

Keywords —Pattern Mining, Frequent Pattern Mining, Uncertain Data, Big Data

I. INTRODUCTION

A Pattern is a geometrical regularity or recurrence in the earthly concern, and the components inside a pattern recur in a predictable or certain mode. Patterns can be noticed through any senses, but candid observance of patterns intends to visual or seeable patterns. Visual patterns in nature never recapitulate precisely and it is often chaotic or disorganized. Peter S. Steven's titles that there are merely a bounded number of techniques that pattern can be incorporated. A pattern comprises a group or a set of objects or numbers where all of it is associated to each other. Unremarkably patterns can be discovered or noticed by senses and the components from those patterns can be reiterated in a predictable process. Sometimes, patterns can be observed by human directly and some of the times they're detected through system analytic thinking. **Data mining which is renowned for its** knowledge **exploration in databases** finds readable and

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

appropriate patterns and their associations within large volumes of data. To scrutinize massive digital agglomerations known as data sets, tools such as neural networks and machine learning from statistics and artificial intelligence are considered [107].

As presented in Figure 1, a pattern can be distinguished through accumulating or pulling together the information from the databases or data storages; clean the gathered up data from the noise or disturbance, distinguish or analyze patterns for similarities by grouping into sections, examine the sections and hence build pattern predictions.

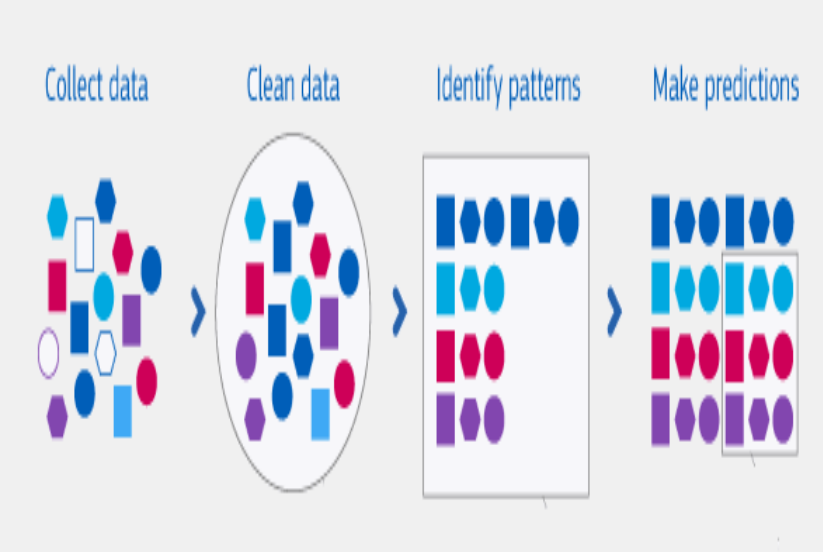

Figure 1: Pattern Recognition Process

Pattern mining is usually done on precise patterns within the data. The study analysis on Market-basket example that was the first applications of data mining, determines items that commonly appear together in purchasing negotiations. With Market-basket analysis data mining is capable of being easily understood the associations in large volume data sets. Above all discovery of unexpected combinations of data items have pave a new path for marketing or research. Another use of pattern mining is the discovery of sequential patterns; for example, sequences of errors or cautions that lead to a devices failure can sometimes be considered as preventative sustainment to overcome design flaws. Pattern Examples: Number Patterns, geometrical patterns, Bubbles, waves, spirals in Nature pattern, Weather graphs on TV and the computer, Tiling's in Art and Architecture, Fractals in Science and Mathematics, Design Patterns in Computer Science, Fashion, a template for manufacturing a dress, Cricket Score graphs [109].

Pattern mining is generally executed on datasets where the state of being or non-being of itemsets is sure. But the data which is addressable could be filled with uncertainty in many real world applications. Uncertainty refers to deficiency of perfect data and this leads to situations which doesn't bring out exact outcomes. Frank Knight says-"You cannot be certain about Data Uncertainty". Uncertainty is seen in many fields like information science, finance, statistics and

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

many more. Data Uncertainty is present in all aspects of big data and show up from innumerable various sources. There are many sources that lead big data to uncertainty when large datasets are generated. Some of them are,

- Outsourcing of data from various data sources has low quality of data.
- Incorrect posts from social media
- Abnormality in data occurs, when data from two sources provide different data sets, Sensor's data.
- Measurement values for any calculation
- Data when filled with vague statements

In recent years, mining data from uncertain data has turn into an interesting study. In "Uncertainty based Big Data", the word Uncertain Data refers to the data that contains noise which makes the data to get changed from the correct and original values. In Big Data, as the data grows exponentially from various data sources irrespective of time. For example: data generated from sensors, enterprise data, web services data are in different formats and there is a possibility for data to be uncertain. This uncertain data is generated in structured and unstructured form, and from numerous applications [1, 2, 3, and 5].

There are three main models of uncertain data in data bases:

1. Attribute Uncertainty:

Each attribute which is uncertain in a tuple is subjected to its own independent probability distribution. Example: If temperature and wind speed readings are taken, both the readings are based on their individual probability, where readings of temperature will not provide any information about readings of wind speed [4].

2. Correlated Uncertainty:

In this, multiple attributes are described by Joint probability. Example: when measuring an objects position its coordinates are needed. The probability of different objects values may depend on the distance from the recorded coordinates of first object. To calculate distance it depends on both coordinates of first and second object, it is appropriate to use Joint probability [6].

3. Tuple Uncertainty:

When all the attributes of a tuple are described by joint probability then it is called as correlated uncertainty. But there is a chance that if a tuple is not relevant to any other [7]. Uncertainty in data is classified into two types:

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

- Existential Uncertainty: Occurs in data streams, with the value being uncertain [8, 9]. For example, tuples are developed when an event or rule is satisfied. If the rule or event is uncertain then generation of new tuples may not be possible.
- Value Uncertainty: A data item which is bound within the possible values [10,11].

Uncertainty is a common issue both in Data Mining and Big Data, Uncertainty exists more in Big Data, as the data gets collected from many sources like web search engines, hospital records, sensor data, weather forecasting data, social media data and many more. Some of the values of these data get changed as the time varies and this leads to data inconsistency or missing values. Traditional methods for processing and analyzing uncertain big data with time and cost efficiency is not possible. Some of the new models have been proposed to handle big data. Map Reduce has been introduced by Google and has become the most basic platform for large-scale mining. MapReduce can be applied upon CPU, GPUs, GRIDs and the cloud with the help of multi-core devices. MapReduce adds up computer nodes to a system and broadcasts the load across them. Many additional large-scale data processing systems are developed to improve horizontal scalability. For Big data applications, Spark offers mutual and repetitive calculations [92], it is a cluster computing model that offers iterative dataflow processing. NoSQL (Not only SQL) databases like Cassandra [93] and MongoDB [94] provides scalability, schema-free and consistency. To process data SQL languages are used over execution engines such as Hive [95]. In Section 2, an overview on pattern mining problem and its differentiations are discussed. In section 3, big data processing paradigms are talked about. In Section 4 explains various criteria used for the comparison of the methods. Section 5, concludes various parallel and distributed computing algorithms to perform pattern mining in big data, Section 6 concludes the discussion about challenges.

II. PATTERN MINING CLASSIFICATION

Various pattern mining approaches like Frequent Itemset Mining, Sequence Pattern Mining, Structure Pattern Mining and Uncertain mining are explained. Figure 2, explains the road map on pattern mining research. Many articles address three pattern mining aspects: the kinds of patterns mined, mining methodologies and their applications. However, different applications may need to mine different patterns, which obviously lead to the development of new mining methodologies

2.1. Frequent Itemset Mining

To The intent of frequent itemset mining is aimed at finding regularities and frequent itemset is defined as, finding the compatability(support) of an Item set, which should be greater than or equal to the minimum support count specified by the consumer. An itemset constitutes collection of items such that group of items which are bought together in a grocery store by a customer. In this context, the itemsets are considered based on number of transactions that are made and algorithms need to search for common item sets which should exist in the same dataset for atleast

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

a minimum number of times.To perform frequent itemset mining,

Definition 1: Support:

Let A=set of items or an itemset, in a transaction database T, itemset set A is present if and only if A ⊆ T.

$$\text{Support (A)} = \frac{\text{Number of transactions in which A appears}}{\text{Total number of Transactions}}$$

Definition 2: Frequent Itemset:

An itemset A is a frequent itemset, if sup (A) is not less than the minimum support threshold assigned by the user (i.e. sup (A) _ minsup).

