International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 04 | Apr 2023 www.irjet.net p-ISSN: 2395-0072

Rider Ready- Bike parts purchase portal with smart cart

Harshit Parasrampuria[1], Manali Rathod[2], Pratik Sherlekar[3], Pranav Temkar[4] ,Prof. Amruta Sankhe[5]

[1]Harshit Parasrampuria, Dept. of Information Technology Engineering, Atharva College of Engineering [2]Manali Rathod, Dept. of Information Technology Engineering, Atharva College of Engineering [3]Pratik Sherlekar, Dept. of Information Technology Engineering, Atharva College of Engineering [4]Pranav Temkar, Dept. of Information Technology Engineering, Atharva College of Engineering
[5]Prof. Amruta Sankhe, Dept. of Information Technology, Atharva College of Engineering, Maharshtra, India

---

Abstract - *This paper presents the design and implementation of a bike parts purchasing portal that utilizes the Apriori algorithm for cart recommendations. The association rules were trained by minimum support=3, minimum confidence=20% and minimum lengths were 9,7 and 6,5. The portal provides an easy and convenient platform for customers to purchase bike parts and also has the option to book for installation services. The Apriori algorithm is employed to analyze customer purchase behavior and suggest items that are frequently bought together, thus enhancing the customer shopping experience. The chatbot is integrated to provide instant support for customers and assist with their purchases. The portal is user-friendly and aims to simplify the process of purchasing bike parts, making it a one-stop-shop for all biking needs.*

Key Words: **Bike, apriori, chatbot, workshop, website, filtering, php, comparative analysis.**

1. INTRODUCTION

Riding a motorcycle is a passion for many and requires constant maintenance to keep it in top shape. With the increase in the number of riders, there is also a growing demand for motorcycle parts and accessories. However, finding the right parts and making the purchase can often be a time-consuming and challenging task. To address this problem, a new portal for purchasing motorcycle parts has been developed. This portal utilizes the Apriori algorithm for cart recommendations and has a chatbot for customer support. It also provides the option to book for part fittings, making it a comprehensive solution for all motorcycle-related needs. The use of the Apriori algorithm ensures that customers receive relevant and personalized recommendations, streamlining the purchasing process. The integration of a chatbot provides instant support and enhances the overall customer experience. This paper aims to present the design and implementation of this unique portal, offering a one-stop-shop for all motorcycle parts and accessories.

1.1-Need

India has the largest number of two wheeled vehicles in the world and hence the largest number of consumers who frequently need to replace certain parts. Going to a company garage is not always feasible so people rely on independent motor garages. It frequently leads to garages using subpar components and cause accidents and increased costs. Our portal will ensure that the parts are purchased from the actual manufacturers who made their vehicles. While purchasing the product the customers can also select the vendors for the garages to replace the parts that they purchased. In this way vendors get new business and the customers get assurance that the parts they purchased are authentic. The vendors who have offline stores get a chance to integrate their business with tech and become a part of the modern tech business world.

2. Application

Every person who uses a bike can use the portal to purchase parts that require periodic replacement like engine oil, brake wires and many more. People who want to purchase spare parts can purchase individual products and keep them for emergency. The smart cart option gives users a recommendation on which products are frequently purchased with their products so they have a better cart of products that is learnt from collective intelligence. The vendors based on location can be booked for consultation or to fit the parts that have been purchased. Portal is also a vendor aggregator where the vendors get their own login ids and they can get businesses and accept or reject the order themselves. They can also view the orders of other vendors which will promote healthy competition and result in competitive prices for the users.

3. Literature Survey

A. Car Recommendation System for Dealers in Different European Countries

This paper discusses the creation of a car recommendation system for various European countries

using collaborative filtering (CF). Unlike content-based filtering, which relies on the attributes of an item, CF recommends items based on each user's historical information. This approach was chosen due to its advantage in personalized recommendations. The user model was built using data on brand, model, car sales, and country. The model calculates the similarity between business pairs to provide recommendations. Our model successfully predicted the top 5 selling cars in each country with a mean square error (MSE) of 8.086. When predicting and testing cars with a rank higher than 3, the MSE was 0.4241379. However, due to a lack of data, the MSE of the full model was slightly high.

