A PARALLELIZED INCREMENTAL FREQUENT PATTERN MINING ALGORITHM FOR UNCERTAIN DATA

Submitted by: Thejaswini D M
171IT243

Motivation

The rapid increase in the amount of data in this big data era has lead to proposal of many approaches to solve the incremental problem of certain data, but these approaches did not address uncertain data

Existing Algorithms

Apriori Algorithm

Fp-Growth Algorithm

UF-Growth Algorithm

Cuf-Growth Algorithm

IFPM-BS Algorithm

RUFP Algorithm

Problem statement

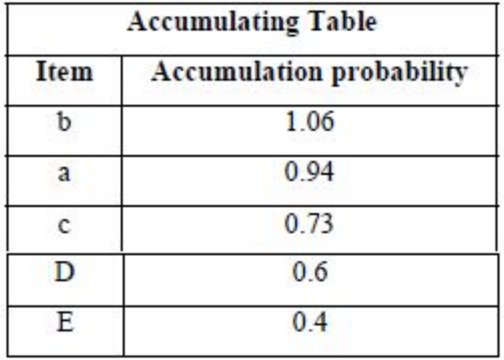
There is a need for a method to retain the tree structure in uncertain data, to maintain the flexibility of the original data, and should not require to rebuild the tree again.

Method description

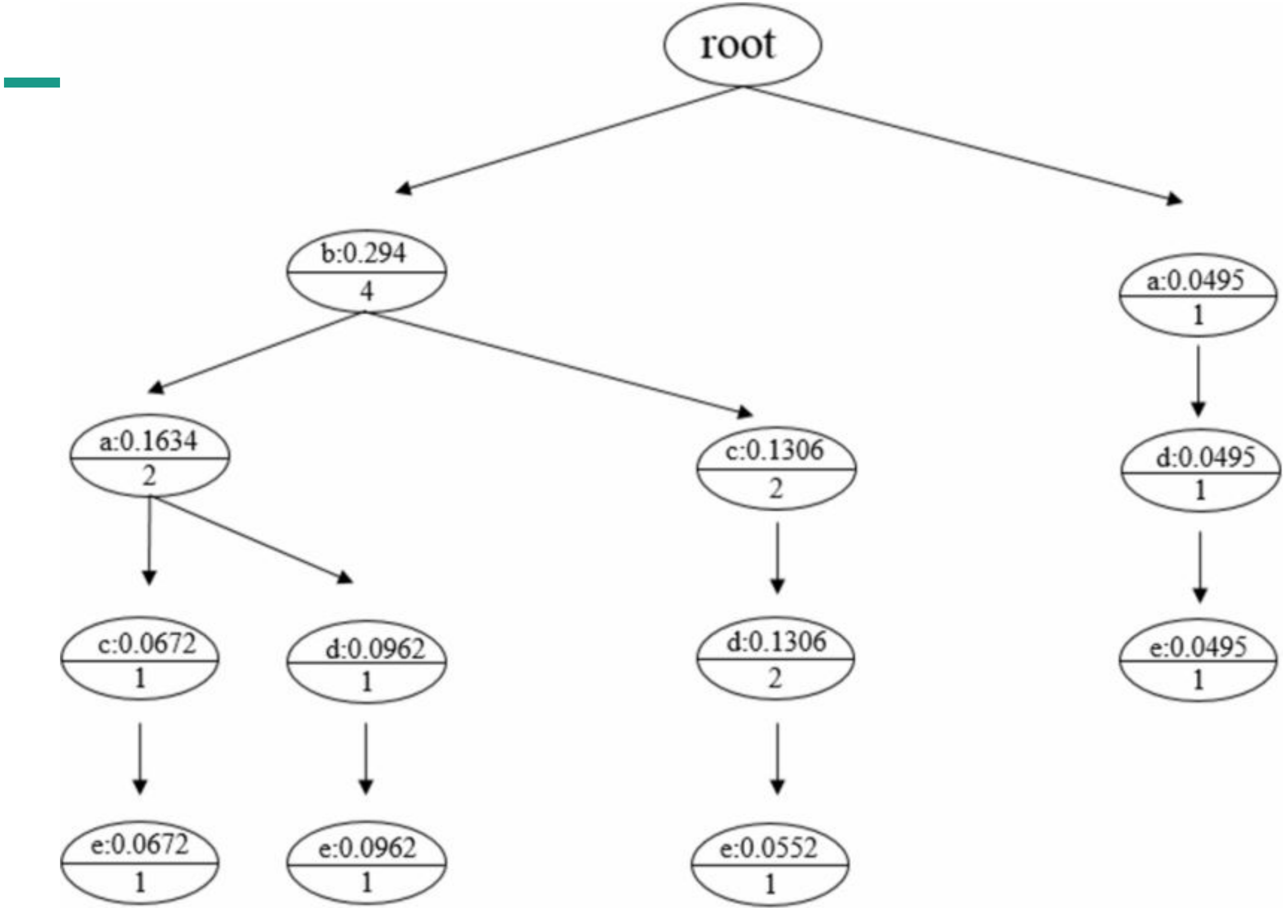
1. Constructing the original tree
2. Updating the accumulation table and tree based on the new transactions added
3. Adding the new transactions to the original tree and obtaining the frequent patterns