Definition 3: Confidence:

If the rule is of the form A➔B, where

$A \subseteq I, B \subseteq I,$ and $A \cap B \neq \Phi$ then

$$\text{Confidence}(A \rightarrow B) = \frac{\text{support of } (A \cup B)}{\text{support of A}}$$

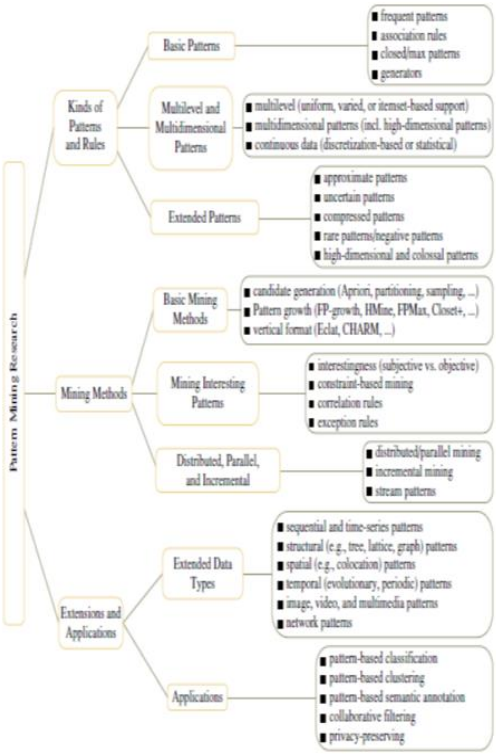

Figure 2: Pattern Mining Classification

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

There are many algorithms for finding frequent itemsets and some algorithms are improved based on its efficiency and scalability. Frequent itemsets are represented in different compact representations;

Constraint Frequent Itemset:

Finding all the independent frequent itemsets available in the database is something which is unrealistic. Users are attracted by only some contents of frequent itemsets which can be directed using query language. Constraint-based mining provides flexibility to users that perform mining based on user constraints [46]. Some of the constraints in Data Mining are;

- Knowledge based Constraint: Clustering, Association and Classification etc.
- Data Constraint: Find the sales of items that are sold together throughout the year in India.
- Dimension based Constraint: Constraint that is based on the column name like age, salary, etc.
- Interestingness Constraint: Constraints or rule that are based on minimum support and confidence.

Constrained frequent itemset mining has three properties; [47] **Anti-monotonicity:**

If an itemset P doesn't satisfy the constraint, so all of its superset doesn't follow or if a Constraint R is not satisfied or violated then the further mining procedure can be stopped.

If P is subset of P`, where P satisfies constraint R, then $->$ P` also satisfies R.

For Example:

$$\text{Min}(P) \geq R, \text{ where P is Itemset and R is constraint}$$

$$\text{If Itemset P} = \{2,5,10,14\}, R = 7, \text{Min}(P) \geq 7$$

$$\text{If } \{2\}\text{violates it, then superset of } \{2\} = \{2,5\},\{2,10\},\{2,14\},\{2,5,10,14\}\text{also violates it.}$$

$$\text{Hence, Min}(P) \geq R \text{ is anti} - \text{monotonicity.}$$

Monotonicity:

A Constraint R is monotone, if a pattern satisfies R and it is not necessary to check R in subsequent patterns or if an itemset P satisfies the constraint, so all of its superset satisfies. $\text{Min}(P) \geq R$, where P is Itemset and R is constraint

$\text{if, Min(item)} \geq 2$

$\text{Item } \{a\}\text{satisfies } R = 2,$

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

so does every superset of {a} is monotone

Succinctness:

Consider P1 as one of the item in an Itemset P which satisfies a succinctness constraint R, which therefore implies that any Itemset P satisfying constraint R is based on one of the item P1 in itemset P i.e.; P contains a subset that belongs to P1.

Closed frequent itemset:

It is a common itemset for which no superset exists but has the same minimum support as the itemset. If the itemset P occurrences are matched with the occurrences of another itemset Q then P is not a closed set. It mines all frequent itemsets, and checks whether any superset exists that has the same minimum support count as that of frequent item set, if found the itemset is not closed itemset else it is closed itemset. To evaluate the support of subsets, appropriate information is given.

Maximal frequent itemset:

Maximal frequent item sets are one of the forms of representing frequent itemsets. Therefore frequent itemsets are part of maximal frequent itemsets. An itemset X is frequent for which no items p can be added in order to remain{X,P} larger than minimum support threshold. Pruning of itemsets can be done if the minimum support threshold is not satisfied. It explains that frequent itemset is used to find transactions that are greater the minimum support set by the user. Closed frequent itemset, is where at least one transaction will have different itemsets and the support of this transaction is greater than the minimum support. When an item is added to an existing itemset, the Maximum Frequent Itemset indicates that the transactions fall below the minimum support.

2.2. Sequence Pattern Mining

Agrawal and Srikant[1995] first discussed sequential pattern mining and is illustrated as follows; they are utilized to examine data and need to detect frequent subsequences from an sequence database. The subsequences that are ever-present in the data are found as set of sequences [34]. In pattern mining, data will be like symbolic representation fixed up in a sequence and are applied in many fields like webpage streaming analysis, text mining, bioinformatics, and so on. The content of the database is updated incrementally in many other areas. In order to determine whole sequential patterns, the various mining algorithm necessitate to be competent even when the database changes as a few data sequences which are not common in the previous database may turn common in the database which are updated. Consequently, every time sequential patterns demand to check into updated database to bring forth sequential patterns and this extends to progressive mining algorithm of sequential patterns.

Rules for finding subsequences in a sequence:

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

- Let I = {I_1, I_2, ……, I_m} be a set of items. If 'I' has its subsets, it is called as itemset.
- Let S = {s_1,s_2, ……..,s_n} be a sequence of an ordered list of items, where s_i set of items and an element of that sequence is also referred to.
- Let a sequence α = <a_1, a_2, ……, a_n> be a subsequence of other sequence β = <b_1, b_2, ……, b_n> if there exists integers $1 \leq k_1 < k_2 < \ldots\ldots, K_n \leq m$, then $a_1 \subseteq b_{K1}$, $a_2 \subseteq b_{K2}$ ……, $a_n \subseteq b_{Kn}$, which can be clarified that if an element has several items, it can be similarly reduced if an element has only one object.
- Let D be a sequence database, D = {s1, s2, …, sm} and represent the set of tuples <sid, s> Where "sid" is a sequence id and s is a sequence. |D| denotes the sequences count in D.
- The support of a sequence α in D can be defined as the number of sequences in D which contains sequence α and is denoted as support (α).
- If the min_support is given as an integer to represent support threshold, and if support (α) > min_support, then sequence α is said to be a sequential pattern in D.Therefore, sequential pattern mining works to find all sequential patterns in D. The subsequences that come along in the sequence database are found out by sequential pattern mining and these sub-sequences are titled frequent sequential patterns [48]. In order to acknowledge count of a sequence that appears in the database, a minimum support need to be determined by the user.

2.3. Structure Pattern Mining

Structure pattern mining is meant for finding patterns which are in substructures within datasets based on three parameters.

- Minimum confidence
- Minimum frequency
- Minimum interest and maximum frequency

A substructure of structure pattern mining includes graphs and trees, the substructure data is considered as a labeled graph. The information structure inside a table does not matter; it's only about retrieving the data from rows and columns without altering the meaning of that data [37]. Graphs are capable of finding substructure data, a graph, sub graph, substructure is said to be frequent if the occurrence of them within a dataset should not be less than minimum support. Some of the examples of structured data are XML data, chemical data and web browsing data can be considered for structured graphs. The frequent structural pattern gives user an interior view of graphs.

2.4. Uncertain Itemset Mining

This part reports the expected support and the existential probability condition linked up along with the uncertain dataset. Within an unclear database transaction T = {i_1,i_2,. ,i_m} each and every item i_q had an existential probability as P(i_q, T), which conveys the probability that iq is present in T with a value $0 < P(i_q, T), \leq 1$.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

Definition 4: Existential Probability: In a transaction T, the existential probability P(A, T) of a pattern A is defined as the product of the associated probability values for all items i_q in A, so that the items are treated as individuals.

Definition 5: Expected Support: An itemset X's predicted support is the sum of X's existential probability for all the transactions inside the database. As the data is collected and stored from various sources, so uncertainty is present in various forms and this leads to low quality of results where accuracy is poor. Therefore, it's difficult to handle data uncertainty and so we focus on mining frequent patterns in uncertain Big Data which has large data sets. Uncertain Big Data Techniques classification is shown in figure 3 [80], where different models are used to handle uncertain data. Various types of data lead to uncertainty in big data that may show negative efficiency and accuracy of the outcomes. Suppose, a data which is used for training is somehow skewed, incomplete, the learning algorithm applied on the corrupted training data that has a possibility to produce inaccurate output results. So, it is difficult for big data analytic methods to deal uncertainty. In order to deal with several kinds of uncertainty, many theories and techniques have been produced to model the different types shown in Fig 3

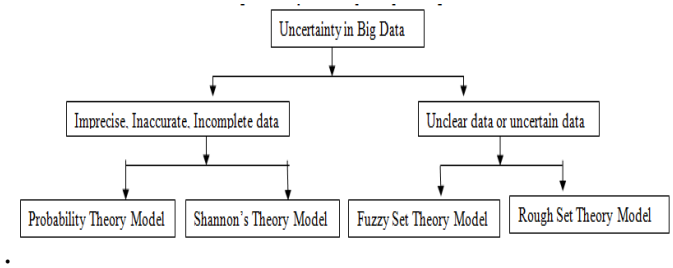

Figure 3: Uncertain Big Data Techniques classification

Probability theory deals with randomness and the statistical features of an input data [81]. The technique of fuzzy decisions is known to emulate human thinking in order to better resolve uncertainty in the real world. [82]. Shannon's entropy determines the amount of information in a variable in order to find out an amount of missing information in a average random source [83, 84]. Shannon implemented entropy that provides a method of information quantification when the weights of the parameters cannot be measured using a decision-maker [85]. To perform reasoning on vague data and deal with uncertain or incomplete data, rough set theory offers a mathematical tool [86]. As shown in figure 3, Probabilistic theory & Shannon's entropy are considered to model imprecise, inaccurate data and incomplete. Fuzzy and rough theories, on the other hand, are used to model vague or ambiguous data. [87]. While there are a number of techniques for analyzing big data, if uncertainty in the data or the methodology itself is overlooked, the accuracy of the analysis leads to negative outcomes. Uncertainty models, such as the theory of probability, fuzziness, rough set theory, etc. are used to increase big data analytic methods in providing more accurate outcome.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