B. Multi-Context Recommendation Systems (CARS) in Autonomous Driving and Other Applications

Recommendation systems (RS) play a crucial role in enhancing user experience by providing instantaneous suggestions for desired items. Context-aware recommendation systems (CARS) aim to optimize the RS further by considering various contextual factors such as location and time. By incorporating multi-contexts, CARS add more nuance to the process of predicting items, resulting in more personalized recommendations. This paper explores the foundations of recommendation systems, including categories, evaluation metrics, datasets, and challenges. Additionally, the paper highlights the effectiveness of CARS in autonomous driving by presenting three CARS models and their experimental results. The study shows that multi-contexts provide drivers with more personalized options, allowing them to make intelligent decisions while on the road.

C. ROS2-based Gadgets for Motorcyclists

In this paper, we introduce a collection of motorcycle gadgets intended to enhance the safety, comfort, and user experience of motorcyclists. The implementation of these gadgets was tested in a simulation environment, and the results of these tests are presented herein. The set of gadgets includes a smart helmet, a haptic jacket, and a pair of haptic gloves. The smart helmet is equipped with a pair of smart glasses and a headset, while the haptic jacket features vibration motors and LED indicators. Additionally, the haptic gloves each include a vibration motor.

D. Smart Security Systems for motorbikes

The objective of this paper is to present the development of a universal algorithm for an intelligent safety system designed for motorcycles. Additionally, the paper discusses the creation of a prototype to enable the final implementation and tuning of the algorithm. The prototype comprises an IoT Kit that uses an ARM M3+ microcontroller, a GPS module for position evaluation, and both GSM and Bluetooth modules for communication. The article also provides a detailed description of the design and function of an Android application, which serves as a GUI for Bluetooth communication between a smartphone and the safety prototype.

4. Proposed approach

The proposed approach for the motorcycle parts purchase portal involves the integration of multiple technologies to provide an enhanced customer experience. Firstly, user has to log in using the user id or generate a new one. After each transaction on the users side, data is collected and analyzed using the Apriori algorithm to generate item-item association rules and make personalized cart recommendations. There will be an option to log in for the admins to manage the website and prevent fraud. The catalogue will be editable directly through the interface and not dependent on the backend team for every minor update. Additionally, the portal offers an option to book part fittings offline, allowing customers to receive professional installation services. These orders will be approved or rejected by the vendors themselves through specialized login ids. Customers also have an option to book vendors to fit the parts they have purchased. By bringing together these features, the proposed approach aims to create a comprehensive solution for purchasing motorcycle parts and accessories, streamlining the process and providing a convenient platform for customers and vendors who will get new customers.

5. Methodology

A. Dataset

The dataset is collected from purchases made by users on the website. The transactions that were finalized were stored in a different database, When the database grew to sufficient number of transactions, the transactions were extracted to a csv file. This csv file contained the transactions which will be used to generate the association rules.

```
store_data = pd.read_csv('train.csv')
```

```
store_data.head()
```

	Unnamed: 0	engine	wire	accelerator	pad	fuel	light	brake	vendor
0	1	40	49	70	88	103	111	132	140
1	2	27	55	65	80	91	113	137	142
2	3	36	43	65	86	94	111	136	139
3	4	27	48	74	84	104	122	134	139
4	5	28	45	61	77	104	122	132	139

Fig. Viewing the dataset

B. Pre-processing

The data was pre-processed by removing incomplete transactions and transactions that were fraudulent as well as duplicate in nature. The rows were numbered and the columns were renamed so that the data labels can be selected at the time of generating rules.

C. Apriori Algorithm

Apriori algorithm predefined class apyori was used to generate the association rules. The function was predefined and hence the support, confidence and length of the association rules needed to be selected according the requirement. The support was 3, confidence was 20% and the lengths were 9,7,6,5 for different lengths of association rules generation.

```
[ ] records = []
    for i in range(0,num_records):

        records.append([str(store_data.values[i,j])for j in range(0,9)])

[ ] association_rules = apriori(records, min_support=0.019, min_confidence=0.2, min_length=8)
    association_results = list(association_rules)
```

Fig. code for the apriori algorithm

D. Results

The results were 12 association rules. 5 rules of length 9, 3 rules of length 7, 2 rules of length 6 and 2 rules of length 5. The rules were frequent item datasets. Hence the results were ready and entered into the priority table. Due to this, the recommended products according to previous data are now displayed before other products.

