JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(1), MARCH, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

Itemset Fusion: A New Technique for Itemsets Generation

[1]Matthew Tunde Ogedengbe, [2]Sahalu B. Junaidu [3]Donfack Kana
[1]*Department of Computer Science,*
University of Agriculture, Makurdi, Nigeria
[2]*Department of Computer Science,*
Ahmadu Bello University, Zaria, Nigeria

ABSTRACT
Association rule mining derives insight from transactional data but suffers from frequent database scans and explosive generation of candidate sets in a large database. This study presents a new items merging technique called, Itemset Fusion Technique (IFT) to generate unique candidate k-itemsets from fused k-items in transaction dataset. The proposed IFT focused on minimizing database scans and generating optimal number of association rules by creating fused k-itemsets locally within each transaction. The fused k-itemsets created are viable candidate itemsets that exist in the database with non-zero support. The performance of the IFT was compared with the classic Apriori in term of number of database scans and number of association rules generated. Experiments was conducted on two real world datasets show IFT scans the database only once and minimizes association rules generated by over 90% against the standard Apriori algorithm. Thus, the proposed IFT accelerates performance and improves mining productivity significantly.

ARTICLE INFO
Article History
Received: November, 2023
Received in revised form: January, 2024
Accepted: January, 2024
Published online: March, 2024

KEYWORDS
Fusion; Itemset; Candidate-set; Support; Confidence

INTRODUCTION

Discovering relationships among data in large databases (Agrawal and Srikant, 1994; Agrawal *et al*., 1993), is important because organizations derive business value from the associations and the connections that exist among these data values. The most common and effective data mining tool used for discovering associations among transaction data is the Apriori algorithm (Teymoornejad, 2018). Apriori Algorithm (Ap) is an unsupervised mining tool which was first proposed for market basket analysis in 1993 by Agrawal (Agrawal *et al*., 1993). It is used to observe customer behavior (purchasing patterns) in retail stores. However, in association rule mining, Ap has currently gained much grounds and has been adopted in various areas of research such as image processing, customer relationship management, aviation mining (Singh and Maiti, 2020), information visualization, air pollution mining, data mining in education and text mining (Abdullah, 2016). Apriori algorithm is simply used to extract meaningful information from a data warehouse or a database. The activities of Apriori are mainly executed on formation of frequent items or itemsets and generation of candidate sets in order to generate a set of required association rules with the help of minimum support and confidence measures (Chee *et al.,* 2019 Riszky and Sadikin, 2019). The values of minimum support and confidence affect association rule generation (Patill *et al*., 2016). The set of rules generated are strictly based on the compliance on the value of minimum support and that of confidence, the rules that meet the minimum value of these measures are considered as set of strong or interesting rules (Ginting *et al*., 2018). Typically, the process of discovering associations, relationships, or correlation between a set of items in a transaction, relational database or data warehouse is called association rule mining (Vivekananda and Gunasekaran, 2023)

Frequent itemsets mining is a key step for association rule learning. The standard Apriori algorithm (Ap) identifies item combinations that satisfy a support threshold by iteratively

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

generating candidate k-itemsets, pruning infrequent ones, and repeating the process. However, a major limitation is that Apriori ends up generating several itemset candidates that do not exist in the actual database. This leads to wasted computations and unnecessary scans to count non-existent candidates during rule learning (Borgelt, 2012; Syahrir and Anggrawan 2023).

The extensive candidate generation (2^n-n-1, where n is number of items) process also leads to combinatorial explosion as the dataset grows larger. For example, for a transaction database with 100 items, Apriori would need to evaluate and test a massive number of over 10^30 candidate itemsets, even though only a tiny fraction has any actual support in the data (Wu *et al.*, 2023; Rana and Mondal, 2021; Khalid *et al.*, 2013). This inefficiency poses significant scalability and performance issues for real-world high volume transaction data.

Therefore, faster techniques that can selectively generate only verified itemsets that appear within the given transactions can offer enhanced efficiency. In this paper, we introduce a "Item Fusion Technique" which restructures the database transactions by fusing items into useful and existence itemsets before mining. By transforming the input, the itemsets will be more rule-oriented and will inherently avoid wasted computations on artificial non-existent combinations during frequent patterns discovery.

RELATED WORKS

Itemset mining has been an important area of research in data mining and knowledge discovery. Frequent itemset mining, first introduced in the context of basket analysis by Agrawal (Agrawal and Srikant, 1994; Agrawal *et al.*, 1993), involves identifying sets of items that frequently occur together in transactional data. Over the past few decades, numerous algorithms have been developed for efficiently mining frequent itemsets, including Apriori, FP-Growth, and Eclat (Zaki *et al.*, 1997). The Apriori algorithm pioneered frequent itemset mining through an iterative, bottom-up candidate generation framework. Apriori suffers from significant computational inefficiencies due to its breadth-first traversal of the itemset lattice and exhaustive enumeration of candidate itemsets. The vast majority of itemset mining techniques like Apriori [1] rely on lexicographic generation of candidate itemsets, which can lead to a combinatorial explosion of candidates that do not actually exist in the dataset.