Table 1: Performance of Uncertain Big Data Models

Uncertainty Model	Performance
Probability Theory	Efficient in handling data randomness and subjective uncertainty based on precision.
Shannon's Theory	Capable of handling complex data [89].
Fuzzy Set Theory	Handles indefinite and uncertain data in systems that are difficult to model. Precision not required, this model is easy to implement and interpret.[89]
Rough Set Theory	Deals with unclear or vague data.[90] Minimal information required to determine set membership. Only considers the information that is available from given data.[91]

Table 1 shows comparison and summarization of techniques between different uncertain strategies are identified. When "Frequent Pattern Mining is performed on Uncertain Big Data", the stored Big Data which is in unstructured format has to be processed into an understandable format before the data can be mined. Processing unstructured data includes searching the data, filtering the data, and then applying meaningful algorithms to obtain datasets with correct formats. When extracting useful information from Big Data more effective and efficient algorithms and approaches are needed. MapReduce programming model supports Big Data in performing parallel or distributed computing. Efficient algorithms can be proposed using MapReduce techniques to find frequent patterns on uncertain Big Data.

III. FREQUENT PATTERN MINING EARLY RESULTS AND APPLICATIONS

To perform frequent pattern mining different frameworks have been defined. The common point among all the frameworks is, itemsets with frequency above a given minimum support need to be found. But, these itemsets sometimes may often not reflect positive correlations in between items, as they do not normalize or fall under the exact frequencies of the items. The main reason that is behind the frequent pattern mining algorithms is computational challenge of a task. The reason is the search space needed by FPM for a medium sized dataset is extremely large, which is exponential to length of the transactions in the dataset. For frequent pattern mining, many variants of algorithms have been developed, many of which are closely related to each other, in reality the execution tree procedure of all the algorithms is mostly unique in terms of the order wherein the patterns are explored, however the counting of items of the candidate is independent of each other. The efficiency of these different methodologies depends on one another in a way that, the efficiency of pruning method may sometimes depend on the retrieving the candidate items. And, the efficiency in counting the items depends on the order of exploration as they are explored at higher levels can be used at the lower levels with imposing certain rules. Overall, all frequent pattern mining algorithms are considered as the complex versions of the simple standard or basic pseudo code. In general, from [75] the algorithms for frequent pattern mining can be categorized into 3 main categories; Join-Based, Tree-Based, and Pattern development as shown in following Figure 4. Using the bottom-up method, the Join-Based algorithms from Figure 4 classify frequent items in a data set and extend them into larger itemsets until those itemsets

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

fulfill more than the minimum user-defined support count in the database. By constructing a lexicographic tree that allows the items to be mined in a number of ways, the Tree-Based algorithms resolve the issue of frequent itemset formation. Finally, depending on the frequent patterns found, Pattern Growth algorithms apply the divide-and-conquer approach to partition databases and extend them into longer databases.

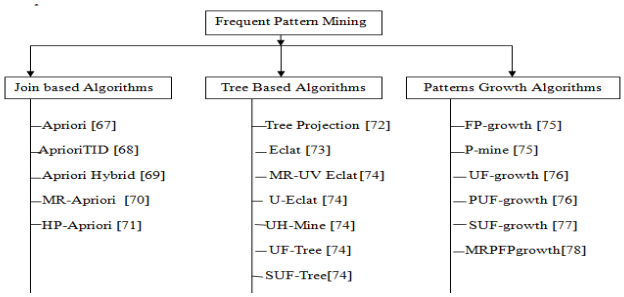

Figure 4: Taxonomy of Frequent Pattern Mining

Apriori Algorithm: [67] This algorithm mines frequent itemsets to produce association rules. Items are searched in an iterative manner to find (k+1) itemsets from the available K-itemsets. The database is checked to find all the frequent candidate 1-itemsets, 2-itemsets where the minimum support count should be met by each candidate itemset. The database is scanned until upto there are no more possible frequent K-itemsets. The itemsets which will satisfy the minimum support count are included into the next cycle for scanning. Apriori algorithm, with the minimum support count reduces the itemsets count significantly and hence performs well. When the count of K-itemsets increases, It is necessary to produce more numbers of candidate itemsets. As, the number of candidate itemsets generation is more, the database is needed to scan repeatedly and need to be verified. **FP-growth Algorithm: [75]** Frequent Pattern Growth algorithm is utilized to mine frequent itemsets without candidate generation process. It uses a strategy of divide and conquer to compact all frequent items into a tree which is called Frequent Pattern Tree. The Frequent Pattern Tree is again partitioned into a set of Conditional FP-Trees for each frequent item. This algorithm identifies long frequent patterns through conditional frequent pattern trees, thereby avoiding the generation of candidate itemsets. It reduces the search time for finding frequent patterns. If the size of dataset is large, constructing an FP-tree is a time consuming process.

EClaT (Equivalence Class Transformation) Algorithm: [73] This algorithm, mines frequent itemsets using vertical data format method i.e; transforming the data from the horizontal format into the vertical format. The vertical data format is < items, TID>. All transactions involving a specific set of items are grouped together in the same record. This method is repeated until no frequent item sets have been identified. The database is scanned only once in order to transform the data from horizontal format into vertical format. The database is not required to find the support count of every itemset, because the count of transactions resembles that itemsets are present in particular transaction. Storing transactions for each itemset consumes lot of memory

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

space and processing time. **Tree Projection Algorithm: [72]** this algorithm mines frequent itemsets by constructing lexicographic tree i.e; breadth first, depth first or it's a combination of both. The support count of each and every frequent itemset in every transaction is counted and put as a node on the lexicographic tree. The algorithm upgrades its performance by showing the transactions which contain a frequent itemset. The search for a frequent itemset is done in a top down approach. Matrix structure or lexicographic tree approach is used for identifying frequent itemsets that have very low support count. Different tree representations perform different in terms of efficiency at memory consumption. **COFI (Co-occurrence Frequent Itemset) Algorithm: [97]** this algorithm mines frequent itemsets by pruning method; thereby it decreases the memory usage space. The pruning method builds small tress from the frequent Pattern growth tree using top down approach. COFI algorithm performs better than FP-Growth algorithm in terms of memory consumption and execution time. If the minimum support count is too low, the algorithm performance degrades. **TM (Transaction Mapping) Algorithm: [98]** this algorithm mines frequent itemsets in vertical data format representation. In this algorithm, transactions ID of every itemset are mapped to another location list of transaction intervals. If the minimum support count is high, the transaction mapping method compresses the transaction ID's into continuous intervals. It performs better than FP-Growth and EClaT algorithm in terms of datasets that contains short frequent patterns. This algorithm is slow in processing speed when compared with FP-Growth algorithm. **P-Mine Algorithm:[99]** Using a parallel disk-based approach, this algorithm mines frequent item sets. It uses VLDBMine data structure and a Hybrid-Tree (HY-Tree) is used in the VLDBMine data structure. To store the data set and other information which is used for data retrieval process VLDBMine is used. In the VLDBMine data structure, the efficiency and scalability of frequent itemset mining is enhanced as the data set is represented. When the process of frequent itemset mining is performed simultaneously across various processor cores, the output on each node is optimized locally. This algorithm can only be reduced to the maximum level if there are several cores in the processor available. **LP-Growth (Linear Prefix Growth) Algorithm: [100]** LP-The growth algorithm mines frequent itemsets in a linear structure using arrays. By building a Linear Prefix Tree (LP-Tree), the method of data mining is simplified since it consists of arrays instead of pointers. This efficiency of memory use is improved by reducing the relation between different nodes. The Frequency Item Collection is stored in various nodes consisting of multiple arrays. All arrays are linked with array header where the first location in it indicates the parent array. This algorithm generates LP-Tree in a faster manner compared to that of the FP-Growth algorithm. As LP-Growth algorithm creates multiple nodes using array operations whereas the FP-Growth algorithm create the nodes one at a time. Traversing through the LP-Tree is done faster because the memory locations can be accessed directly using arrays. To insert a node into the LP-Tree the memory needs to be freed continuously. **Can_Mining (canonical- Order) Algorithm: [101]** Frequent itemsets are mined using this algorithm from a Canonical-order tree using incremental manner.The Canonical – Order Tree consists of a header table which has information about the items in the database. A collection of frequent items is needed for the algorithm to execute the mining operation in order

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

to extract frequent patterns from the Canonical-Order Tree. This algorithm performs well in terms of reducing the frequent itemsets mining, as the Canonical-order tree appends only the items that are frequent into the tree in a predefined order. It performs good than FP-Growth algorithm when the minimum support count is high. If the minimum support count is too low, this algorithm is not efficient, whereas FP-Growth algorithm perfoms efficently.