69	41	47	68	85	105	112	130	139
70	35	56	71	75	103	117	131	142
71	35	52	69	82	102	118	138	140
72	28	43	65	80	104	107	131	139
73	34	44	74	81	102	107	131	139
74	42	57	60	75	94	113		
75	42	55	73	82	91	119		
76	38	45	66	90	95	121		
77	38	47	70	89	101			
78	36	53	61	87	105			
79	28	54	66	79				
80	28	48	73	88				

Fig. Frequent itemsets

5. ALGORITHMS USED

5.1 Apriori Algorithm

Apriori is a popular data mining algorithm that is used for discovering frequent itemsets and association rules in large datasets. The algorithm is based on the Apriori principle, which is used to prune the search space efficiently and reduce the number of candidate rules that need to be evaluated. The algorithm consists of two phases: the candidate generation phase and the candidate evaluation phase. In the candidate generation phase, the algorithm generates a set of candidate itemsets of length k based on the frequent itemsets of length k-1. This is done by joining each frequent itemset with itself and pruning any resulting itemsets that are not frequent. The algorithm then counts the support of each candidate itemset, which is the number of transactions that contain the itemset. If the support of a candidate itemset is greater than or equal to the minimum support threshold, it is added to the set of frequent itemsets. In the candidate evaluation phase, the algorithm generates association rules from the frequent itemsets. For each frequent itemset, the algorithm generates all possible non-empty subsets and computes the confidence of each rule. The confidence of a rule is the ratio of the support of the rule's antecedent and consequent to the support of the rule's antecedent. If the confidence of a rule is greater than or equal to the minimum confidence threshold, the rule is added to the set of association rules. Apriori has some limitations, such as high computational complexity and sensitivity to the minimum support threshold. To address these limitations, several variations and improvements of the algorithm have been proposed, such as FP-growth and ECLAT. However, Apriori remains a popular and effective algorithm for association rule mining in various domains.

A. Apriori Principle

The Apriori principle is a concept used in the Apriori algorithm, which is a popular data mining algorithm for discovering association rules in large datasets. The principle is based on the observation that if a set of items is frequent, then all of its subsets must also be frequent. In other words, if a group of items occurs frequently together in a dataset, then any subset of that group must also occur frequently. This principle is important because it allows the algorithm to efficiently prune the search space and reduce the number of candidate itemsets that need to be considered. By only considering frequent itemsets, the Apriori algorithm is able to focus on the most relevant relationships between items, which greatly improves its efficiency and effectiveness.[5][6][7][8]

B. Candidate itemset

A candidate itemset is a set of items that is generated during the first phase of the Apriori algorithm. This phase is called

the candidate generation phase, and it involves generating all possible combinations of items up to a certain length. The candidate itemsets are then evaluated in the second phase to determine their frequency and filter out infrequent itemsets. The remaining frequent itemsets are used to generate association rules. The size of the candidate itemset grows rapidly as the length of the itemset increases, and this can lead to a combinatorial explosion. To address this issue, the Apriori algorithm uses the Apriori principle, which states that if an itemset is frequent, then all its subsets must also be frequent. This principle enables the algorithm to efficiently prune the search space and reduce the number of candidate itemsets that need to be evaluated.[9][10]

C. Frequent itemset

In frequent itemset mining, a frequent itemset refers to a set of items that frequently appear together in a transactional database. It is used as the basis for generating association rules in data mining. The frequency of an itemset is measured by the number of transactions that contain all the items in the set. A frequent itemset can be of any size, from a single item to several items. Finding frequent itemsets is an important step in market basket analysis and other applications, as it helps identify patterns and relationships between items. The Apriori algorithm, for example, generates frequent itemsets in a systematic way, by first identifying all frequent single items and then gradually extending them to larger itemsets.[11][12]

