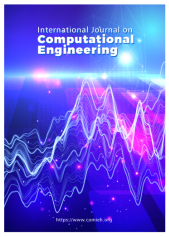
International Journal on Computational Engineering

Journal Homepage www.comien.org/index.php/comien



Analyzing Trie Graph-Based Apriori Algorithm

Ade Hodijah [a,*], Urip Teguh Setijohatmo [a], Irwan Setiawan [b], Hilman Permana [a], Lolla Mariah [a], Devina Lusiana [a]

[a] *Computer and Informatics Engineering, Bandung State Polytechnic, Gegerkalong Hilir Road, Bandung, Indonesia*
[b] *Computer Science and Information Engineering, National Cheng Kung University, University Road, Tainan, Taiwan*
Corresponding author: [*]ade.hodijah@polban.ac.id

***Abstract*—Finding meaningful patterns in big data is one of the active research in data mining. Market basket analysis [MBA] is one of the most helpful modeling techniques in data mining. It involves the mining and analysis of Association Rules. The association rules problem carried by the Apriori algorithm is represented in various issues ranging from market basket analysis, co-occurrence in natural language processing, behavioral similarity, and so on, representing the recognition of co-occurrence tendencies. The problem of the speed of combinatorial itemset acquisition from the Apriori algorithm in previous studies using memory [RAM] is still very possible to be studied. In this study, graph-based technology was chosen based on the study's results that graph algorithms are available [Neo4j'sNeo4j's graph data science or GDS] to support the process of obtaining association rules from the Apriori algorithm. The research used a network-based approach to mine transactional data and get detailed information about the products bought together in the market basket. The network-based approach gives insights into frequently purchased products and products, defined as "Communities." Then, we analyzed these groups of communities using different measures, such as Centrality and PageRank algorithms. From this study, we concluded that finding communities compared to the old traditional method of finding Association Rule is more giving broader information due to graph representation. The combination of the Trie data model and GDS algorithm can be used for further MBA research, specifically in the precision of frequent item set results.**

***Keywords*— Market basket analysis; Apriori algorithm; graph data science.**

Manuscript received 5 Nov. 2024; revised 19 Dec. 2024; accepted 23 Feb. 2025. Date of publication 31 Mar. 2025.

I. INTRODUCTION

Market basket analysis [MBA] allows us to identify 'patterns' in customer purchases by extracting co-occurrences from a store's transactional datasets [1]. It takes its name from the customers who keep all their purchases in a shopping cart [market basket] during shopping. One of the data mining algorithms that supports MBA is the Apriori algorithm. The problem of the speed of obtaining frequent item sets from the Apriori algorithm [2], [3] is still very likely to be studied. Another problem is the use of memory [RAM], which, in previous studies, has limited the number of item sets that can be processed at the combined item sets stage. The limitation is that each stage of the process used RAM. So, another mechanism is needed to be able to produce the frequent item sets, especially in the combined item sets process, where this paper used database storage technology. The use of graph-based technology in association rules algorithms of data mining [4], [5], [6], [7] has been widely used in current studies, and some of them use various graph algorithms provided by Neo4j [8], [9], [10], [11] others used in machine learning [12], [13], [14], [15] where graph learning has demonstrated significant promise in extracting information from various graph types, another introduces new general algorithms for searching unknown graphs under the absolute error model [16].

Verma [4] conducted a study using the Association Rules Networks [ARN] algorithm to visually show the level of association between each item in different communities. This information provides an overview of the items collected together in the dataset. Furthermore, [4] applied various techniques to analyze the communities of products found. ARN is another algorithm for market basket analysis [4]. This algorithm models transactional data as a network. ARN detects communities of items with a network that can find definite and expressive relationships between items, including relationships that are difficult to find with traditional association rules. In detecting communities, [4] did not set a minimum threshold. It can lose information since most items are removed from the network. However, communities of products help us determine which products are bought

together frequently in the market basket, called frequent item sets [4], where a frequent item is any set of items that appears in at least the minimum number of transactions [1]. Therefore, a target product was chosen by a centralized approach [Degree centrality] instead of detecting communities [Louvain]. In this research, the ARN approach used Neo4j's Graph Data Science [GDS] to find the group of products called centralized communities. The objective of this paper is 1] to propose a novel approach named "Trie Graph-based Apriori" to mine the frequent item set from the trie graph-structured data in a more precise result than the preceding work [4] while still implementing the minimum threshold idea from [1] to assess the precision of the approach used Kaggle's groceries dataset [17].