1. Constructing the original tree

1. Constructing accumulation table
2. Sorting the items according to their probabilities in transactions record
3. Sorting the transactions
4. Constructing the tree
5. Identifying the frequent patterns

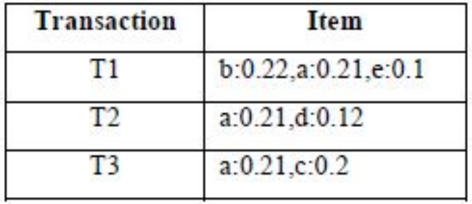


Accumulating Table	
Item	**Accumulation probability**
b	1.06
a	0.94
c	0.73
D	0.6
E	0.4

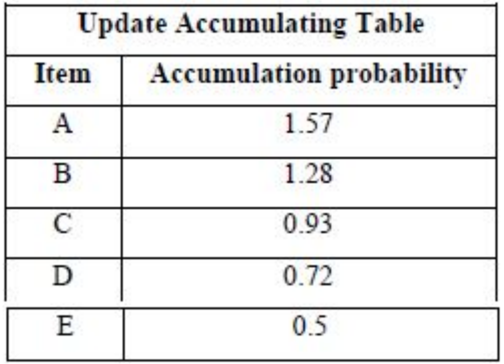

root
b:0.294
4
a:0.0495
1
a:0.1634
2
c:0.1306
2
d:0.0495
1
c:0.0672
1
d:0.0962
1
d:0.1306
2
e:0.0495
1
e:0.0672
1
e:0.0962
1
e:0.0552
1