Many optimizations of Apriori's core candidate generation process have consequently been proposed over the past 25 years. For example, mining frequent itemsets without candidate generation (Leung, 2008) used a lattice traversal method that only created length (k+1) itemsets from length k itemsets if they appeared as sub-patterns. The AprioriHybrid algorithm in (Balasubramanian *et al.*, 2010) introduces a depth-first traversal to decompose the mining task into smaller subtasks. The Fast Apriori algorithm (Borgelt, 2012) avoids unnecessary re-scans of itemsets by directly determining local support counts. DP-Apriori (Cheng *et al.*, 2015) reduces candidate sizes with a dynamic programming-based pruning technique. Adaptive Apriori technique (Patill *et al.*, 2016) was implemented to reduce database scans and itemset generation time. It dynamically prunes transactions to shrink databases for each itemset level separately - retaining only necessary rows per round. Experiments demonstrated that database size reductions at each iteration, culminating in lower overall runtimes versus standard Apriori's repeated full database passes.

Other work such as FP-Tree growth focused on improving the structure of the itemset lattice used to drive generation. For example, an adaptive FP-Tree growth (Han and Nv, 2017) condensed the dataset into a compressed prefix tree to avoid repeated database scans. Several recent works have continued exploring refinements to the core Apriori itemset generation process.

An improved Apriori algorithm, called Intelligent Apriori (IAP) was proposed in (Karimtabar and Fard, 2020), to enhance data mining efficiency. The method introduced an intelligent approach, demonstrating that creating itemsets with multiple items reduces the number of steps. Additionally, storing transaction numbers

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

for each itemset is shown to significantly reduce search time. Comparative analysis with the existing Apriori algorithm revealed a 46% reduction in runtime for obtaining frequent itemsets using the proposed IAP algorithm, showcasing its improved performance in generating frequent items from datasets.

A Spark-based Apriori with Reduced Shuffle Overhead (SARSO) was developed in (Raj *et al*., 2021) to mine frequent itemsets at scale. SARSO utilizes Spark's distributed, in-memory processing to enable fast iterative scanning but further restricts shuffle overhead between iterations for efficiency gains. Experiments on benchmark datasets demonstrated that SARSO outperformed existing Spark-Apriori approaches in running time and scalability - showing continued performance advances by creatively optimizing enduring algorithm foundations.

A data mining method, investigated consumer buying patterns in retail supermarkets in (Rana and Mondal, 2021). It analyzed customer baskets to reveal product relationships, aiding retailers in strategic planning and layout design. The focus was on frequent itemsets mining as the initial step, followed by association rules mining to uncover customer preferences. The proposed methodology addressed challenges in traditional association rules mining by introducing a multilevel approach, identifying seasonal patterns and associations, enhancing the decision-making process for season-based merchandising. The study, based on a dataset from a large supermarket in Bangladesh, discovered 442 seasonal patterns and 1032 seasonal association rules, complementing traditional rules with improved insights.

Support calculation was optimized in (Hodijah *et al*., 2022) using multi-threaded and trie data structures for market basket analysis. The support calculation, crucial for association rule identification, is identified as a bottleneck process causing delays. The study explored five multi-threaded models based on Flynn's taxonomy to reduce processing time, highlighting that the double single process, multiple data (SPMD) variant outperforms others, achieving almost three times faster performance in handling 5-itemsets and 10-itemsets. The research not only considered processing time but also evaluated the time differences among multi-threaded models as the number of item variants increased.

An optimized FP-Growth approach (Shoawkat *et al*., 2022) condenses the dataset into linked lists for more efficient candidate pruning. A Matrix Based Apriori algorithm (MB_Apriori) was proposed in (Vivekanandan and Gunasekaran, 2023) to efficiently mine frequent itemsets by converting the transaction database into a boolean matrix representation. Five pruning strategies were introduced to reduce the candidate itemset search space. Experiments on multiple datasets demonstrated that MB_Apriori decreases frequent itemset generation time by 71-86% over traditional Apriori. When benchmarked against other optimized approaches, MB_Apriori still indicates a 20% improved performance - validating continued research to address enduring inefficiencies.

Apriori's time performance was improved in (Wu *et al*., 2023) via database scan reductions, including maintaining Map tables for support counting and splitting transactions into disjoint partitions for localized mining. Experiments demonstrated that the optimized algorithm decreased database scans and outperformed Apriori variants by 94% in frequent itemset and rule filtering runtimes and it enhanced result accuracy from exam data mining. Despite advances, excessive scanning persists as a core inefficiency, spurring continued innovation in data mining.

An improved Apriori algorithm based on the influence weight is proposed in (Niu *et al.,* 2023). Firstly, according to the impact of items on affairs and the relationship between items, a weight distribution method based on item influence is used to weigh items, so as to extract more hidden and valuable associated information. Secondly, to reduce the scanning times of the database and the overhead of I/O ports, the weight vector and compression matrix were introduced to represent the transaction database. Finally, it was compared with the classical Apriori algorithm and the Apriori algorithm based on compressed matrix. Experimental results showed

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(1), MARCH, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

that the improved Apriori algorithm had higher efficiency, and could mine out frequent patterns with high influence.