II. Materials and Method

This research used GDS or graph-based algorithms of Neo4j [10] to support the Apriori algorithm and used the groceries dataset from [17], same with [4] paper for the experiment with a total of products is 167 items and 200 transactions, as shown in Table 1 (product network) and Table 2 (Trie).

TABLE I
SAMPLE OF KAGGLE'S GROCERIES DATASET FOR CREATING PRODUCT NETWORK

Member number	Date	Item description
1249	01-01-2014	citrus fruit
1249	01-01-2014	coffee
1381	01-01-2014	curd
…	…	…

Table 1 contains detailed transaction data, recording the date of purchase and the item description for each purchase made by a member [indicated by the member number]. Each row represents one item purchased by a member on a specific date. Member 1249 purchased citrus fruit and coffee on January 1, 2014. In Table 1, member 1249 purchased {citrus fruit} and {coffee} on the same day. It is reflected in Table 2 as a pair of items, {citrus fruit, coffee} for the same member.

TABLE III
SAMPLE OF KAGGLE'S GROCERIES DATASET FOR CREATING TRIE

Member number	Item1	Item2	...
1249	citrus fruit	coffee	
1381	curd	…	
…	…	…	

Four algorithms are used in [4]: Louvain community, Network density, Betweenness centrality, and Page rank. The GDS Neo4j provides these algorithms except for Network density. Therefore, the Network density was not implemented in this paper. Another centrality algorithm was mentioned by [4] to be further used called as Centre-piece subgraph, which was Degree centrality in Neo4j [8].

Fig.1 shows the problem definition in this paper is about knowing the graph-based algorithms for the Apriori algorithm of data mining in generating association rules of market basket analysis [MBA] with the frequent item set forms the association rules. MBA is used to uncover associations between products that are frequently bought together by customers. To enhance this, Graph Data Science is applied. Here, products [Kaggle's groceries] are treated as nodes, and the co-occurrences of products in transactions are represented as edges between nodes in a graph. This allows for a more sophisticated visualization and analysis of product relationships beyond traditional statistical methods, leveraging network structures to identify communities of related items. In the final phase, the results of the graph-based Apriori algorithm are evaluated in terms of precision.

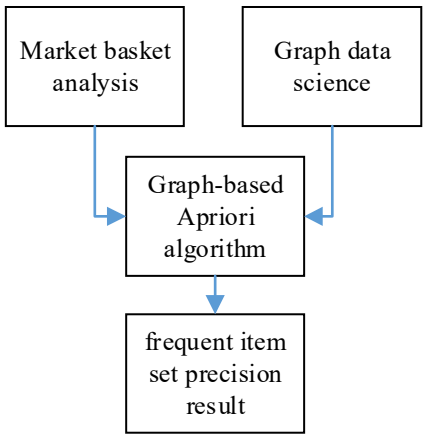


Fig. 1 Research methodology

In [4], identifying strong relationships among the products used the modularity of the Louvain method to calculate the difference between edges in a set of communities extracted from the product network.

Fig. 2 shows the product network algorithm using Cypher. The Cypher query you provided creates a network of products bought together by members on the same date, linking products through a `BOUGHT_WITH` relationship, where code of `WHERE i1 <> i2` ensures that only different items are considered in pairs.

```
LOAD CSV WITH HEADERS FROM
'file:///Groceries_dataset.csv' AS row
WITH row.Member_number AS member, row.Date AS
date, row.itemDescription AS item
MERGE [m:Member {Member_number: member}]
MERGE [i:Item {itemDescription: item}]
MERGE [d:Date {date: date}]
MERGE [d]-[:INCLUDES]->[i]
WITH m, d, COLLECT[i] AS items
MERGE [m]-[:PURCHASED_ON {date: d.date}]->[d]
WITH items, d
UNWIND items AS i1
UNWIND items AS i2
WITH DISTINCT i1, i2, d
WHERE i1 <> i2
MERGE [i1]-[:BOUGHT_WITH {date:
d.date}]->[i2];
```