Updating the accumulation table and tree based on the new transactions added

1. Loading the new transactions

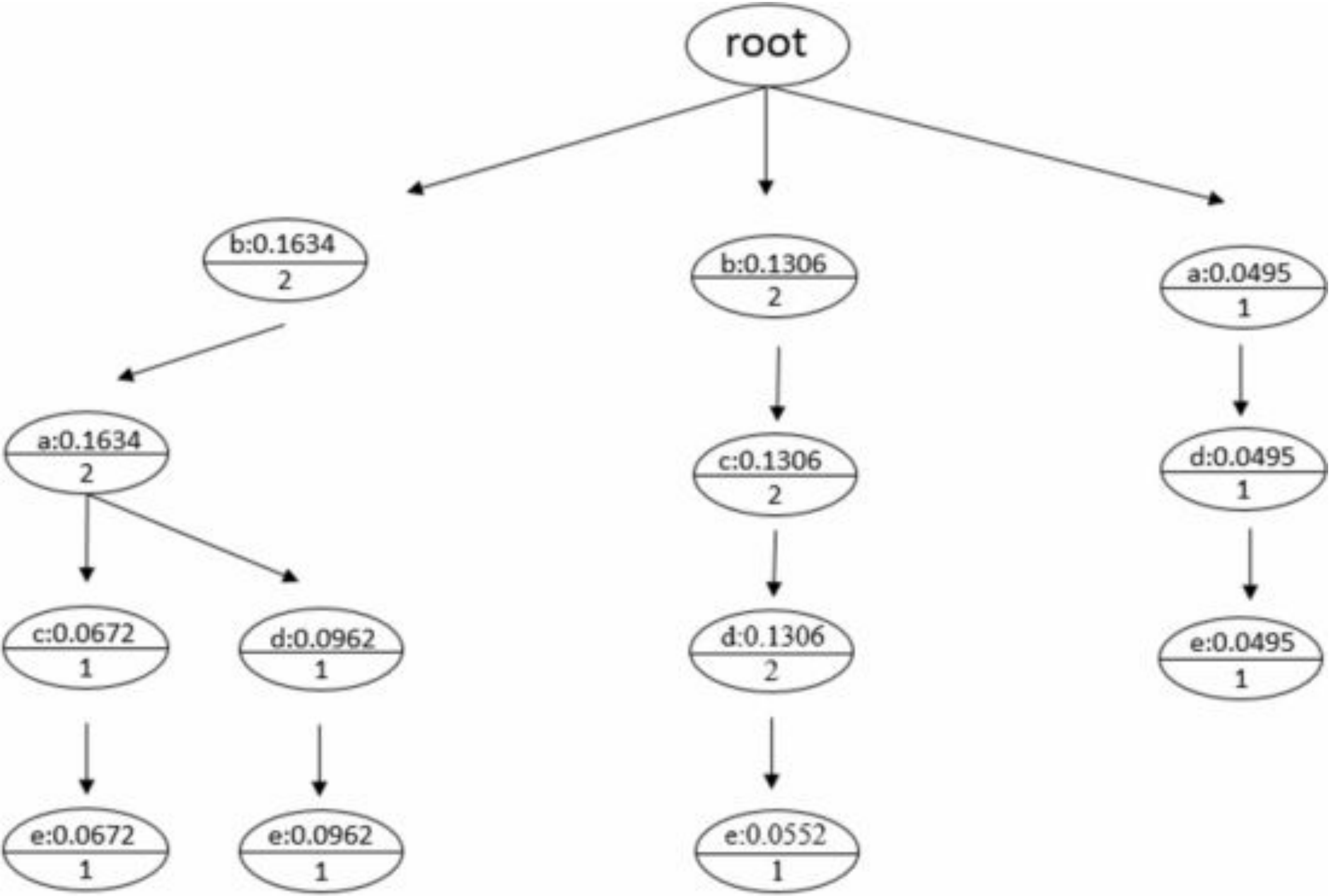


Transaction	Item