Apriori optimizations were developed in (Li *et al*., 2023) using matrix representations and association rule mining to improve candidate generation, demonstrating superior time and spatial complexity over classical Apriori. Applied to equipment reliability data, the enhanced algorithm provided an effective approach for quality degradation analysis and remaining useful life prediction. Though limitations exist in data type flexibility, optimized mining of rich equipment data expands applications while enduring Apriori foundations. The optimized algorithm applied a more selective, targeted candidate generation strategy by introducing an upfront screening process to filter the connections between frequent itemsets. This avoids wasted downstream efforts on useless candidates. The emphasis was on smarter generation rather than a matrix-centric technique.

An Apriori optimization using four-way database partitioning and vertical TID-list transformation was presented in (Syahrir and Anggrawan, 2023) to accelerate itemset counts for faster frequent pattern mining. Restricting antecedent/consequent cardinality further focuses useful rules from the streamlined generation. Though partitioning complexity persists, targeted database adaptations continue advancing decades-old quests for efficiency. While incremental performance gains on the standard Apriori paradigm have been achieved, arguably the core downfall of excessive, redundant candidate generation remains largely unsolved. The proposed fusion-based approach, discussed in the next section, sidesteps this issue entirely by avoiding generating non-existence candidate sets and multiple database scans. The next section presents the methodology and algorithms for the proposed itemset fusion approach.

METHODOLOGY

This section presents the methods adopted for association rule mining, laying the foundation for the proposed research approach. Firstly, fundamental theories of association rule mining were introduced, elucidating key concepts such as support, confidence, and lift, which underpin the analytical framework. Subsequently, we detail the chosen algorithm, modifications, or adaptations made to accommodate the specific objectives of our study.

Association Rule Mining

Association rule mining is a fundamental data mining method employed to extract significant associations or correlations between itemsets within extensive datasets. For formal representation, let *D* denote a dataset encompassing all elements in the database. Each attribute of a record in *D* is designated as an item, collectively forming an itemset. Transactions, denoted as *T*, represent nonempty records (Liu and Wu, 2018). In a transaction *T*, consider two items, *X* and *Y*, both proper subsets of *T*. If *X* and *Y* are nonempty subsets with an empty intersection, the association rule $X \rightarrow Y$ is established within the item set *T*.

i. **Support:** This is a vital metric used to quantify the frequency of occurrence of a particular item in a dataset. Mathematically, the support of an item X, denoted as $(Support(X))$, is calculated as the ratio of the number of transactions containing X to the total number of transactions in the dataset *D* as shown in Equation 1. It provides insights into the popularity of an item set and is a fundamental measure for identifying significant associations. Higher support values indicate a more prevalent occurrence of the item set, making it a key parameter in determining the relevance and strength of association rules (Wu and Liu, 2019).

$$Sup(X) = \frac{frequency\ (X)}{Total\ \ transactions\ (D)} \qquad (1)$$

ii. **Confidence**: This metric measures the reliability or strength of an association rule. Mathematically, the confidence of a rule $X \rightarrow Y$, denoted as

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

$Confidence\ (X \rightarrow Y)$, is calculated as the ratio of the support of the item set $(X \cup Y)$ to the support of X as depicted in Equation 2.

$$Confidence(X \rightarrow Y) = \frac{Sup(X \cup Y)}{Sup(X)} \quad (2)$$

This formula quantifies the **likelihood** that if *X* occurs, *Y* will also occur in the same transaction. Confidence values range from 0 to 1, where a higher confidence indicates a stronger association between the item sets *X* and *Y*.

iii. **Lift**: This is another metric that assesses the strength and significance of an association rule $X \rightarrow Y$ beyond what would be expected by chance. Mathematically, Equation 3 defines lift of a rule as the ratio of the confidence of the rule to the support of *Y*:

$$Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{Sup(Y)} \quad (3)$$

A lift value greater than 1 suggests that the occurrence of *X* has a positive impact on the occurrence of *Y*, indicating a meaningful association. Conversely, a lift less than 1 implies a weaker or potentially negative association. Lift provides valuable insights into the practical significance of association rules beyond individual support and confidence measures. However, association rule mining involves two basic steps; generate frequent itemset from candidate set whose support value is at least a minimum support threshold and generate strong association rules from the frequent itemset based on minimum confidence threshold value (Liu and Wu, 2018; Wu and Liu, 2019).

Apriori Algorithm

The Apriori algorithm is a fundamental association rule mining technique widely employed to discover relationships and associations among items in large datasets. It operates on the principle of generating frequent itemsets by iteratively identifying and pruning infrequent itemsets. The key concept is the *Apriori property*, stating that if an itemset is frequent, then all of its subsets must also be frequent. The algorithm consists of two main steps:

i. **Joining phase:** this phase combines pairs of frequent k-itemsets to form larger (k+1)-itemsets called candidate sets of size (k+1).
ii. **Prune**: Eliminates itemsets that do not meet the minimum support threshold. Those itemsets that meet the minimum support threshold are termed frequent itemset.