5.2 Priority Algorithm

A priority algorithm is a commonly used technique for assigning importance or significance to a set of items such as tasks, projects, or research papers. It works by evaluating different factors such as urgency, potential impact, and the amount of effort required to complete a task. By analyzing these factors, the algorithm assigns a priority value to each item which helps individuals or organizations to determine which items require the most attention and resources. For instance, in project management, a priority algorithm is used to prioritize tasks and allocate resources efficiently. Urgent tasks are given a higher priority and addressed first, while tasks that can wait are assigned a lower priority. Similarly, in academic research, a priority algorithm can help researchers prioritize research papers to read and evaluate by considering factors such as relevance to their research, the reputation of the authors, and the significance of the findings. In general, the use of a priority algorithm helps individuals and organizations to manage their resources effectively and achieve their goals efficiently. By focusing on the most important or urgent items, they can increase productivity, meet deadlines, and ultimately achieve success. The specific factors considered in a priority algorithm may vary depending on the context and goals of the individual or organization, but the overall objective remains the same - to assign priority values that reflect the relative importance of each item.[13][14][15][16][17][18]

6. Architecture

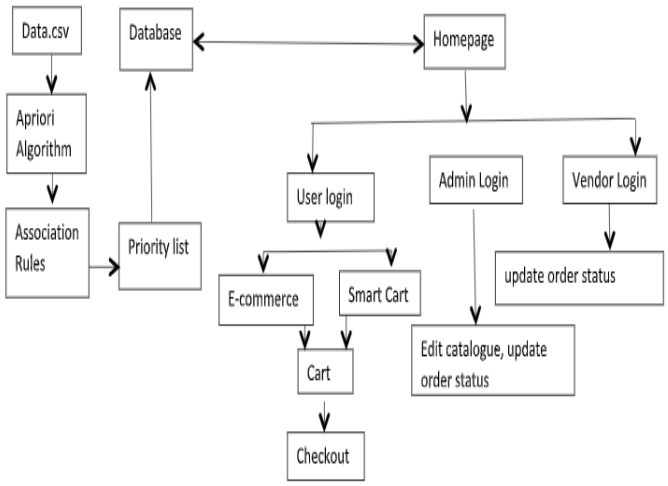

Fig. architecture of the project

7. Methodology

A website using php for the backend was created. Navigation bar items such as ecommerce, user login, vendor login and admin login were added. The backend was done in sql where multiple tables were created to store the permanent data such as the products, categories and user accounts. Every single data entry in the tables can be edited or deleted with the exception of the transaction saving database which stores the user transactions to train apriori algorithm. The admin portal has a graphical interface to edit the product description and pictures. The different categories can be switched off from the admin login itself in case the product is not is stock. The chatbot and the payment gateway were added separately.[3][4]

The apriori algorithm was used to generate the frequent association rules based on a database train.csv generated by extracting the transaction data from bikw_transaction.sql.

The code was written in python 3 and the rules were mined with a minimum support of 0.019 and confidence of 20%.

13 rules were generated and the size of the itemsets varied between 4-8 items. The results were then extracted in a separate csv files and integrated into the portal.[1][2]

8. Software Requirements

1. Operating System: Windows 10 or later

2. Server: XAMPP 7.4

3. Programming Languages:

- PHP 7.0 and above
- HTML 3.0 and above
- CSS
- JavaScript
- Python 3.10

4. File Formats:

Comma Separated Values (CSV)

9. Results

The website starts with a start customizing page which has the smart cart recommendations.