T1	b:0.22,a:0.21,e:0.1
T2	a:0.21,d:0.12
T3	a:0.21,c:0.2

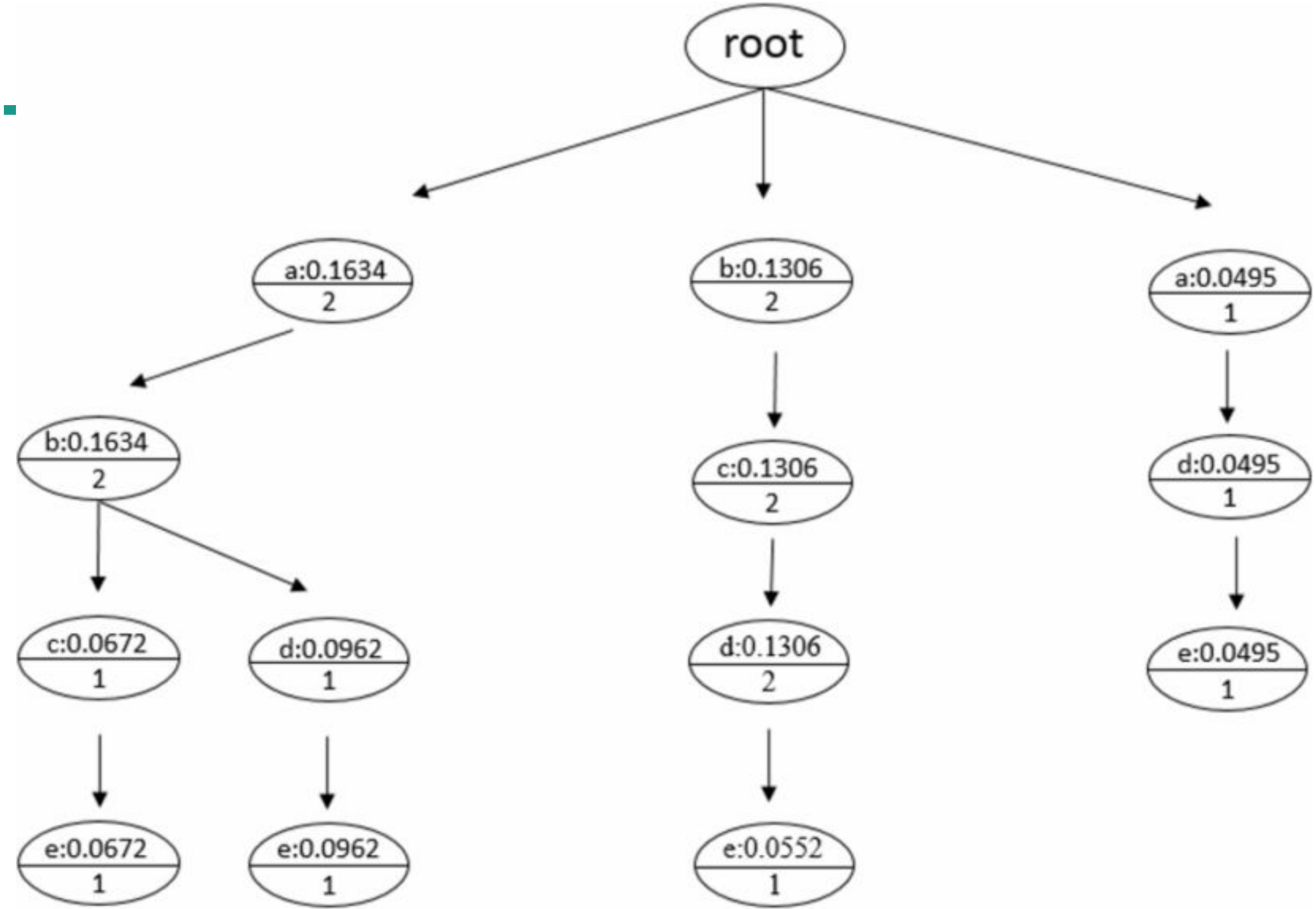


Update Accumulating Table	
Item	Accumulation probability
A	1.57
B	1.28
C	0.93
D	0.72
E	0.5