Through repeated iterations, Apriori progressively builds larger itemsets until no further expansion is feasible. It is efficient for handling sparse datasets and plays a crucial role in tasks like market basket analysis, where discovering associations between items is essential for understanding consumer behavior. Despite its effectiveness, Apriori's performance can be impacted by the curse of dimensionality in large datasets, multiple database scans, expensive candidate generation, lack of scalability, and assumptions about main memory feasibility impose practical challenges on the Apriori algorithm (Liu and Wu, 2018; Wu and Liu, 2019; Rana and Mondal, 2021).

Proposed Itemset Fusion Technique (IFT)

The term *fusion* means the combining or merging of items, and *itemset* refers to a set of items occurring together in a transaction. Therefore, *itemset fusion* refers to merging individual items into a combined itemset within each existing transaction record in a dataset. The proposed technique creates a new itemset by combining items that appear together horizontally within the same transaction.

Definition 1: Let $T = \{t_1, t_2, \dots, t_m\}$ be the set of all transactions in a dataset. Let I be the set of all unique items across all transactions in dataset, such that $I = \{i_1, i_2, \dots, i_n\}$, then a fusion set t_i^f is defined as the set of all unique pairwise unions of the items in transaction t_i as shown in Equation 4 as;

$$t_i^f = \{x \cup y | x, y \in t_i \ and \ x \neq y\}, \quad (4)$$

Where:

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(1), MARCH, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

- t_i^f is the fusion set or fused transaction t_i.
- *x, y* denotes unique *items* in transaction t_i
- $t_i \subseteq I$ each transaction t_i is a set of some items which are subsets to overall items within *I*

Consider for example, two transactions t_1 and t_2 are fused as follows;

$t_1 = \{A, B, C, D\}$

$t_1^f = \{AB, AC, AD, BC, BD, CD\}$

$t_2 = \{B, C, D\}$

$t_2^f = \{BC, BD, CD\}$

The above example illustrates how all possible 2-itemsets were generated from fusion set by combing distinct pairs of items from the given transactions and taking their union.

The itemset fusion technique begins by taking k-item to generate itemset of size (*k+1)* to generate set of all possible 2-itemset in the first iteration.

In the first iteration *k = 1*, this builds all size-2 itemset from *T* as shown in the example above In the second iteration *k = 2*, this generates itemset of size-3 (3-itemset) from fused 2-itemset, and so forth up to the (*n-1*)th iteration generates size-*n* itemset from the previous size-(n-1) itemset. This definition properly generalized the iterative fusion set to allow incrementally creating larger-sized itemset from transaction *T*, rather than just pairs.

The proposed IFT comprises of the following special features;

- Database is scanned just once to generate the fused itemsets at *k = 1*, as 2-itemset using fusion technique. The subsequent itemset of size-3 is generated from the fused itemset of size-2. This method continues until no more items to be fused together.
- It applies the binomial combinational operation $\binom{n}{k}$, to fuse the items column-wise to generate all possible combination of itemsets, where *n* is the number of items in a row (each transaction) and *k* is the size of the items.
- The IFT generates only sets of items that exist in the database unlike the standard Apriori which sometimes generates some non-existing itemsets in the database.

Pseudocode for the proposed itemset fusion technique (IFT)

The proposed IFT comprises four key steps as outlined below.

1. *Iterate over each transaction*: the IFT occurs at *k* = 1, by fusing the frequent 1-itemsets in each transaction to generate the next 2-itemset.
2. *Generate all combinations*: in each transaction, each item[x] is fused to the next item[y] iteratively from line 3 to 8 to generate only set of existing itemset in the database.
3. *K-candidate sets*: the fused itemsets are updated as the new fused transaction which is regarded as k-itemset i.e. fused itemset of size *k* at each iteration as illustrated from line 9 to 11.
4. *Frequent itemset*: At line 12 the k-itemsets were pruned to generate the frequent itemset in line 13 and the fused transaction list represent the transactions in the database and the procedure is repeated until no more frequent itemsets.
5. *Rules generation*: the frequent itemset is used to generate the set of association rules as expected in line 16 to 17.