Fig. 2 Product network algorithm

After community detection, The betweenness centrality and page rank are used in [4] to generate the Apriori's association rules. In this paper, another method was applied to get the frequent item set [Degree centrality] with Trie data model implementation, while the association rules generation used GDS' Personalized page rank and Cypher's set difference operation.

```
MATCH [t:Transaction]-[:CONTAINS]->[i:Item]
WITH t.id AS transactionId, collect[i.name]
AS items
WITH transactionId, items,
apoc.coll.combinations[items,1, size[items]]
AS combinations
RETURN transactionId, items, combinations
ORDER BY transactionId
```

Fig. 3 Trie-based algorithm

Fig. 3 shows the Trie-based algorithm to generate all possible pairwise combinations of items within transactions using the `apoc.coll.combinations` function from the APOC library in Neo4j based on `size[items]`. The code of `[t:Transaction]-[:CONTAINS]->[i:Item]` was used to finds all transactions [`Transaction` nodes] and the items [`Item` nodes] contained within them. One transaction node based on member number.

III. Results and Discussion

This section explains the analysis, design, and implementation of how Apriori conducts frequent item set generation with a graph-based approach [18] according to the research methodology.

A. Analysis

Market Basket Analysis [MBA] with the Apriori algorithm approach is carried out in two steps: the formation of frequent item sets and the determination of association rules from a transaction dataset. The working principle of this research used the GDS approach, which is communities [Louvain algorithm] and centralities [Degree centrality algorithm]. The frequent items are the connection of items purchased together, whereas the graph was called a network of products [19], [5]. The following [Fig. 4] is an illustration of the formation of a network of products from four transactions: Transaction T1: {p1, p2, p4} T2: {p1, p2, p3} T3: {p3, p11} T3: {p7, p8, p11} T4: {p8, p11, p13}.

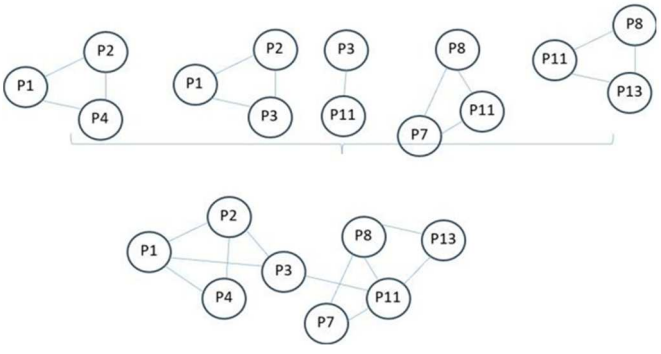


Fig. 4 Product Network

The network of products is not enough to describe the connection of items often purchased together, so it is necessary to separate groups often purchased together. In the graph, this is done by forming communities where each internal community has a close connection between them. The approach to identifying communities from the network of products can be done in several ways, such as biconnected components [the strongly connected components algorithm] and modularity-based [the Louvain algorithm]. The Louvain method obtains partitions/components with high modularity in a short time [4].

Each community has a level of closeness between its members, and the maximum closeness occurs when the community is in the form of a complete graph. The level of closeness is measured by network density so that this stage can select only a few communities that will be forwarded to determine Association Rules [4], while in this research called as the frequent item was measured by degree centrality's score as the threshold. It is possible to determine Association Rules by directly applying two centrality methods in a pipeline, namely Degree centrality and PageRank, but the precise results have yet to be discovered. Therefore, this study considers the possibility of generating association rules using the GDS algorithm.