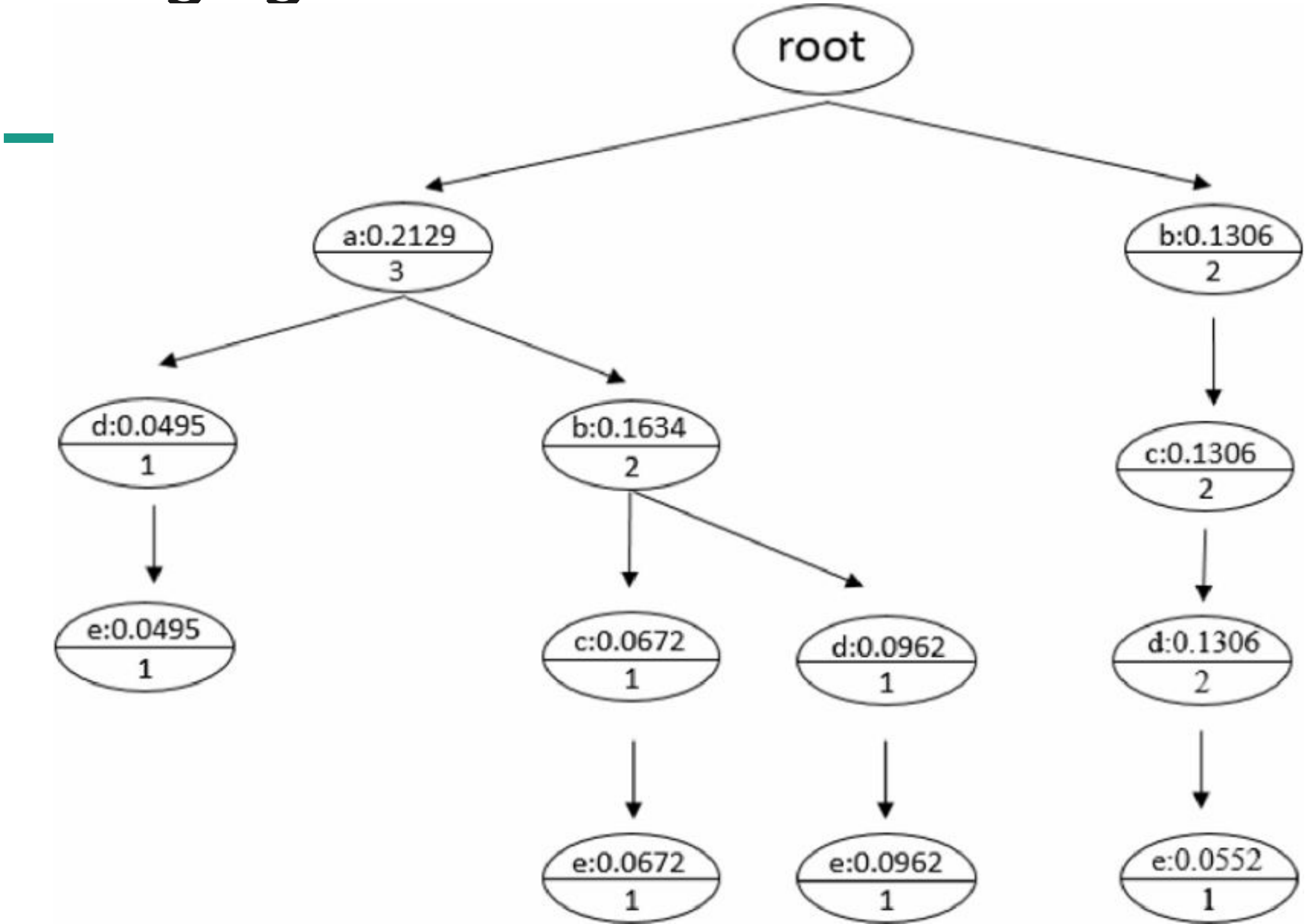
Splitting


root
b:0.1634
2
a:0.1634
2
c:0.0672
1
e:0.0672
1
d:0.0962
1
e:0.0962
1
b:0.1306
2
c:0.1306
2
d:0.1306
2
e:0.0552
1
a:0.0495
1
d:0.0495
1
e:0.0495
1