Algorithm: Pseudocode for the proposed Itemset Fusion Technique (IFT)

Input: transactions, min_supp
Output: rules: Association rules

```
1: for k = 1 to max_len
2:  freq_items = get k-freqitems
3:   for each transaction t in  transactions
4:   fused_itemsets = [ ]
5:   for each itemset s in t:
6:    for each item x in s:
7:     for each item y in s where x < y:
8:     fused = union(s[x],s[y])
9:  add fused to fused_itemsets
10:  update fused_itemsets to fused_transactions
```

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

11: k_candidate_sets = Get itemsets from fused_transactions
12: prune k_candidate_sets based on min_support
13: frequent_items = pruned k_candidate_sets
14: transactions = fused_transactions
15: k++
16: generate rules from frequent_items
17: Return rules

Illustrative example for itemset generation in the standard Apriori algorithm

This section demonstrates how candidate sets are generated in both the standard Apriori and the proposed IFT. Firstly, a given transaction dataset will be used to demonstrate candidate set generation in the standard Apriori algorithm, thereafter, the same dataset will be employed on the proposed IFT to demonstrate the number of candidate sets generated in each technique. Consider a transaction dataset T which consists of six (6) transactions and five (5) items, as shown in Table 1. Assume the minimum support threshold to be 2:

Table 1: Transaction dataset

T_id	Items
T1	A B E
T2	B D
T3	B C
T4	A B D
T5	A B C E
T6	A B C

In the standard Apriori, the Table 2 depicts the list of 1-itemset (C1) sets generated with their corresponding support count (Sc) from the transaction dataset. The transaction dataset is first scanned to determine the C1 candidate sets with their corresponding support. The C1 candidate set whose Sc meets the minimum support threshold is considered as frequent itemset (L1).

Table 2: 1-itemset

S/N.	C3	Sc	L3	Sc
1.	A	4	A	4
2.	B	6	B	6
3.	C	3	C	3
4.	D	2	D	2
5.	E	2	E	2

Table 3 depicts 2-itemset (C2) which was generated based on joining of frequent (L1) itemset to itself. The database is scanned again to determine the support of C2. However, it was observed that Apriori sometimes generates lists of k-itemsets which do not exist in the database. In C2, ten (10) itemsets was generated but only eight (8) actually exist in the database. This is one of the major limitations of Apriori, i.e. frequent database scanning and large number of candidates sets of which some may not exist in the transaction dataset.

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

Table 3: 2-itemset

S/N.	C2	Sc	L2	Sc
1.	AB	4	AB	4
2.	AC	2	AC	2
3.	AD	1	AE	2
4.	AE	2	BC	3
5.	BC	3	BD	2
6.	BD	2	BE	2
7.	BE	2		
8.	CD	0		
9.	CE	1		
10.	DE	0		

Likewise in Table 4, seven (7) itemsets were generated of which only five (5) actually exist in the database. Therefore, in a large database this limitation could incur numerous non-existent itemsets which could increase computational complexity and increases the search space in the dataset. In each candidate generation it was observed that the database must be scanned to determine the occurrence of both existing and non-existing itemsets as the case may be.

Table 4: 3-itemset

S/N.	C3	Sc	L3	Sc
1.	ABC	2	ABC	2
2.	ABD	1	ABE	2
3.	ABE	2		
4.	ACE	1		
5.	BCD	0		
6.	BCE	1		
7.	BDE	0		

Therefore, the proposed IFT resolves these limitations of frequent database scanning and generation of non-existence itemset by scanning the database once and generating only sets of existing itemsets in the database.

Illustrative example for itemset generation in the proposed IFT

In this approach, database is scanned once to determine the support of 1-itemset as that of Apriori but the subsequent k-itemset are generated from the previous (k-1)-itemset. In this approach, the database is scanned once and 1-itemset is generated, the same as in Tab. 2. Thereafter, the dataset in Tab. 1 is updated with only the set of frequent items (L1), but all the items in 1-itemset are frequent therefore there is no pruning of any item in this case. To generate 2-itemset, each item(i) is fused with item(i+1), item(i+2),..., item(n), likewise item(i+1) is fused with item(i+2),...,item(n) this procedure is repeated in each transaction until all the items are fused with their subsequent items then the fusing process forms k-itemset of size k. Table 5 depicts the fused transaction dataset of size k (= 2).

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

Table 5: Fused Transaction dataset

T_id	Items
T1	AB, AE, BE
T2	BD
T3	BC
T4	AB, AD, BD
T5	AB, AC, AE, BC, BE, CE
T6	AB, AC, BC

The support of the fused 2-itemset was determined by counting the number of occurrences of each fused itemset in the fused transaction dataset as shown in Table 6. The frequent fused 2-itemsets (L2) were also determined with respect to the minimum support threshold.

Table 6: Fused 2-itemset

S/N.	C2	Sc	L2	Sc
1.	AB	4	AB	4
2.	AC	2	AC	2
3.	AD	1	AE	2
4.	AE	2	BC	3
5.	BC	3	BD	2
6.	BD	2	BE	2
7.	BE	2		
8.	CE	1		

The IFT approach generates eight (8) itemsets which actually exist in the transaction dataset, compared to the standard Apriori which generates two extra non-existent itemsets (CD and DE). The fused transaction dataset is updated based on the frequent (L2) itemset as shown in Table 7.

Table 7: Updated 2-itemset

T1	AB, AE, BE
T2	BD
T3	BC
T4	AB, BD
T5	AB, AC, AE, BC, BE
T6	AB, AC, BC

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

The 3-itemset is generated by repeating the fusing procedure on the updated 2-itemset as shown in Table 8. It was observed that T2 and T3 returned empty items in fused 3-itemset. This was simply because there was no item to fuse the items of T2 and T3 with.