EXTRACT Algorithm:[102] EXTRACT algorithm mines frequent itemsets using "Galois lattice" which is a mathematical concept. This concept is partitioned into four functionalities; to calculate support count, to combine itemsets, removing the itemsets that are repeated, and in order to extract association rules from the frequent itemsets. This algorithm verifies the support count of each and every frequent 1-itemset which satisfies the minimum support. The frequent 1-itemset which does not satisfy the minimum support will be removed from the calculation. Then, In order to find all possible combinations of frequent itemsets by eliminating the duplicates, the remaining itemsets will also be combined. If all the unique frequent itemsets are found, the association rules that satisfy the minimum trust will produce the itemsets that do not satisfy the minimum trust will be excluded from the process of rule discovery. For mining many itemsets this algorithm outperforms the apriori algorithm. Once mining of frequent itemsets is performed they are not stored in any database. When dataset changes the new frequent itemsets need to be mined again. There are several progressive mining algorithms; to the highest degree it follows apriori property, which brings forth large set of candidate sequences even when the size of the database is large. To bring down candidate sequence a different approach called pattern growth is suggested to mine sequential patterns [35, 79]. The subsequences that come along in the sequence database are found out by sequential pattern mining and these sub-sequences are titled frequent sequential patterns [48, 49, 50]. In order to acknowledge count of a sequence that appears in the database, a minimum support need to be determined by the user. To find subsequences which are frequent among a set of sequences called as frequent subsequence. The appearance of these patterns should not be lower than the minimum support count stated by the user.

GSP (Generalized Sequential Patterns algorithm [51]): This follows the apriori property, which scans database multiple times in order to generate candidate itemsets which satisfy minimum support count or itemsets with more than the support count. This process is continued until no new candidate items or itemsets are found. Thus the apriori property generates more number of candidate sequences and checks each sequence with user defined minimum support count. **SPADE (Sequential Pattern Discovery with equivalence classes [52]):** This reduces the search space within the database as the sequence patterns are aggregated into equivalent classes. The data in the database is organized in vertical format and thereby it reduces the search space. **SPAM (Sequential Pattern Mining [53]):** This algorithm works only when the sequence data items are loaded into memory from the database. The candidate items that are generated using this algorithm are represented in a tree format both for adding an item into another transaction and into same transaction. **PrefixSpan (Prefix projected sequential pattern mining [54]):**

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

Finds sequence patterns using pattern growth approach but explores all the prefix subsequences instead of frequent subsequences. Then only prefix subsequences are found instead of finding all possible frequent subsequences. **LAPIN (Last Position Induction Algorithm [55]):** when the sequences are long, this algorithm reduces the search space. **CM-SPAM and CM-SPADE (Co-Occurrence MAP- SPAM and SPADE [56]):** This algorithm is an extension of SPAM and SPADE algorithms to which a new structure Co-Occurrence MAP is added. The Co-Occurrence MAP is used to store co-occurrence information of substructures. The closed sequential pattern does not include the any other sequential pattern which has the same support. **CloSpan (Closed Sequence Patterns [57]):** this algorithm explores on mining frequent closed sequences instead of frequent sequences. The algorithm works by dividing into two stages, firstly all frequent sequences are generated and secondly it removes the non-closed sequences. It mines long sequences by reducing time and space. **BIDE+ (Bidirectional Extension [58]):** is an extension of BIDE algorithm, this algorithm extracts closed sequence patterns and overcomes the problem of more candidate items. The BIDE algorithm consumes more memory where as this is overcome in BIDE+. **ClaSP [59]:** It mines the closed frequent sequence patterns in vertical database format. This algorithm works on two phases; in first phase it generates a subset of frequent sequences from main memory and in second step it eliminates all non- closed sequences from frequent sequences to obtain frequent closed sequences. **CM-Clasp [60]:** It is a modification for ClaSP, it is faster than ClaSP. It takes input as sequence database with user specified minimum support. A maximal sequential pattern generates many sequences, where users feel difficult to analyze and understand. **MaxSP (Maximal Sequential Pattern [61])**: this is an extension for PrefixSpan algorithm. It extracts maximal sequence patterns without any candidate items thereby removing the duplicate sequence patterns as it is a time consuming process and needs lots of storage space. **VMSP (Vertical Maximal Sequence Patterns [62])**: this pattern is similar to SPAM which searches for candidate items having similar prefix in a recursive order. Compressing Sequential Pattern algorithms are meant for reducing duplicity of itemsets and hence reduce the time of extraction.

GoKrimp and SeqKrimp [63] are the algorithms which explore patterns that are compressed and remove the space occupied by the candidate itemsets. GoKrimp removes various tests on candidate itemsets and performs faster than SeqKrimp. When mining is performed on Patterns, acquiring the required and necessary patterns based on minimum support is a bit problematic and time consuming. In order to overcome this problem Top-k Sequential pattern mining algorithms are used. **TSP (Top-K Closed Sequential Patterns [64])** uses pattern growth method to find patterns and minimum length constraint is applied to reduce the search space. TKS (Top-K Sequential Pattern [64]) are introduced to rectify the mentioned problem. This algorithm uses the concepts of PrefixSpan algorithm, to increase the size of patterns by performing multitasks mining. TKS algorithm uses the concept of SPAM by exploring the patterns and changing it into top-k algorithm. This algorithm finds patterns with high minimum support and removes the patterns with low minimum support. However FCloSM and FGenSM [67] are more efficient

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

algorithms for solving the problem such as to find a sequential pattern which generates the most utility pattern. Sequential pattern mining, works efficiently when patterns are discovered in large amounts and to retrieve desired patterns it consumes lot of time for retrieval [36]. Table 2, explains applications of frequent pattern mining algorithms. It explains about the scope of frequent pattern mining algorithms to be applicable in different aspects irrespective of the applications. Based on the gaps that are present in mining frequent patterns on traditional there are many frequent pattern algorithms that are used to mine frequent patterns on big data. Table 3, explains about the results of frequent pattern mining algorithms.

Table: 2 Applications of Frequent Pattern Mining Algorithms

YEAR	APPLICATIONS	AUTHOR
2011	Finding FPM on uncertain data in Bioinformatics Applications	Leung,C,k,. [21]
2013	No.of nodes in MapReduce need to be found, schedule computation on available mining resources	Leung,C,k,. [25]
2014	How to handle constraints that are not Anti-monotone	Leung,C,k,. [15]
2014	To Mine uncertain Big data for Frequent patterns based on user constraints	Leung,C,k,. [27]
2016	How to handle constraints that are neither monotone or Antimonotone,	Leung,C,k,. [26]

IV. FREQUENT PATTERN MINING IN UNCERTAIN BIG DATA, REUSLTS AND APPLICATIONS

Mining frequent patterns on big data plays a crucial role, as data is collected in various forms. This impacts the mining process, accessing speed, number of database scans and scalability. So this process divides the tasks into subtasks and the data operation is performed on each subtask separately and the local results produced by every subtask are aggregated to test the final result [108]. The partition procedure asks that mapper main memory would contain candidate k-itemsets such that, the candidate itemsets which are large in volume cannot cope with datasets that are less in volume. Load balance is essential, as it increases the performance of various computing techniques and allows distribution of resources based on user task. Load balance splits up the task between the nodes that are within the cluster to finish the task quickly. The various ways of dividing the task affect the load balance, as the tasks need to be assigned to a particular processor to complete its execution. This task distribution to each processing node, if it is done perfectly, will minimize the imbalance. Two significant troubles are handled by big data mining. The first issue is the data propagation frequency is more when compared with that of a

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

machine's available memory, and the second one is to calculate the frequent patterns in such large data. To address these issues parallel processing techniques can be considered, as algorithms utilize multiple processing units, along with quick working possibility for computationally more time consuming applications. The parallel computing is the headstone for performance betterment during the situation where the data set may fit into your computer memory [96,106], it may be viable that in-between data as candidate itemsets or patterns or data structures utilized for frequent pattern mining might not adjust into the memory.

Table 3: Pattern Mining Algorithms Results

AU Algorithm A Authors	Algorithm	Results
Agarwal and srikanth	AprioriAll(1995)	Generates lots of candidate sequences & takes time to prune them.[65]
Srikanth and Agarwal	GSP (1996)	It generates candidate sequences which satisfy minimum support count or itemsets with more than the support count until no new candidate itemsets are found and checks each sequence with user defined minimum support count.[66]
Zaki	SPADE (2001)	Huge set of candidates are generated and requires multiple database scan.
Pei et al.	PrefixSpan (2001)	To scan the database repeatedly it costs more in terms of runtime.[48]
Ayers	SPAM (2002)	It works only when the sequence data items are loaded into memory from the database. [53]
Yang et al.	LAPIN(2007)	when the Sequences are long, this algorithm reduces the search space.[55]
Yan et al Wang and Han Bac Le	CloSpan(2003) BIDE FCloSM and FGenSM(2017)	Avoids the unnecessary scanning of search space. It consumes more memory in extracting closed sequential patterns It works faster in discovering frequent closed sequences like CloSpan, BIDE, ClaSP and CM-ClaSP by consuming much less memory.

Many pattern mining algorithms have been developed but many of them are not capable in

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

working with the type of data that is present today which is Big Data. So, scalable parallel algorithms solve this problem based on three considerations; memory scalability, work partitioning, and load balancing. Computational or scalable parallelism is necessary tool for handling huge amounts of data along with working data on a single machine it allows to speedup applications computationally. When designing frequent pattern mining algorithms for Big Data;

- Memory scalability is necessary as the applications need to cope with the datasets which are large in size thus by increasing the parallelism.
- Decomposition of a problem into group of tasks where each of it indicates a unit of work, so that the tasks can executed independently and concurrently.
- Equal amount of work need to be assigned to all processor such that all processes complete the computation at an equal time.