From the explanation above, the computational model needed to obtain Association Rules to support this research. The research used two data models to save transaction data in Neo4j, which are product network [19], and Trie data structure [2], [3]. The tree model used also by [20] for implementing graph-based search solution. The complete models used in this paper are:

- Model-1: Agrawal's Apriori
- Model-2: Louvain [product network] and Betweenness
- Model-3: Louvain [product network] and PageRank
- Model-4: Degree [product network] and Personalized PageRank
- Model-5: Louvain [trie] and Betweenness
- Model-6: Louvain [trie] and PageRank
- Model-7: Degree [trie] and Personalized PageRank
- Model-8: Degree of Neo4j's GDS [trie] and Cypher's set difference operation

Model-1 used [1] algorithms and an online application from [21] to generate frequent item sets. For Model-2 and Model-3 used product network [association rules network], the research results of [4], where used R and Python to process Louvain, Betweenness, and PageRank algorithms for generating association rules, while this paper used Neo4j's GDS algorithm [10]. The last four models [Model-4, Model-5, Model-6, Model-7] used Trie. The Model-8 was different from others. It was based on Agrawal's mechanism, where the frequent item set generated by Degree of Neo4j's GDS with the Trie as graph data modeling and the association rules were formed using Cypher's set difference operation.

The frequent item set generated by Model-1 [21] compared to other models. The association rules on Model-2, Model-3, Model-5 and Model-6 are based on all frequent itemsets in a particular community. In contrast, Model-4 and Model-7 were based on sourceNodes that can generate association rules of Model-1. Meanwhile, for Model-8, the association rules are generated based on the frequent item set.

B. Design and Implementation

The product network graph is shown in [Fig.5], the product network example of three transactions: Customer1{Product1, Product2, Product3}, Customer2{Product2, Product4}, and Customer3{Product4} [10] using Neo4j. Another model used the Trie data structure [2], [3] to generate the product network, where a vertex is the item's combination of products bought

together in the market basket dataset. A Trie graph is shown in Fig.6.

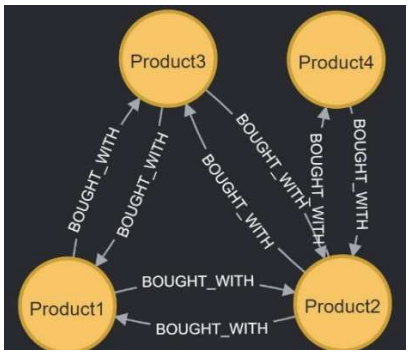


Fig. 5 Product network data model

Table 3 shows the frequent item sets from the Model-1 with threshold >2. The most frequent item set was items {[rolls/buns, wholemilk]}, which had eight support counts and two item set combinations. Still, the association rules generated based on the largest item set combination of frequent item set, which consists of 3 item set combinations and had a support count value of 2, which are {sausage, wholemilk, yogurt} and {chocolate, rolls/buns, wholemilk}.

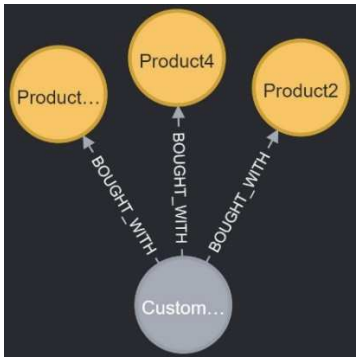


Fig. 6 Trie data model

TABLE III
FREQUENT ITEM SET OF THE MODEL-1

Frequent item set	Combination	Support
{[sausage,wholemilk,yogurt], [chocolate,rolls/buns,wholemilk]}	3	2

As mentioned in [4] paper, two scenarios were examined in generating the association rules for the Model-2 and the Model-3: betweenness centrality [Model-2] and PageRank [Model-3]. The result of association rules generation for each scenario was chosen from a certain community detected by the Louvain algorithm. Five communities were created from 167 items created by Louvain algorithms, as shown in Table 4.

Compared to Model-1 results [Table 3], Community-1 and Community-2 contain wholemilk and yogurt, showing a potential preference for dairy products, but there is no mention of sausage, chocolate, or rolls/buns in this community. In Community-3 and Community-4, wholemilk, yogurt, and sausage appear in both lists, and their order relative to each other is consistent, with whole milk coming before yogurt, and yogurt coming before sausage. Community-5 is the only community where all five items [whole milk, yogurt, sausage, chocolate, and rolls/buns] appear together, suggesting a stronger connection or preference for these items as a group in this community. In addition to these, there are other products like bottled water and soda, showing a much broader and more diverse consumption pattern. The relationship between these five items strengthens progressively across the communities. Starting with only wholemilk and yogurt in simpler communities [1 and 2], sausage appear in Community-3 and Community-4, with rolls/buns and chocolate only becoming prominent in the most diverse Community-5.