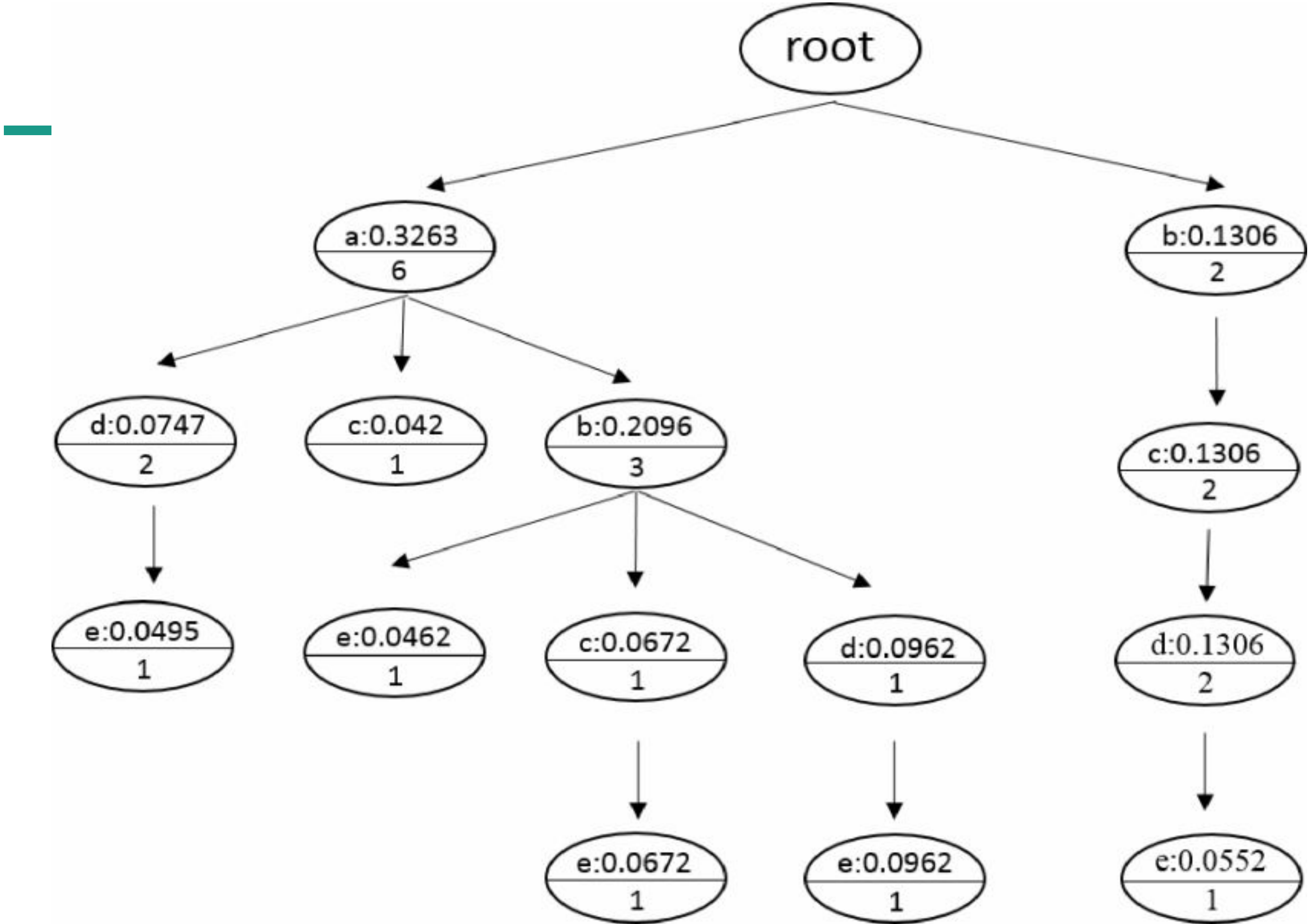
Adjusting


root
a:0.1634
2
b:0.1306
2
a:0.0495
1
b:0.1634
2
c:0.1306
2
d:0.0495
1
c:0.0672
1
d:0.0962
1
d:0.1306
2
e:0.0495
1
e:0.0672
1
e:0.0962
1
e:0.0552
1

Merging


root
a:0.2129
3
b:0.1306
2
d:0.0495
1
b:0.1634
2
c:0.1306
2
e:0.0495
1
c:0.0672
1
d:0.0962
1
d:0.1306
2
e:0.0672
1
e:0.0962
1
e:0.0552
1

Tree Updating



Parallelization of one module

Very first module which is independent of previous outputs can be parallelized.