Table 8. Fused 3-itemset

T_id	Items
T1	ABC
T4	ABD
T5	ABC, ABE, ACE, BCE
T6	ABC

This IFT approach also shrinks the size k-itemset since some fused k-itemset will not be frequent over time. As shown in Table 9, BCD and BDE were never generated because such combinations never exist in the dataset, and five 3-itemsets were generated compared to the standard Apriori which generates seven 3-itemsets. The 3-itemsets were generated from the previous frequent fused 2-itemset without scanning through the main transaction dataset.

Table 9: Fused 3-itemset

S/N.	C3	Sc
1.	ABC	2
2.	ABD	1
3.	ABE	2
4.	ACE	1
5.	BCE	1

COMPARATIVE PERFORMANCE EVALUATION

This section discussed the evaluation of Apriori and the proposed IFT algorithms by comparing their performance with each other in terms of number of database scans, number rules, execution time and the results were represented graphically using two real-world datasets.

Data Collections

In this study, two (2) real-world datasets (student stress factor dataset and psychological capital dataset) were considered for evaluating the performance of the proposed IFT algorithm. The student stress factors dataset was acquired from Kaggle and it consists of 54 instances and 6 stress factors as transaction items. The Psychological Capital Dataset is a psychometric experimental dataset comprising of the psychological capital information of 329 employees with 20 psychological items. In this study, the proposed IFT was written in python programming language and was implemented on google collaboration environment with Python 3 as runtime type and Tensor Processing Unit (TPU) as hardware accelerator.

Experimental Results

The performance of the proposed IFT was evaluated and compared with the standard Apriori in terms of database scans and number of

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

association rules generated. The experiment was performed on both student stress factors and psychological dataset with different minimum support thresholds to prune k-candidate sets and to determine the k-frequent itemsets. Minimum confidence threshold was selected in the range 0.5 to 1.0 to generate and determine the most interesting rules in the required research domains.

Experimental performance evaluation on student stress factor dataset

Table 10 shows the experimental results obtained in term of number of database scans when both Apriori and IFT were experimented on the student stress factor dataset. The result shows that the proposed IFT only scanned the dataset once irrespective of the value of the minimum support threshold, whereas the number of database scans varies with respect to changes in minimum support threshold in the standard Apriori.

Table 10. Comparing the performances on Student stress factor

S/N.	Min_Sup. (%)	No. of Database Scans	
		Apriori	IFT
1.	0.20	5	1
2.	0.30	4	1
3.	0.40	3	1
4.	0.50	3	1
5.	0.70	1	1
6.	0.80	1	1
7.	1.00	-	1

The proposed IFT outperformed Apriori in terms of database scans since IFT does not need to scan through entire large dataset at each iteration, this performance is represented graphically in Figure 1. However, Figure 1 shows that both algorithms have the same performance at minimum support of 0.70. This is because the higher the threshold value, the lower the candidate sets generated and the fewer the database scans. At minimum support threshold 1.00 Apriori could not return any candidate therefore, no database scan.

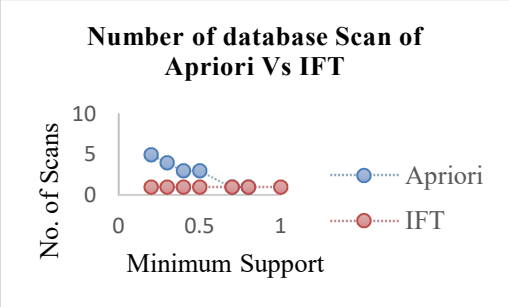

Figure 1. Number of database scans on student stress factors

The strength of association rules is determined by the performance metric know as minimum confidence threshold (Han et al., 2004; Syahrir and Anggrawan, 2023). It shows how

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

interesting the rules are among others. Figure 2 depicts the number of rules generated based on the specified minimum confidence threshold. At 50% threshold the Apriori generated 132 rules while the proposed IFT generates 29 rules at 50% to 70% minimum confidence threshold. Therefore, the proposed IFT outperformed the Apriori in terms of the number of rules generated. As minimum confidence tends to 100%, the number of rules generated decreases.

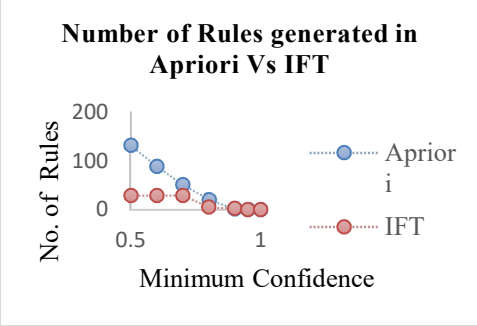

Figure 2. Number of association rules generated on student stress factor dataset

Experimental performance evaluation on psychological capital dataset

In this section, experimental performance evaluation was carried out on psychological capital dataset with both Apriori and IFT. As shown in Figure 3, IFT consistently required 1 scan across all minimum support values tested, from 20% down to 85%, while Apriori required 13 scans at 50% minimum support and 5 scans at 80% minimum support. This demonstrates IFT's efficiency in generating frequent itemsets through its single database scan fusion approach. The major advantage of IFT over Apriori is that the IFT requires only 1 database scan regardless of minimum support, while Apriori needed multiple, costly scans