A Serial Algorithm can be made as parallel algorithm if the memory constraints are not considered. A Parallel Algorithm which is designed works for shared memory systems and distributed memory systems. A shared memory system has their own set of constraints or consequences like hardware cache size and concurrent memory access. In distributed memory system, processors have access to a private local memory address space. In, distributed memory systems two programming frameworks Message Passing and MapReduce are used. Message Passing is used in scientific computing community and MapReduce is designed particularly to work with Big Data. MapReduce is one of the programming models for distributed memory systems that provide a simple technique of writing parallel programs. In [12] an algorithm named P**rivacy preserving item centric for mining frequent patterns in Big Uncertain Data**" inside Apache Spark environment is proposed. Both privacy and security is imposed on big data and the algorithm explains item centric mining, frequent pattern mining, big data mining, uncertain mining, privacy preserving mining the performance of this algorithm in terms of finding frequent patterns in uncertain big data is effective but more extensive evaluation and different ways need to be used to impose the privacy-preserving mining on intermediate processing so as to speed up the privacy-preserving frequent pattern further. **Big data mining for interesting patterns with MapReduce Technique** [13] is used to mine frequent patterns in uncertain data with user specified constraints using Map Reduce technique, where user constraints refer to minimum support count and constraint on attribute. A Map Reduce model is used to discover finds valid single tone and valid non-single tone patterns from uncertain database based on both valid single tone and non-single tone items it finds the valid frequent pattern from uncertain data which satisfy user constraints. The Map Reduce model helps to find frequent patterns based on user specified constraints in the uncertain big data with respect to time and space effectively. **Uncertain Big data Strategical Miner** [14], **Reduce search space for Big data mining for finding interesting patterns from uncertain data** [15], these papers explains that there are situations in which data are uncertain, however the Items in probabilistic data-bases of uncertain

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

data are usually associated with existential probabilities expressing the chances of these items to be present in the transaction, this leads to the situation where mining from uncertain data is much larger than mining from precise data. In some applications, users may be interested in only a tiny portion of this large search space. Hence to overcome wasting lots of time and space, a tree-based algorithm that allows users to specify their constraints in terms of only anti-monotone (AM) constraints and MapReduce is used to mine uncertain Big data for frequent patterns that satisfy the user-specified constraints and therefore this algorithm returns only those patterns that are user specified constraints. Algorithms like UFgrowth, CUF-growth, PUF-growth, tubeS-growth, tubePgrowth are considered to find frequent patterns from uncertain data is discussed in **Algorithms to mine frequent pattern from uncertain data** [16] and MR-growth algorithm is used to mine frequent patterns on Big Data but Big Data tools can be considered to mine frequent patterns of uncertain data that can be done further. **Big data mining for interesting patterns from uncertain data using with MapReduce Technique** [17] is used to mine frequent patterns in uncertain data with user specified constraints using Map Reduce technique, where user constraints refer to minimum support count and constraint on attribute. A Map Reduce model is used to discover finds valid single tone and valid non-single tone patterns from uncertain database based on both valid single tone and non-single tone items it finds the valid frequent pattern from uncertain data which satisfy user constraints. The Map Reduce model helps to find frequent patterns based on user specified constraints in the uncertain big data with respect to time and space effectively. **Reducing the search space for interesting patterns from uncertain data** [18] uses a Map Reduce model with apriori algorithm that mines uncertain Big Datasets has been proposed and this algorithm works efficiently in returning patterns that are user constraint and it handles an Anti-monotone constraint that shows better performance. **Item centric mining of frequent patterns from big uncertain data** [19] uses an algorithm that works in Apache Spark environment, the performance of this algorithm in terms of finding frequent patterns in uncertain big data is effective but more extensive evaluation and different ways need to be used to explore more functional features to be provided to the users. Authors of [20] **On efficient mining of frequent itemsets from big uncertain database** explains, the usual technique used to find frequent itemsets from uncertain databases is known as Possible Word Semantics (PWS). As the size of data base increases PWS has performance issues. To overcome these issues three techniques are used; 3D linked array based strategy, connected tree techniques, average probability based setup. In 3D linked array strategy it scans the database only once and stores the support information of the item and its association with other items in a 3D array. The tree-based method uses 1D array which is associated with each node of the tree, stores information about items in the database and their associations with other items. The average probability approach computes the average probability factor and uses it to map the uncertain database to a tree. The three techniques are compared with four algorithms that are used for uncertain data, mining threshold-based (MB) technique, frequent itemsets using nodesets (FIN), prepost+, and uncertain apriori (UApriori) and the performance results shows better overall large uncertain databases by consuming 60% less time as compared to the other four probabilistic

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

frequent itemsets mining methods. Analysis on algorithms like U-Apriori, UH-mine, UF-Tree, U-Eclat, U-FPS, U-FIC, UF-Growth and proposed SUF-growth, SUF-tree algorithms to mine uncertain data for items that are likely to be frequent is done in **Mining uncertain Data** [21] algorithms have been proposed to mine uncertain data for items that are highly likely to be frequent, multi-item patterns that are highly likely to be frequent, as well as association rules that are highly likely to be interesting and these algorithms are used to mine uncertain Data for frequent sequences and frequent graphs and to perform mining uncertain data in Bio informatics applications. **Tightening upper bounds to the expected support for uncertain frequent pattern mining [22],** A compact tree structure is proposed in finding uncertain data and a technique is used to tighten the upper bound to find expected support count in the tree structure and frequent patterns are mined based on this tightened bounds and the performance of the algorithm leads to significantly low number of false positives because of the tightness of the upper bound. **Approximation to expected support of frequent itemsets in mining probabilistic sets of uncertain data [23],** conveys that users mine frequent itemsets from probabilistic sets of uncertain data where users are uncertain about number of items in transactions. Each item in these probabilistic sets of uncertain data is often associated with an existential probability expressing the items present in that transaction. To mine frequent itemsets from these probabilistic datasets, many existing algorithms capture lots of information to compute expected support. In this paper, several upper bound values are examined and allows user to use upper bound consumes less space for finding frequent item sets in mining uncertain data and the performance of these algorithms provide upper bounds support count up to 3+ frequent itemsets. Hybrid Apriori algorithm to find frequent patterns, **Mining Frequent Patterns from Big Data Sets using Genetic Algorithm** [24], usually data mining methods tend to be slow and are considered best when they yield precise results. In this paper hybrid Apriori algorithm is used to generate frequent patterns and then the association rule generated by the Apriori algorithm is optimized using genetic algorithm. To generate strong association rules, Genetic Algorithm operators like selection, crossover and mutation have been applied on association rule generated by Apriori algorithm. A parallel algorithm has been proposed to mine the frequent patterns with a user specified minimum support. The work is shared among n number of processors to compute frequent item sets. So there will be communication between the processors. The time taken to finish the task is very less when compared to other algorithms. A tree-based algorithm (MR-growth) that uses MapReduce to mine frequent patterns from big uncertain data some enhancements are proposed to further improve its performance, **Mining Frequent Patterns from Uncertain Data with MapReduce for Big Data Analytics [25].** MapReduce programming model is used for mining frequent patterns from uncertain data to perform big data analytics. The use of MapReduce yields significant speedups to our MR-growth algorithm. The performance results demonstrates the MapReduce programming model works efficiently for mining frequent patterns from uncertain data for Big data analytics and evaluating the number of nodes in the MapReduce environment during runtime need to be processed. Enhancements in evaluating the number of nodes in the MapReduce during run time need to be

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

found and an extension of MapReduce, which would allow us to recursively build sub-trees and schedule their mining on available computation resources. **Finding efficiencies in frequent pattern mining from big uncertain data [26],** to avoid wasting lots of time and space in computing all frequent patterns and removing uninteresting ones, a tree-based algorithm is proposed that allows users to express their interest in terms of succinct anti-monotone (SAM) constraints, anti-monotone (AM) constraints, and monotone constraints. The algorithm uses MapReduce to mine uncertain big data for frequent patterns that satisfy the user-specified constraints. An efficient partitioning algorithm is proposed to achieve load balanced workloads within the MapReduce framework for big data frequent pattern mining. Enhancements are done in exploring how to handle constraints that are neither monotone nor AM. **A Data Science Solution for Mining Interesting Patterns from Uncertain Big Data** [27] focus on reducing computation and mining performance, a data science solution for finding frequent patterns based on user constraints using MapReduce framework is presented. The user constraints are expressed in Anti-Monotone constraints and the performance of proposed algorithm shows BigAnt as an efficient data science solution in mining interesting patterns from uncertain Big Data. **Fast implementation of pattern mining algorithms with time stamp uncertainties and temporal constraints [103]** an approach is used to write efficient codes for pattern mining problems. The approach includes: (1) cleaning datasets with removal of infrequent events, (2) presenting a new scheme for time-series data storage, (3) exploiting the presence of prior information about a dataset when available, (4) utilizing vectorisation and multicore parallelisation. We present two new algorithms, FARPAM (FAst Robust PAttern Mining) and FARPAM (FARPAM with prior information about prior uncertainty, allowing faster searching). FARPAM has been successfully tested on a publicly available weather dataset and on a confidential adult social care dataset, reproducing results obtained by previous algorithms in both cases. It has been profiled against the widely used SPAM algorithm (for sequential pattern mining) and RobustSPAM (developed for datasets with errors in time points). The algorithm outperforms SPAM by up to 20 times and RobustSPAM by up to 6000 times. In both cases the new algorithm has better scalability. **An Efficient algorithm for frequent pattern mining over uncertain data stream [104],** this paper explains that many algorithms of uncertain data stream mining are based on tree structure. To find frequent patterns all branches in the tree need to be retrieved which is a time consuming process. Therefore a new structure uncertain Item-lists (UIT-lists) and uncertain frequent stream mining algorithms (UFS) is proposed. A temporary array is used to store patterns TID and to get TIDs combination of patterns. UIT reduces mining time and memory usage is less. **Efficiently Predicting Frequent Patterns over Uncertain Data Streams [105],** Authors convey that a forecast method for predicting frequent patterns with hidden Markov models is suggested to effectively find frequent patterns over unknown data sources. The method comprises of a construction and development phase. The hidden Markov model is built in the former stage by the number of previous data, and the results of testing and adjustments of transformation probabilities occur in the latter stage. Frequent patterns can be easily obtained by building models of patterns. Experimental findings show that the proposed method can save time when