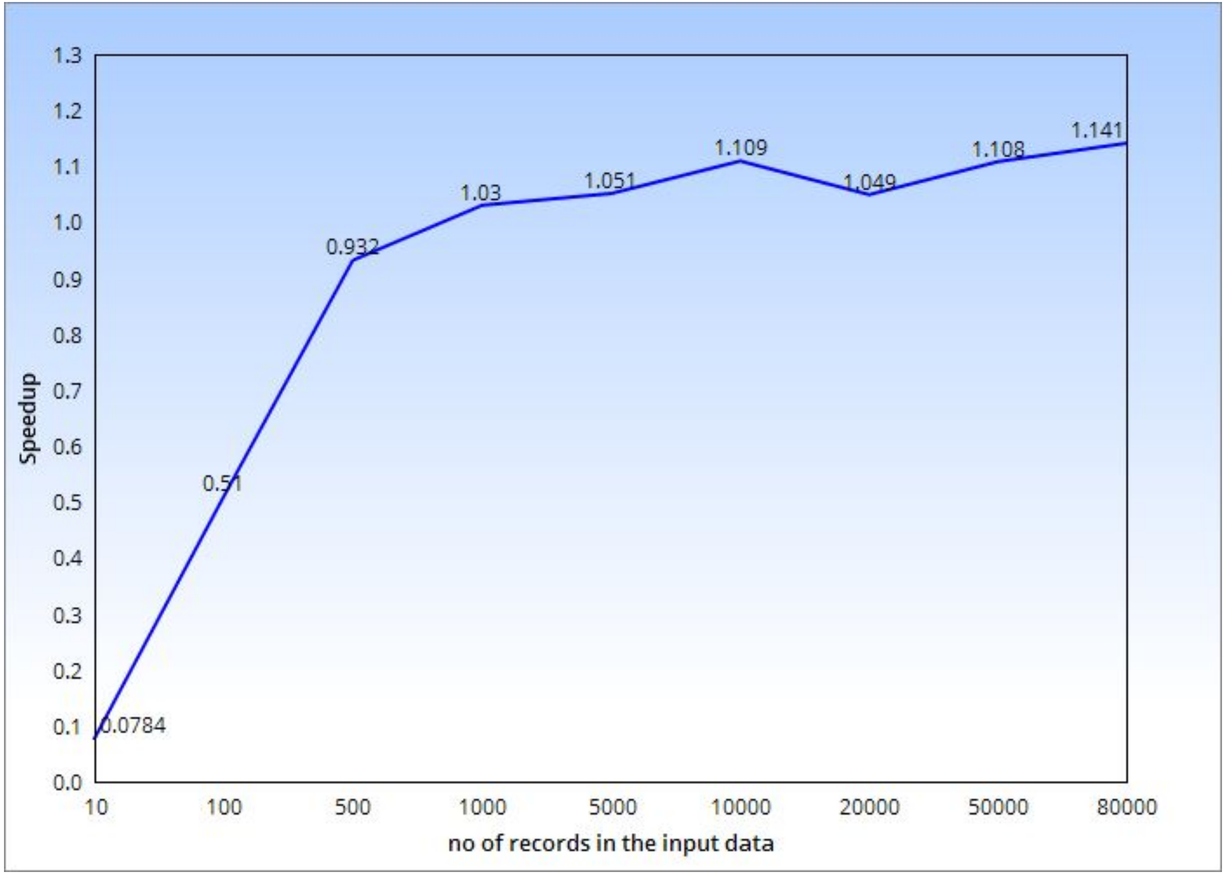
Work Done

Random dataset generation is also completed

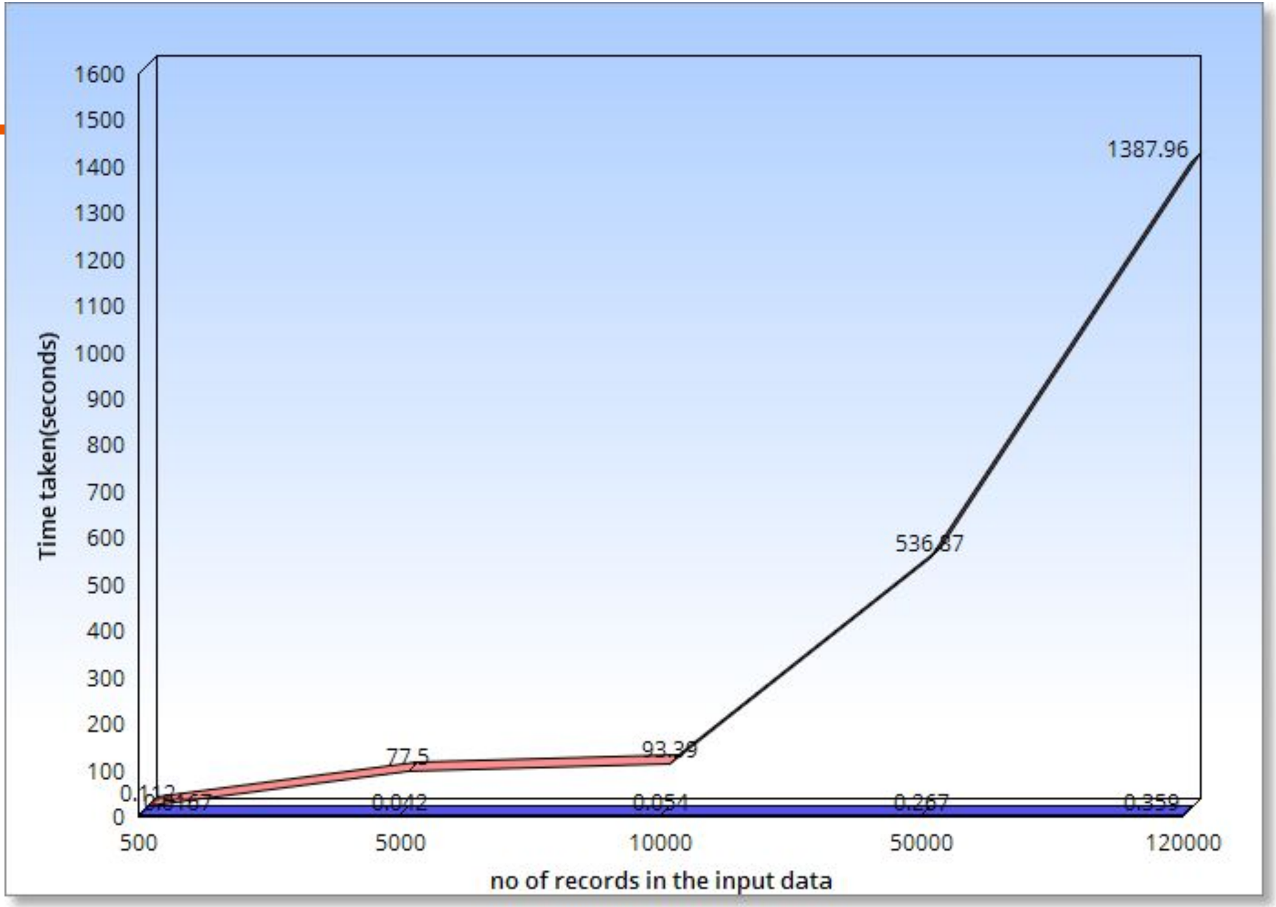
- Creation of Initial trie and displaying initial frequent itemsets.
- Splitting, adjusting, merging steps
- Tree updation
- Parallelization of the frequency counting step