TABLE IIIV
SAMPLE OF THE MODEL-2 RESULTS

Community ID	Items et count	Items
1	47	{[wholemilk,yogurt,…]}
2	80	{[wholemilk,othervegetables,yogurt,…]}
3	113	{[wholemilk,othervegetables,yogurt,soda,sausage,…]}
4	143	{[wholemilk,othervegetables,soda,yogurt,sausage,…]}
5	167	{[wholemilk,rolls/buns,othervegetables,soda,yogurt,bottledwater,sausage,…,chocholate,…]}

The following is a list of frequent itemset searches of the Model-1 that were successfully obtained or became one of the items in the community generated by the Model-2, as shown in Table 5.

TABLE V
FREQUENT ITEM SET OF THE MODEL-2

Community ID	Combination	Frequent item set
1	2	{wholemilk,yogurt}
2	2	{wholemilk,yogurt}
3	3	{sausage,wholemilk,yogurt}
4	3	{sausage,wholemilk,yogurt}
5	5	{sausage,wholemilk,yogurt,chocolate,rolls/buns}

In the Model-2 [Table 3], the frequent item set {sausage, wholemilk, yogurt} was also found in community-3 and community-4 with the order of occurrence for each item in community-3 was {5,1,3} and community-4 was {5,1,4}. Meanwhile, the item set {sausage, wholemilk, yogurt, chocolate, rolls/buns} appeared in the list of items in community-5, where five items formed the two frequent item set generated by the Model-1, as seen in Table 3. The order of occurrence for each item in community-5 was {7,1,5,32,2}.

TABLE VI
FREQUENT ITEM SET OF THE MODEL-3

Community ID	Combination	Frequent item set
1	2	{wholemilk,yogurt}
2	2	{wholemilk,yogurt}
3	3	{sausage,wholemilk,yogurt}
4	3	{sausage,wholemilk,yogurt}
5	5	{sausage,wholemilk,yogurt,chocolate,rolls/buns}

In the Model-3 [Table 6], the frequent item set {sausage, wholemilk, yogurt} was also found in community-3 and community-4, where the order of occurrence for each item in community-3 was {5,1,4} and community-4 was {7,1,4}. The number of communities, item count, and item list for each community from the Model-3 were similar to the results from the Model-2 but differed in the order of the items. For example, the 3rd order in the Model-2 was items {bottledwater}, while in the Model-3 it was items {tropicalfruit}. The five communities generated by Louvain are then PageRanked. Table 4 shows the results of searching the items list from the frequent itemset for each community generated by the Model-3 based on the five-item lists from the two association rules generated by the Model-1, namely {sausage, wholemilk, yogurt, chocolate, rolls/buns}.

The Personalized PageRank algorithm was used in the Model-4, unlike the Model-3 used the PageRank algorithm from [4]. The Personalized PageRank is a variation of PageRank, which is biased towards a set of sourceNodes configuration parameters [8].

Table 7 shows the results of the Personalized PageRank based on two frequent item sets generated by the Model-1, namely {sausage, wholemilk, yogurt} or {chocolate, rolls/buns, wholemilk}. Unlike the Model-3, which only found {sausage, wholemilk, yogurt}, in the Model-4 [Table 5], only one of the two frequent item sets from the Model-1, {chocolate, wholemilk, rolls/buns} was obtained when personalized with sourceNodes {chocolate}. Although the order of items is not the same as the frequent item from the Model-1 {chocolate, rolls/buns, wholemilk}, both models obtained candidate association rules that indicate three items were purchased together, namely {chocolate},{wholemilk}, and {rolls/buns}. In addition, the order of items from the Degree centrality for {chocolate, rolls/buns, wholemilk} was {31, 3, 1}. Based on this order, it can be seen that not all items are included in the items that occupy the top 3 or 5 degree values, where chocolate was ranked 31st from the results of the Degree centrality calculation. If traced using the Personalized PageRank, for example, with source nodes {sausage}, frequent item set {sausage, whole milk, yoghurt} can be obtained when hopping five times until {yogurt} was found, that got {sausage, wholemilk, other vegetables, rolls/buns, soda, yogurt}.