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

exploring frequent patterns and provide approximately 72 % average accuracy in 50 sequential predictions; thus, frequent patterns over unknown data streams can be predicted accurately and efficiently. Table 4, shows the serial and parallel frequent pattern mining algorithms which works for Big Data. Serial type indicates, the algorithms that work under memory constraints where as parallel type indicate the algorithms which doesn't have any memory constraints. Based on the outline of finding frequent patterns in uncertain big data shown in Table 5, existing algorithms can be used in finding constraint frequent item set mining to reduce the uncertain values and imprecise values. In this section, classification of algorithms for frequent pattern mining on standard datasets and Big Data is shown. To perform frequent pattern mining on uncertain Big Data and algorithms that are listed may be or may not be used but that should be able to work in parallel environment satisfying the considerations like memory scalability, work partitioning, and load balancing. There will be situations where existing algorithms will not come up with new constraints, new technologies which are emerging and with framework that are existing.

Table 4: Frequent Pattern Mining Algorithms on Big Data

TYPE	ALGORITHM
Serial	Apriori
Serial	ECLAT (Equivalence Class Transformation)
Serial	FP-growth (Frequent Pattern Growth)
Serial	Partition
Serial	SEAR (Sequential efficient Association Rules)
Serial	TreeProjection
parallel	BigFIM (Frequent Itemset Mining for Big Data)
parallel	FP-growth (Frequent Pattern Growth)
parallel	CD (Count Distribution)
parallel	CCPD (Common Candidate Partitioned Database)
parallel	CD TreeProjection (Count Distributed Tree Projection)
parallel	DD (Data Distribution)
parallel	Dist-Eclat (Distributed Eclat)
parallel	DPC (Dynamic Passes combined counting)
parallel	FPC (Fixed Passes combined counting)
parallel	HPA (Hash Partitioned Apriori)

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

parallel	ParEclat (ParallelEclat)
parallel	SPC (Single pass counting)

Table 5: Mining uncertain Big data Frequent Patterns

YEAR	SCOPE/AREA OF RESEARCH	AUTHOR
2011	Finding FPM on uncertain data in Bioinformatics Applications	Leung,C,k,. [21]
2013	No.of nodes in MapReduce need to be found, schedule computation on available mining resources	Leung,C,k,. [25]
2014	How to handle constraints that are not Anti-monotone	Leung,C,k,. [15]
2016	How to handle constraints that are neither monotone or Anti-monotone,	Leung,C,k,. [26]
2016	use Big Data tools to find FPM in uncertain Big Data	Vani [16]
2016	Use Hadoop Framework for finding FPM from Big uncertain data in Apache Spark	D.Kumari [17]
2018	Privacy Preserving frequent pattern mining	Leung,C,k,. [12]
2018	provide extensive evaluation for finding FP from big uncertain data in Apache Spark	Peter Brauna [19]
2018	Algorithms need to handle tuple deletion & updation done to database	Ahsan Shah [20]

V. OPEN ISSUES AND CHALLENGES

There are many pattern mining techniques that have been proposed but still there remain many challenges.

Complex Types of Data and Patterns: Most mining algorithms work on data that has come from different sources, such that the data will be available in different forms. Yet, complex data which are in different types in different diligences still need to be studied in uncertain data. This often can take place when information collected from multiple sources is mined. Future challenge is to develop more flexible techniques to handle a lot of complicated data types in any parallel and distributed computing.

Scalability: Scalability plays significant role in dealing with big data problems. When the computing devices need scalability then complexities occur. Therefore a proper parallel algorithm can be developed such that the systems have an ability to adapt to the changes in terms of workload such that it automatically allocates and removes the required resources to perform a

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

task.

Load balancing: Load balance is essential, as it increases the performance of various computing techniques and allows distribution of resources based on user task. Load balance splits up the task between the nodes that are within the cluster to finish the task quickly. The various ways of partitioning the task affects the load balance, as the tasks need to be assigned to a particular processor to complete its execution. This distribution of task to each and every processing node would decrease the imbalance if it is done perfectly. The dynamic load balancing addresses this instance in such a way that tasks are assigned to processors when they are idle but difficult in finding or monitoring the resources required for each job so that they can be divided among the processors that are computing. Applying frequent pattern mining on

huge data which is uncertain is a big challenge in recent trends. Algorithms that are existing deals with mining user constraint frequent patterns on uncertainty based Big Data using Map Reduce Model are inappropriate sometimes. However, there are situations where the nodes of Map Reduce model need to be reduced during runtime however, use of Big Data tools can be considered for finding frequent patterns in uncertain Big Data for handling real-time applications.

VI. CONCLUSION

The term Uncertain Data refers to the data which is imprecise, data changed from correct and original values. There are many traditional algorithms for pattern extraction but finds difficult during large-scale data processing; however uncertainty is present in various forms and this leads to low quality of results where accuracy is poor. Therefore, it's difficult to handle uncertainty data and so we focus on uncertain datasets by surveying parallel and distributed algorithms that are used for frequent pattern mining on uncertain data such that Parallel and distributed mining have turned out to solve this issue and some of the parallel and distributed pattern mining challenges are discussed.

REFERENCES

1. Aggarwal, C. C.,On Density Based Transforms for Uncertain Data Mining, IEEE 23rd International Conference on Data Engineering2007, Volume: 1, pp.866-875.
2. Burdick, D., and Deshpande, P., OLAP Over Uncertain and Imprecise Data, Proceedings of the 31st VLDB Conference, Norway 2005, pp.970-981.
3. Sarma, A, D., Benjelloun, O., Halevy, and A., Widom, J., Working Models for Uncertain Data, Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), 2006, IEEE Computer Society, DOI: 10.1109/ICDE.2006.174, pp.1-22.
4. Evfimievski, A, V., Srikant, R., Agrawal, R., and Gehrke, J., Privacy preserving mining of association rules, In Proceedings of Eigth ACM, SIGKDD International Conference on Knowledge Discovery & Data Mining, 2002, pp.217-228.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

5. Molina, G, H., and Porter, D., The Management of Probabilistic Data, IEEE Transactions on Knowledge and Data Engineering, Vol. 4, 1992, pp.487–501.
6. Kriegel, H, P., and Pfeifle,M., Density-Based Clustering of Uncertain Data, In Proceedings of Eleventh ACM, SIGKDD International Conference on Knowledge Discovery & Data Mining, 2005, pp.672-677.
7. Kriegel, H, P., and Pfeifle,M., Hierarchical Density Based Clustering of Uncertain Data, 5th IEEE International Conference on Data Mining (ICDM'05), IEEE Computer Society, pp.1-4.
8. Dalvi, N., and Suciu, D., Management of Probabilistic Data Foundations and Challenges June 11–14, Beijing, China, Copyright 2007 from ACM, pp.1-12.
9. Nilesh N. D. and Suciu, D. Efficient Query Evaluation on Probabilistic Databases, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004, pp.864-875.
10. Cheng, R., Kalashnikov, D., and Prabhakar, S., Evaluating Probabilistic Queries over Imprecise Data, Proceedings of the ACM SIGMOD International Conference on Management of Data, June 2003, pp.551-562.
11. Cheng, R., Kalashnikov, D., and Prabhakar, S. Querying Imprecise Data in Moving Object Environments, IEEE Transactions on Knowledge and Data Engineering, vol. 9, 2004, pp.1112-1127.
12. Leung, C, K., Hoi, C, S, H., Pazdor, A, G, M., and Wodi, B, H., Privacy-Preserving Frequent Pattern Mining from Big Uncertain Data, IEEE International Conference on Big Data, 2018, pp.5101-5110.
13. Jamdar, N., and Lakshmi, A, V., Big Data Mining for Interesting Patterns with MapReduce Technique, Innovare Academic Sciences Pvt Ltd, special issue - April 2017, Asian Journal of Pharmaceutical and Clinical Research, pp.1-3, DOI https://doi.org/10.22159/ajpcr.2017.v10s1.19634
14. Sapte, H, V., and Pallati, S, S., Uncertain Big Data Strategical Miner, Volume-5, Issue-6, IJCSE-2017, pp.237-243.
15. Leung, C, K., MacKinnon, R, K., and Fan Jiang, Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data, IEEE International Congress on Big Data, 2014, DOI 10.1109/BigData.Congress.2014.53, pp.315-322.
16. Bhogadhi, V., and Chandak, M, B., Overview of Important Algorithms to mine Frequent Patterns from Uncertain Data, Vol-2, Issue-5, International Journal of Advanced Engineering, Management and Science, 2016, pp.323-329.
17. Kumari, D., Leena, H, P., and Thakur, U, K., Big Data Mining for Interesting Patterns from Uncertain Data using Map Reduce Technique, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 6, 2016, pp.476-483.
18. Meiappane, A., and Nilavazhagan, K., Reducing the search Space for Interesting Patterns from Uncertain Data, International Research Journal of Engineering and Technology, Volume: 03, 2016, pp.219-225.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