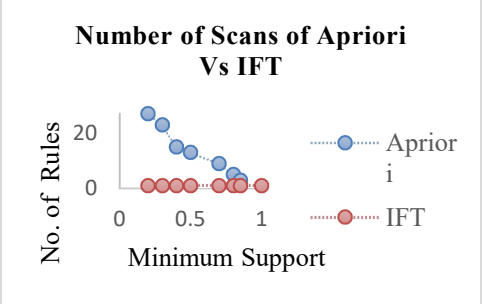

Figure 3. Number of database scans on psychological dataset

IFT generated substantially fewer association rules across all minimum confidence thresholds compared to Apriori as represented in Figure 4. For example, at 50% minimum confidence IFT produced only 119 rules while Apriori generated 1,424 rules - over 10 times more. IFT maintained lower rule counts as confidence increased, dropping to just 1 rule at 100% confidence versus 13 for Apriori. IFT generates far fewer association rules than Apriori at all minimum confidence levels tested, ranging from over 10 times fewer rules at 50% confidence to 13 times fewer rules at 100% confidence. This highlights IFT's advantage in producing compact, focused rule sets versus Apriori's larger, redundant output.

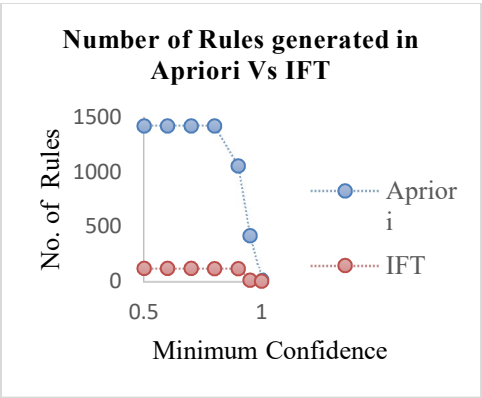

Figure 4. Number of rules generated on psychological dataset

In summary, the benefit of IFT by fusing frequent itemsets locally within each transaction,

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

creates candidates that are guaranteed to exist in the database and have non-zero support. The above evaluations demonstrate that IFT is efficient in producing only significant, high-quality rules through its fusion approach.

CONCLUSION

This research presented the Itemset Fusion Technique (IFT) to improve association rule mining by fusing frequent itemsets algorithmically. As demonstrated in the results, IFT reduced database scans by 98.9% and generated 91% fewer rules compared to Apriori. This addresses key limitations of Apriori - costly scanning and exponential candidate generation. IFT shrinks the search space by fusing itemsets within transactions to create focused candidate sets.

While this research focused on comparing IFT and Apriori performance in terms of database scans and rule generation, optimizing IFT execution time is an important future work to be addressed. Techniques like parallelization of the fusion process, optimized data structures, and early pruning would be explored to investigate IFT runtime in the follow up study. However, IFT is a promising new approach that overcomes traditional algorithm challenges through its fusion-based mining technique and further enhancement can strengthen IFT as an efficient, practical technique for actionable association rule mining.

REFERENCES

Abdullah, Z., Pramana Gusman, A., Herawan, T., & Mat Deris, M. (2016, June). MILAR: Mining Indirect Least Association Rule Algorithm. In Ar early version of this paper has been appeared in Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)(pp. 159-166). Springer Singapore. Journal of Computer Science and Information Technology (Vol. 1, No. 1, pp. 60-70).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).

Balasubramanian, C., Duraiswamy, K., & Palanisamy, V. (2010). A Fuzzy Approach to Weighted Temporal Mining Using Automatic Weight Assignment With Reduced Complexities. Journal of Algorithms & Computational Technology, 4(4), 425-441.

Borgelt, C. (2012). Frequent item set mining. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2(6), 437-456.

Chee, C.H., Jaafar, J., Aziz, I.A., Hasan, M.H. & Yeoh, W. (2019) "Algorithms for frequent itemset mining: a literature review". Art. Intel. Rev., vol. 4, pp. 2603-2621.

Cheng, X., Su, S., Xu, S., & Li, Z. (2015). DP-Apriori: A differentially private frequent itemset mining algorithm based on transaction splitting. Computers & Security, 50, 74-90.

Ginting, D., Mawengkang, H. & Efendi, S. (2018). Modification of Apriori Algorithm focused on Confidence Value to Association Rules. Nommensen Int. Conf. on Tech. and Eng. (NICTE). IOP Publishing IOP Conf. Series: Materials Science and Engineering, doi:10.1088/1757-899X/420/1/012125

Han, D. H., & Nv, Z. (2017). Mining Frequent Patterns in Large Scale Databases Using Adaptive FP-Growth Approach. Bonfring International Journal of Industrial Engineering and Management Science, 7(2), 17-20

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

approach. Data mining and knowledge discovery, 8, 53-87.