TABLE VII
FREQUENT ITEM SET OF THE MODEL-4

Source nodes	Frequent item set
{sausage}	{sausage,wholemilk,othervegetables}
{wholemilk}	{wholemilk,othervegetables,rolls/buns}
{yogurt}	{yogurt,wholemilk,othervegetables}
{chocolate}	{chocolate,wholemilk,rolls/buns}
{rolls/buns}	{rolls/buns,wholemilk,othervegetables}

However, the number of communities formed for the Model-5 and the Model-6 was the same as the result of item sets generating combinations, namely 2^n - 1. Therefore, the Trie data model was not appropriate to be used in the Louvain algorithm for obtaining the association rules.

Furthermore, in the Model-7, the Trie data model can be applied to the Degree centrality algorithm, where frequent item sets from the Model-1 {[sausage, wholemilk, yogurt], [chocolate, rolls/buns, wholemilk]} were also found after conducting the Degree centrality algorithm, where these frequent item sets were on the first and the second because it had the longest item set and the highest degree's score. However, the Personalized PageRank was not appropriate to use after the Degree centrality algorithm with this Trie data model because the combination generated from each item had no connection with other combinations of items. The only relationship was between the Transaction and Combination nodes, unlike the Product Network data model, where there were relationships among items.

Finally, in the Model-8, a frequent item set was obtained consisting of 3 combinations of items like the Model-1, namely: {[wholemilk, rolls/buns, chocolate], [wholemilk, sausage, yogurt]}. Although the order of the frequent item set list from the Model-8 was different from the results of the frequent item set of the Model-1, the combination of items produced by the Model-1 and the Model-8 were the same, where the first frequent consisted of items: {[yogurt, [sausage], [wholemilk]} and the second frequent consisted of items: {[rolls/buns], [wholemilk], [chocolate]}.

Inspired by [22], Table 8 shows 12 candidate association rules created based on two frequent item sets of the Model-8. In general, association rules have the form LHS → RHS, where LHS and RHS are a label of item sets for association rules retrieval]. If every item in LHS is present in a transaction, then the items in RHS are also present in the transaction.

TABLE VIII
THE CANDIDATE ASSOCIATION RULES OF THE MODEL-8

LHS	RHS
{chocolate}	{wholemilk,rolls/buns}
{wholemilk}	{rolls/buns,chocolate}
{rolls/buns}	{wholemilk,chocolate}
{rolls/buns,chocolate}	{wholemilk}
{wholemilk,chocolate}	{rolls/buns}
{wholemilk,rolls/buns}	{chocolate}
{yogurt}	{wholemilk,sausage}
{wholemilk}	{sausage,yogurt}
{sausage}	{wholemilk,yogurt}
{wholemilk,sausage}	{yogurt}
{sausage,yogurt}	{wholemilk}
{wholemilk,yogurt}	{sausage}

The algorithm to generate candidate association rules is shown in Fig. 7. Given a frequent itemset [e.g., {A, B, C}], the algorithm [Fig. 7] can generate association rules by finding subsets of the itemset and then computing the difference between the itemset and each subset. The rules like LHS: {A, B} → RHS: {C} or LHS: {A} → RHS: {B, C}, etc. The code of `WITH a1, currentArray, [n IN a1.name WHERE NOT n IN currentArray] AS difference` calculates the set difference between the full frequent itemset [`a1.name`] and the current subset [`currentArray`]. The difference will represent the items excluded from the current array, which becomes the rule's right-hand side [consequent] or RHS. After that, `WHERE size[difference] > 0` ensures that only valid rules are generated where there is something on the left-hand side or LHS to be the antecedent [i.e., the difference is non-empty]. Finally, `MERGE [:LEFT {name: currentArray}]-[:RULES]->[:RIGHT {name: difference}]`