Results

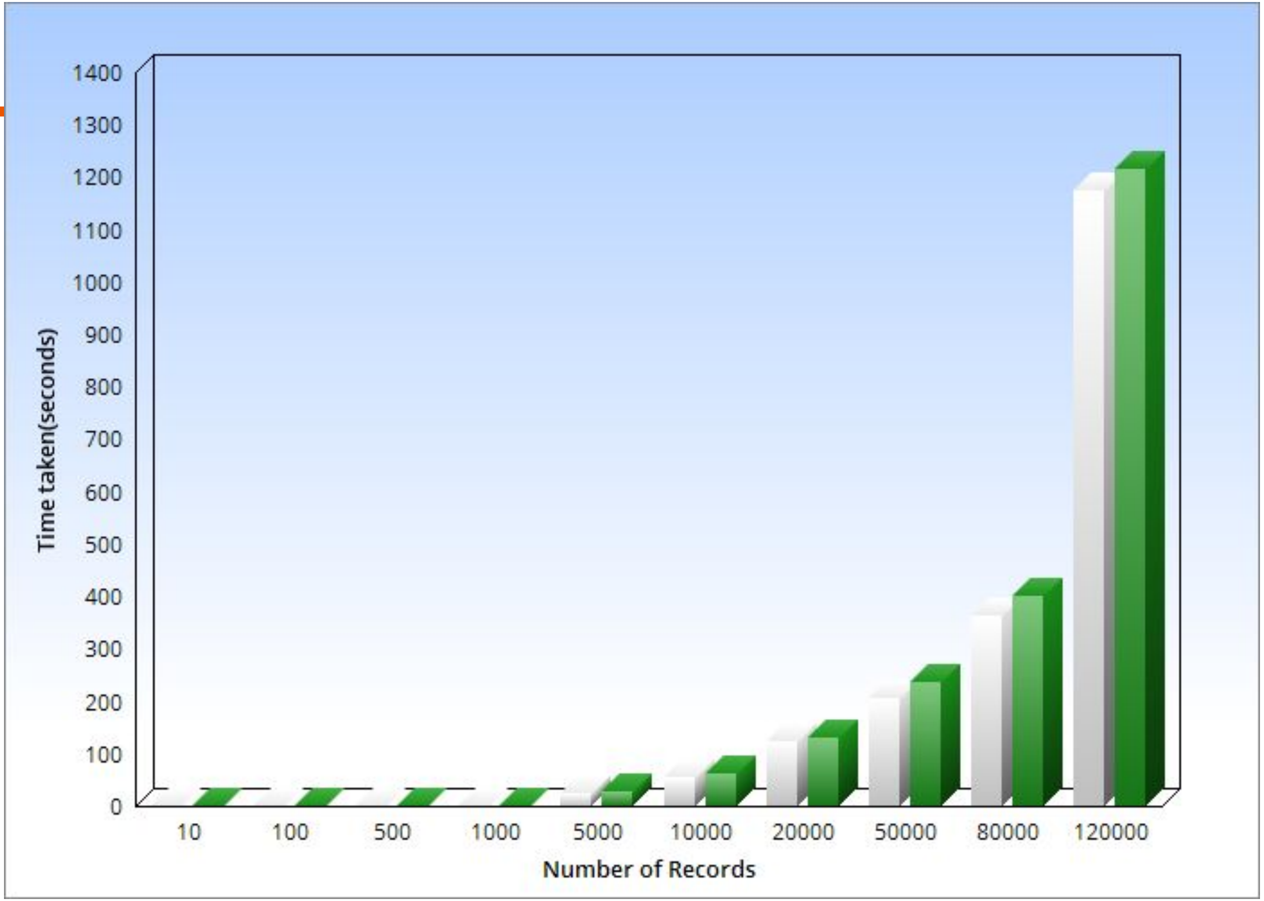
- The Speedup obtained :



Parallelized Module time taken analysis



Overall Time taken analysis



Thank You