Herawan, T., Chiroma, H., Vitasari, P., Abdullah, Z., Ismail, M. A., & Othman, M. K. (2015). Mining critical least association rules from students suffering study anxiety datasets. Quality & Quantity, 49, 2527-2547.

Hodijah, A., Setijohatmo, U. T., Munawar, G., & Ramadhanty, G. S. (2022). Multi-threaded approach in generating frequent itemset of Apriori algorithm based on trie data structure. TELKOMNIKA (Telecommunication Computing Electronics and Control), 20(5), 1034-1045.

Karimtabar, N., & Fard, M. J. S. (2020). An Extension of the Apriori Algorithm for Finding Frequent Items. In 2020 6th International Conference on Web Research (ICWR) (pp. 330-334). IEEE.

Khalid, F., Nikoloski, Z., Tröger, P., & Polze, A. (2013). Heterogeneous combinatorial candidate generation. In Euro-Par 2013 Parallel Processing: 19th International Conference, Aachen, Germany, August 26-30, 2013. Proceedings 19 (pp. 751-762). Springer Berlin Heidelberg.

Leung, C. K. S., Mateo, M. A. F., & Brajczuk, D. A. (2008). A tree-based approach for frequent pattern mining from uncertain data. In Advances in Knowledge Discovery and Data Mining: 12th Pacific-Asia Conference, PAKDD 2008 Osaka, Japan, May 20-23, 2008 Proceedings 12 (pp. 653-661). Springer Berlin Heidelberg.

Li, F., Meng, C., Wang, C., & Fan, S.(2023) "Equipment Quality Information Mining Method Based on Improved Apriori Algorithm". J. of Sensors, 2023.

Liu, F., & Wu, G. (2018). An improved Apriori algorithm based on compressed matrix. Journal of Shandong University (Engineering Edition), 48(6), 82-88.

Niu, Z., Fang, G., Zeng, D., & Yuan, H. (2023). An improved Apriori algorithm based on influence weight. In Third International Conference on Computer Vision and Pattern Analysis (ICCPA 2023) (Vol. 12754, pp. 673-679). SPIE.

Patil, S. D., Deshmukh, R. R., & Kirange, D. K. (2016). Adaptive Apriori Algorithm for frequent itemset mining. In 2016 International Conference System Modeling & Advancement in Research Trends (SMART) (pp. 7-13). IEEE.

Raj, S., Ramesh, D., & Sethi, K. K. (2021). A Spark-based Apriori algorithm with reduced shuffle overhead. The Journal of Supercomputing, 77(1), 133-151.

Rana, S., & Mondal, M. N. I. (2021). A Seasonal and Multilevel Association Based Approach for Market Basket Analysis in Retail Supermarket. European Journal of Information Technologies and Computer Science, 1(4), 9-15.

Riszky, A. R., & Sadikin, M. (2019). Data Mining Menggunakan Algoritma Apriori untuk Rekomendasi Produk bagi Pelanggan. Jurnal Teknologi dan Sistem Komputer, 7(3), 103-108. .

Shawkat, M., Badawi, M., El-ghamrawy, S., Arnous, R., & El-desoky, A. (2022). An optimized FP-growth algorithm for discovery of association rules. The Journal of Supercomputing, 1-28.

Singh, K., & Maiti, J. (2020). Mining Frequent Patterns with Temporal Effect: A Case of Accident Path Analysis. In Proceedings of ICETIT 2019: Emerging Trends in Information Technology (pp. 596-603). Springer International Publishing.

Syahrir, M., & Anggrawan, A. (2023). Improvement of Apriori Algorithm Performance Using the TID-List Vertical Approach and Data Partitioning. International Journal of Intelligent Engineering & Systems, 16(2).

Teymoornejad, K. (2018) Improving Apriori Algorithm with Various Techniques, Available at https://www.researchgate.net/publication/329736159_Improving_Apriori_Algori

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.

thm_with_various_ techniques, Accessed on 12/02/2024.

Vivekanandan, S. J., & Gunasekaran, G. (2023). Computation of frequent itemset using matrix based apriori algorithm. Int. J. Exp. Res. Rev, 30, 247-256.

Wu, A., & Liu, A. (2019). Improvement of apriori algorithm based on Boolean matrix reduction. Computer engineering and science, 9.

Wu, X., Xiao, Y., & Liu, A. (2023). Application of improved Apriori Algorithm in Innovation and Entrepreneurship Engineering Education Platform. Scalable Computing: Practice and Experience, 24(3), 609-620.

Zaki, M.J., Parthasarathy, S., Ogihara, M. & Li, W. (1997). New algorithms for fast discovery of association rules. In KDD Vol. 97, pp. 283-289.

Corresponding author: Mathew Tunde Ogedengbe
matt.agedi@uam.edu.ng
Department of Computer Science, University of Agriculture, Makurdi, Nigeria.