creates a new node representing the left-hand side [antecedent] and another representing the right-hand side [consequent], then creates a RULES relationship between them.

```
MATCH [i:frequentItemset]
WITH i.name AS items
WITH items, apoc.coll.combinations[items, 1,
size[items]] AS arrays
UNWIND arrays AS currentArray
MATCH [a1:frequentItemset]
WITH a1, currentArray, [n IN a1.name WHERE
NOT n IN currentArray] AS difference
WHERE size[difference] > 0
MERGE [:LHS {name: currentArray}]-
[:RULES]->[:RHS {name: difference}]
DELETE a1
```

Fig. 7 Candidate association rules algorithm

C. Experiments

Table 9 shows the experimental results of all models. Model-5, Model-6, and Model-7 algorithms have yet to obtain frequent item set. The frequent item set {sausage, wholemilk, yogurt} and {chocolate, rolls/buns, wholemilk} of Model-1 were also found on Model-8. Whereas Model-2 and Model-3 only had {sausage, wholemilk, yogurt} on community-3 and community-4 with {sausage} ranked 5th [Model-2] and {sausage} ranked 7th, while Model-4 only had {chocolate, wholemilk, rolls/buns} on sourceNodes {chocolate}.

TABLE IX
EXPERIMENTAL RESULTS

Scenario	Frequent item set
Model-1	{sausage,wholemilk,yogurt}, {chocolate,rolls/buns,wholemilk}
Model-2	{sausage,wholemilk,yogurt}
Model-3	{sausage,wholemilk,yogurt}
Model-4	{chocolate,wholemilk,rolls/buns}
Model-5	-
Model-6	-
Model-7	-
Model-8	{wholemilk,rolls/buns,chocolate}, {wholemilk,sausage,yogurt}

Table 9 explains that in Model-1, there is a strong purchasing pattern between dairy products [wholemilk] and processed meats [sausage] and yogurt, as well as between dairy [wholemilk] and bakery products [rolls/buns] with chocolate. Model-2, Model-3, and Model-4 are similar to Model-1, where the purchasing pattern involving wholemilk, sausage, and yogurt is consistent [Model-2 and Model-3]. Meanwhile, the scenario of Model-4 shows the same combination found in Model-1, with wholemilk being purchased alongside chocolate and rolls/buns. The Model-8 is the best, where the scenario repeats patterns from Model-1, showing strong associations between wholemilk and other products like rolls/buns, chocolate, sausage, and yogurt. The last several scenarios [Model-5, Model-6, and Model-7] show no significant purchasing patterns are identified. The combination of wholemilk with yogurt, sausage, rolls/buns, and chocolate frequently appears across several scenarios, indicating that wholemilk is central to the strongest purchasing patterns. The wholemilk is always the first item in all communities generated by the Louvain algorithm, as shown in Table 5.

IV. CONCLUSION

In this study, a new traversal method for a priori with trie was found. The graph-based Apriori algorithm in this research successfully obtained a model [Model-8] that can produce the entire list of the same frequent item set using the Apriori algorithm from Agrawal [Model-1]. Furthermore, candidate association rules have also been successfully obtained in the Model 8. However, the formation of each LHS → RHS rule must be based on fulfilling the minimum threshold value of confidence, which has yet to be done in this research. In the Model-2, the Model-3, and the Model-4, only one of the frequent item lists of the Model-1, but the list of items from the Model-1 was obtained after the hop operation was performed on the Model-4, such as the item {sausage, wholemilk, othervegetables, rolls/buns, soda, yogurt}. The Model-4 provided an alternative or other information on how the list of items {othervegetables and rolls/buns} was obtained when searching for frequent item sets {sausage, wholemilk, yogurt} with source nodes {yogurt} using the Personalized PageRank algorithm. It means the graph-based data mining model in this paper gives more information about other item set combinations by using the Personalized PageRank algorithms. Further, Louvain algorithm results needed to use another measure to calculate ranking position in the Model-2, the Model-3, and the Model-4, for example, Normalized Discounted Cumulative Gain [NDCG] compared to the Agrawal's association rules.