19. Brauna, P., Alfredo, Leung, C, K., Pazdora, A, G, M., and Souzaa, J., Item-centric mining f frequent patterns from big uncertain data, 22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2018-Elsevier, Procedia Computer Science 126, pp.1891–1900.
20. Shah, A., and Halim, Z., On Efficient Mining of Frequent Itemsets from Big Uncertain Databases, J Grid Computing, Springer-2018, pp.1-20.
21. Leung, C, K., Mining uncertain data, Vol.1, WIREs(Wiley Interdisciplinary Reviews) Data Mining Knowledge Discovery, pp.316-329, DOI: 10.1002/widm.31.
22. Leung, C, K., MacKinnon, R, K., and Tanbeer, S., Tightening upper bounds to the expected support for uncertain frequent pattern mining, 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2014-Elsevier, Procedia Computer Science, pp.328 – 337.
23. Cuzzocrea, A., Leung, C, K., and MacKinnon, R, K., Approximation to expected support of frequent item sets in mining probabilistic sets of uncertain data, 19th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2015-Elsevier, Procedia Computer Science, pp. 613 – 622.
24. Babi,C., Rao, M, V., and Rao, V, V., Mining Frequent Patterns from Big Data Sets using Genetic Algorithm, International Journal of Engineering Research and Technology, Volume 11, 2018, pp. 287-306.
25. Leung, C, K., and Hayduk, Y., Mining Frequent Patterns from Uncertain Data with Map Reduce for Big Data Analytics, Springer-Verlag Berlin Heidelberg 2013, pp. 440–455.
26. Leung, C, K., MacKinnon, R, K., and Fan Jiang, Finding efficiencies in frequent pattern mining from big uncertain data, Springer Science Business Media New York 2016, pp.571-594,vol.20,2017, DOI 10.1007/s11280-016-0411-3
27. Leung, C, K., and Fan Jiang, A Data Science Solution for Mining Interesting Patterns from Uncertain Big Data, 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, IEEE Computer Society, pp.1-8, DOI 10.1109/BDCloud.2014.136.
28. Yassine, A., Singh, S., and Alamri, A., Mining Human Activity Patterns From Smart Home Big Data for Health Care Application, 2017, IEEE Access 5, pp. 13131–13141.doi:10.1109/access.2017.2719921
29. Zou, Z., Li, J., Gao, H., and Zhan, S., Frequent Sub graph Pattern Mining on Uncertain Graph Data, Proceedings of the 18th ACM Conference on Information and Knowledge Management,2009, pp.583-592.
30. Jiménez, M., Triguero,I. and John, R., Handling uncertainty in citizen science data: Towards an improved amateur-based large-scale classification, Information Sciences, 2019, pp.301-320.
31. Mohamed M., Zaghdoud, M. and Akaichi, J., A New Framework of Frequent Uncertain Subgraph Mining, Procedia Computer Science- 2018, pp.413-422.
32. Li, Y., and Beaubouef, T., Data Mining: Concepts, Background and Methods of Integrating Uncertainty in Data Mining, pp.1-6, 2010

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

33. Chau, M., Cheng, R., Kao, B., and Ng, J., Uncertain Data Mining: An Example in Clustering Location Data, Lecture Notes in Computer Science, pp.199–204. doi:10.1007/11731139_24
34. Mooney, C. H., and Roddick, J. F., Sequential pattern mining approaches and algorithms, ACM Computing Surveys, 45(2), pp.1–39, 2013,doi:10.1145/2431211.2431218
35. MABROUKEH, N.R., and EZEIFE, C, I., A Taxonomy of Sequential Pattern Mining Algorithms, ACM Computing Surveys, Vol. 43, No. 1, Article 3, 2010, pp.3:0-3:41.
36. Antunes,C. and Oliveira,A.L., Sequential Pattern Mining with Approximated Constraints, pp.1-8, 2004
37. Nijssen, S., and Kok, J. N., A quick start in frequent structure mining can make a difference, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-2004, pp.647-652, doi:10.1145/1014052.1014134
38. R. Goldman, and J. Widom, DataGuides: Enabling Query Formulation and Optimization in Semi structured Databases, Proceedings of the 23rd International Conference on Very Large Data Bases-1997, pp.436-445.
39. Chen, M, S., Park, J, S., and Yu, P, S., Efficient Data Mining for Path Traversal Patterns, IEEE Transactions on Knowledge and Data Engineering, Volume: 10 , Issue: 2 , Mar/Apr 1998, pp.1-22.
40. Lin, X., Liu, C., Zhang, Y., and Zhou, X., Efficiently computing frequent tree-like topology patterns in a web environment, 99th Proceedings of the 31st International Conference on Technology of Object Oriented Language and Systems, IEEE Computer Society, 1999, pp. 440-446.
41. Nanopoulos, A., and Manolopoulos, Y., Finding Generalized Path Patterns for Web Log Data Mining. Lecture Notes in Computer Science, pp.215–228, 2000,doi:10.1007/3-540-44472-6_17
42. Ke Wang, and Huiqing Liu, Discovering structural association of semi structured data, IEEE Transactions on Knowledge and Data Engineering- 2000, pp.353–371.doi:10.1109/69.846290
43. Kuramochi, M., and Karypis, G., Frequent sub graph discovery, Proceedings 2001 IEEE International Conference on Data Mining, pp.313-320. doi:10.1109/icdm.2001.989534
44. Yan, X., and Han, J., gSpan: graph- based substructure pattern mining, IEEE International Conference on Data Mining -2002, pp.721-724.
45. Aggarwal, C, C., and Han, J., Frequent Pattern Mining, Springer International Publishing Switzerland 2014, pp.1-85, DOI 10.1007/978-3-319-07821-2
46. Leung, C. K. S., Hao, B.,and Jiang, F., Constrained frequent item set mining from uncertain data streams, 2010 IEEE 26th International Conference on Data Engineering Workshops, doi:10.1109/icdew.2010.5452736
47. Bonchi, F., and Lucchese, C., Pushing Tougher Constraints in Frequent Pattern Mining, Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2005, pp. 114-124.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

48. Viger,P,F., Lin,J,C., Kiran,R,U., Koh,Y,S., A Survey of Sequential Pattern Mining, Data Science and Pattern Recognition, Volume 1, Number 1, February 2017
49. Slimani,T., an d Lazzez,A., SEQUENTIAL MINING: PATTERNS AND ALGORITHMS ANALYSIS, publishes in ArXiv,2013,pp.1-10.
50. Rjeily,C,B., Badr,G., El Hassani,A,H., Andres,E., Sequential Mining Classification, International Conference on Computer and Applications (ICCA),2017
51. R. Srikant, and R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, In International Conference on Extending Database Technology, pp.1-17, Springer Berlin Heidelberg, 1996.
52. M.J. Zaki, SPADE: An efficient algorithm for mining frequent sequences, Machine learning, 42(1-2), pp.31-60, 2001.
53. J. Ayres, J. Flannick, J. Gehrke, J. and T. Yiu, Sequential pattern mining using a bitmap representation, In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 429-435,2002.
54. J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C.Hsu, Prefixspan: Mining sequential patterns efficiently by prefix projected pattern growth, In proceedings of the 17th international conference on data engineering, pp. 215-224, 2001.
55. Z. Yang, Y. Wang, and M. Kitsuregawa, M., LAPIN: effective sequential pattern mining algorithms by last position induction for dense databases, In International Conference on Database systems for advanced applications (pp. 1020-1023), Springer Berlin Heidelberg, 2007.
56. P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, Fast vertical mining of sequential patterns using co-occurrence information, In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 40-52), Springer International Publishing, 2014.
57. X. Yan, J. Han, R. Afshar R., CloSpan: Mining Closed Sequential Patterns in Large Datasets, Proceedings of the 2003 SIAM International Conference on Data Mining, 2003.
58. J. Wang, and J. Han, BIDE: Efficient mining of frequent closed sequences, In Data Engineering Proceedings, pp. 79-90, IEEE, 2004.
59. A. Gomariz, M. Campos, R. Marin, and B. Goethals, Clasp: An efficient algorithm for mining frequent closed sequences, In Pacific- Asia Conference on Knowledge Discovery and Data Mining pp. 50-61, Springer Berlin Heidelberg, 2013.
60. P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, Fast vertical mining of sequential patterns using co-occurrence information, In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 40-52, Springer International Publishing, 2014.
61. P. Fournier-Viger, C.W. Wu, and V.S. Tseng, Mining maximal sequential patterns without candidate maintenance, In International Conference on Advanced Data Mining and Applications (pp. 169-180), Springer Berlin Heidelberg, 2013.
62. P. Fournier-Viger, C.W. Wu, A. Gomariz, and V.S. Tseng, VMSP:Efficient vertical

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

mining of maximal sequential patterns, In Canadian Conference on Artificial Intelligence, pp. 83-94, Springer International, 2014.