ACKNOWLEDGMENT

Bandung State Polytechnic supported this research on the Penelitian Madya Utama [PMU] funding program.

REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1993, pp. 207-216.

[2] A. Hodijah and U. T. Setijohatmo, "Analysis of frequent itemset generation based on trie data structure in Apriori algorithm," *Telkomnika*, vol. 19, no. 5, pp. 1553-1564, 2021.

[3] A. Hodijah, U. T. Setijohatmo, and G. M. GSR, "Multi-threaded approach in generating frequent itemset of Apriori algorithm based on trie data structure," *Telkomnika*, vol. 20, no. 5, pp. 1034-1045, 2022.

[4] N. Verma and S. Orlando, "Market basket analysis with network of products," Univ. Venice, Tech. Rep., 2017. [Online]. Available: http://dspace.unive.it/handle/10579/10564

[5] G. Manuel, I. Caldas, Y. Jesus, and G. Lopez, "A product network analysis using Apriori algorithm for extending the market basket in retail," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2023, pp. 2025-2034.

[6] R. Chen et al., "Hypergraph-clustering method based on an improved Apriori algorithm," *Appl. Sci.*, vol. 13, no. 19, 2023.

[7] Z. Zhang, J. Huang, and Q. Tan, "Association rules enhanced knowledge graph attention network," *Knowl.-Based Syst.*, vol. 239, 2022.

[8] Neo4j, "Neo4j graph data science," 2023. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/algorithms/

[9] S. Sen et al., "Recommendation of influenced products using association rule mining: Neo4j as a case study," *SN Comput. Sci.*, vol. 2, no. 2, 2021.

[10] J. Wang, G. Li, and T. Ai, "Revealing association rules within intricate ecosystems: A spatial co-location mining method based on Geo-Eco knowledge graph," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 133, Jan. 2024.

[11] J. Wang et al., "Graph-based spatial co-location pattern mining: Integrate geospatial analysis and logical reasoning," *Int. J. Digit. Earth*, vol. 17, no. 1, pp. 1-23, 2024, doi: 10.1080/17538947.2024.2390434.

[12] H. Ko et al., "Machine learning and knowledge graph based design rule construction for additive manufacturing," *Addit. Manuf.*, vol. 37, p. 101620, 2021, doi: 10.1016/j.addma.2020.101620.
[13] S. Mežnar et al., "Ontology completion with graph-based machine learning: A comprehensive evaluation," *Mach. Learn. Knowl. Extr.*, vol. 4, no. 4, pp. 1107-1123, 2022.
[14] B. Alwasel et al., "Leveraging graph-based representations to enhance machine learning performance in IIoT network security and attack detection," *Appl. Sci.*, vol. 13, no. 13, 2023.
[15] C. Jiao, "Optimization algorithm of big data mining based on machine learning model," *Rev. Intell. Artif.*, vol. 34, no. 1, pp. 51-57, 2020.
[16] A. F. DePavia, E. Tani, and A. Vakilian, "Learning-based algorithms for graph searching problems," in *Proc. Mach. Learn. Res.*, vol. 238, 2024, pp. 928-936.
[17] Kaggle, "Groceries dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset.
[18] C. Cummins et al., "PROGRAML: A graph-based program representation for data flow analysis and compiler optimizations," in *Proc. Mach. Learn. Res.*, vol. 139, 2021, pp. 2244-2253.
[19] H. K. Kim, J. K. Kim, and Q. Y. Chen, "A product network analysis for extending the market basket analysis," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7403-7410, 2012, doi: 10.1016/j.eswa.2012.01.066.
[20] V. A. Darvariu, S. Hailes, and M. Musolesi, "Solving graph-based public good games with tree search and imitation learning," *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 1739-1751, 2021.
[21] "Apriori algorithm demo," 2023. [Online]. Available: https://www.philippe-fournier-viger.com/tools/Apriori_algorithm_demo.php
[22] L. Chen, V. Ravichandran, and A. Stolcke, "Graph-based label propagation for semi-supervised speaker identification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2583-2587.