63. H.T. Lam, F. Mörchen, D. Fradkin, and T. Calders, Mining compressing sequential patterns, Statistical Analysis and Data Mining, Vol.7, pp.34-52, 2014.
64. P. Tzvetkov, X. Yan, and J. Han, TSP: Mining Top-k Closed Sequential Patterns, Knowledge and Information Systems, vol. 7, no. 4, pp. 438-457, 2005.
65. Nguyen, T,T., and Nguyen, P,K., A New Approach for Problem of Sequential Pattern Mining, Lecture Notes in Computer Science, pp.51–60,2012, doi:10.1007/978-3-642-34630-9_6
66. Aggarwal C,C., Bhuiyan M,A., Hasan M,A., Frequent Pattern Mining algorithms: a survey, Springer, pp 19–64,2014.
67. Agrawal R, Srikant R, Fast algorithms for mining association rules, In proceedings of the 20th international conference on very large data bases, Santiago,1994,pp.1-38.
68. Zhi-Chao Li, Pi-Lian He, & Ming Lei, A high efficient AprioriTid algorithm for mining association rule, 2005 International Conference on Machine Learning and Cybernetics, doi:10.1109/icmlc.2005.1527239
69. Arwa,A., and Ykhlef,M., Hybrid Approach for Improving Efficiency of Apriori Algorithm on Frequent Itemset, International Journal of Computer Science and Network Security, VOL.18, 2018.
70. Xueyan Lin, MR-Apriori: Association Rules algorithm based on MapReduce, Published in IEEE 5th International Conference on Engineering and Service Science, 2014.
71. Nadimi-Shahraki, M.,H., and Mansouri, M., Hp-Apriori: Horizontal parallel-apriori algorithm for frequent itemset mining from big data, 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA),2017,doi:10.1109/icbda.2017.8078825
72. Rak, R., Kurgan, L., and Reformat, M.,A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation, Data & Knowledge Engineering, 2008, pp. 171–197, doi:10.1016/j.datak.2007.05.006
73. Zaki MJ, Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering,Volume: 12 , May/Jun 2000, pp.372–390.
74. Jialiang Yu, Fast and Scalable MapReduce-Based Vertical Mining, Thesis,2018
75. Chee,C,H., and Jaafar,J., Algorithms for frequent itemset mining: a literature review, Artif Intell Rev,2018, https://doi.org/10.1007/s10462-018-9629-z
76. Leung, C. K.S., and Tanbeer, S. K., PUF-Tree: A Compact Tree Structure for Frequent Pattern Mining of Uncertain Data, Lecture Notes in Computer Science, pp. 13–25, 2013, doi:10.1007/978-3-642-37453-1_2
77. Liwen Yue, Review of Algorithm for Mining Frequent Patterns from Uncertain Data, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

78. Xia,D., Lu,X., Li,H., Wang,W., Li,Y., and Zhang,Z., A MapReduce-Based Parallel Frequent Pattern Growth Algorithm for Spatiotemporal Association Analysis of Mobile Trajectory Big Data, Volume 2018, Article ID 2818251, 16 pages https://doi.org/10.1155/2018/2818251
79. Zhou,Z,H. ,Li,H., and Yang,Q., Advances in Knowledge Discover and Data Mining, 11th pacific Asia conference china,May-2007
80. Hariri, R. H., Fredericks, E. M., & Bowers, K. M., Uncertainty in big data analytics: survey, opportunities, and challenges, Journal of Big Data, Springer,2019 ,pp.1-16, doi: 10.1186/s40537-019-0206-3
81. Wang Xizhao, Huang JZ, Wang X, Huang JZ. Editorial: uncertainty in learning from big data. Fuzzy Sets Syst. 2015, pp.1–4.
82. Özkan I, Türkşen IB. Uncertainty and fuzzy decisions. In: Chaos theory in politics. Dordrecht: Springer; p. 17–27. 2014
83. Lesne A. Shannon entropy: a rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. Math Struct Comput Sci. 2014;24(3).
84. Vajapeyam S. Understanding Shannon's entropy metric for information. 2014. arXiv preprint arXiv:1405.2061.
85. Shannon CE. A mathematical theory of communication. Bell Syst Tech J. 1948;27(3):379–423.
86. Rissino S, Lambert-Torres G. Rough set theory—fundamental concepts, principals, data extraction, and applications. In: Data mining and knowledge discovery in real life applications. New York: InTech; 2009.
87. Tavana M, Liu W, Elmore P, Petry FE, Bourgeois BS. A practical taxonomy of methods and literature for managing uncertain spatial data in geographic information systems. Measurement. 2016;81:123–62
88. Chau, M., Cheng, R., and Kao, B., "Uncertain Data Mining: A New Research Direction," in Proceedings of the Workshop on the Sciences of the Artificial, Hualien, Taiwan, December 7-8, 2005.
89. Salahdine F, Kaabouch N, El Ghazi H. Techniques for dealing with uncertainty in cognitive radio networks. In: 2017 IEEE 7th annual computing and communication workshop and conference (CCWC). Piscataway: IEEE. p. 1–6. 2017.
90. Pawlak Z. Rough sets. Int J Comput Inform Sci. 1982;11(5):341–56.
91. Düntsch I, Gediga G. Rough set dependency analysis in evaluation studies: an application in the study of repeated heart attacks. Inf Res Rep. 1995; pp.25–30.
92. Djenouri, Y., Belhadi, A., Lin, J.C.W., Cano, A., 2019. Adapted K-nearest neighbors for detecting anomalies on spatio-temporal traffic flow. IEEE Access 7, 10015–10027.
93. Espejo, P.G., Ventura, S., Herrera, F., 2009. "A survey on the application of genetic programming to classification." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)40(2),121–144.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

94. Cano, A., & Ventura, S. (2014, July). "GPU-parallel subtree interpreter for genetic programming" In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (pp. 887-894). ACM.
95. Li, Q., Salman, R., Test, E., Strack, R., Kecman, V., 2013. Parallel multitask cross validation for support vector machine using GPU. J. Parallel Distrib. Comput. 73 (3), 293–302.
96. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Stoica, I., 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing 2-Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association.
97. El-Hajj M, Zaiane OR (2003) COFI-Tree mining—a new approach to pattern growth with reduced candidacy generation. workshop on frequent itemset mining implementations (FIMI'03) in conjunction with IEEE-ICDM, Melbourne.
98. Song M, Rajasekaran S (2006) A transaction mapping algorithm for frequent itemsets mining. IEEE Trans Knowl Data Eng 18(4):472–481
99. Baralis E, Cerquitelli T, Chiusano S, Grand A (2013) P-Mine: parallel itemset mining on large datasets. In: Paper presented at the 2013 IEEE 29th international conference on data engineering workshops (ICDEW), Brisbane.
100. Pyun G, Yun U, Ryu KH (2014) Efficient Frequent Pattern Mining based on linear prefix tree. Knowl-Based Syst 55:125–139
101. Hoseini MS, Shahraki MN, Neysiani BS (2015) A new algorithm for mining frequent patterns in CanTree. In: Paper presented at the international conference on knowledge-based engineering and innovation, Tehran.
102. Feddaoui I, Felhi F, Akaichi J (2016) EXTRACT: new extraction algorithm of association rules from frequent itemsets. In: Paper presented at the 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), San Francisco.
103. Titarenko.S.S., Titarenko, V,N., Aivaliotis, G., and Palczewskia,J., Fast implementation of pattern mining algorithms with time stamp uncertainties and temporal constraints, Springer Open 2019,pp:1-34.
104. Xie, M., & Tan, L. (2019). An efficient Algorithm for Frequent Pattern Mining over Uncertain Data Stream. 2019 12th International Symposium on Computational Intelligence and Design (ISCID). doi:10.1109/iscid.2019.00026
105. C.M.Liua., K.T.Liao., Efficiently Predicting Frequent Patterns over Uncertain Data Streams, 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN)-2019,pp:15-22.
106. B.K.Tripathy, S.K. Sahu, K.K. Barik and M.V.N. Prasad: Study on Remote Sensing Data Classification Using Statistical Techniques, Universe of Emerging Technologies and Science, Volume II, Issue V – May 2015, pp.1-6.
107. RiaElin Thomas, SharmilaBanu K and B.K. Tripathy: Image Anonymization Using Clustering with Pixelization, Proceedings of the ISCSC 2018.

ISSN: 0011-9342 | Year 2021
Issue: 7 | Pages: 3983-4013

108. B.K.Tripathy, M.K.Sishodia and S.Jain: Societal Networks: The networks of dynamics of interpersonal associations, In: Social Networks- Mining, Visualisation and Security, Intelligent Systems Library, 65, Springer, (2014), DOI: 10.1007/978-3-319-05164-2_5©Springer International Publishing Switzerland, pp.101-127
109. Sujata Dash, B. K. Tripathy and Ata Ur Rahman:Modeling, Analysis and Application of Nature-Inspired Metaheuristic Algorithms, IGI Publications, (2016).