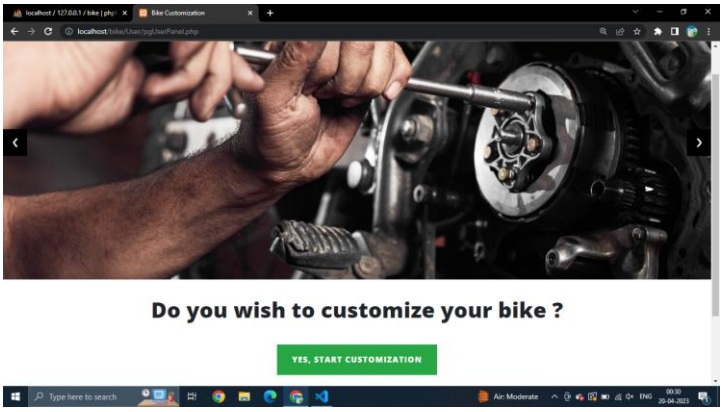

Fig. Homepage

The products are shown according the budget selected by the user. The budget can be selected using a simple dropdown menu.

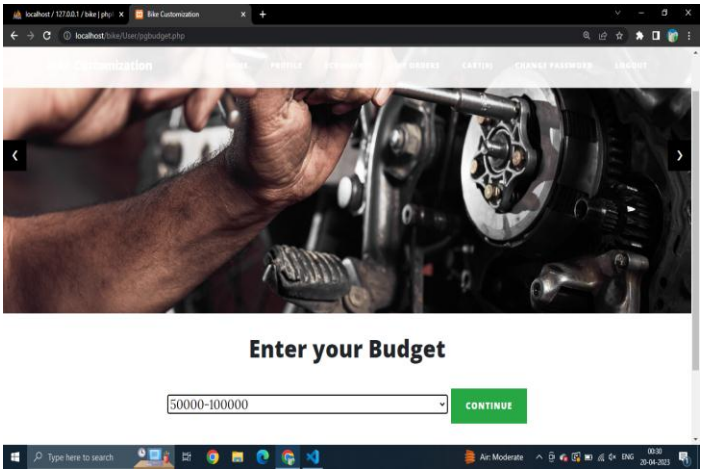

Fig. Budget entry

The login page for admin, user and vendor are different. Login can be done by entering correct userid and password combination

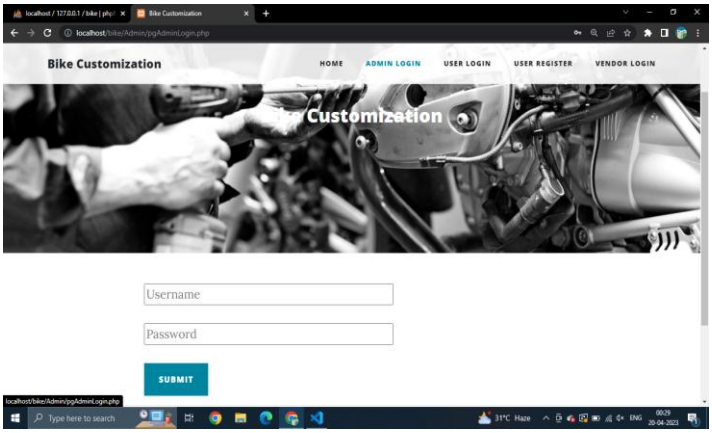

Fig. Admin Login

E commerce section can different categories in case a user wants to be specific with their purchase and not buy according to the smart cart.

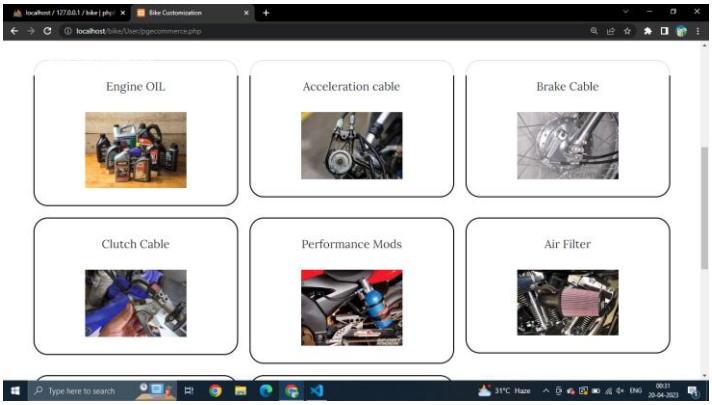

Fig. Ecommerce page

After completing the purchase users are redirected to the cart page where they can review and edit their orders.

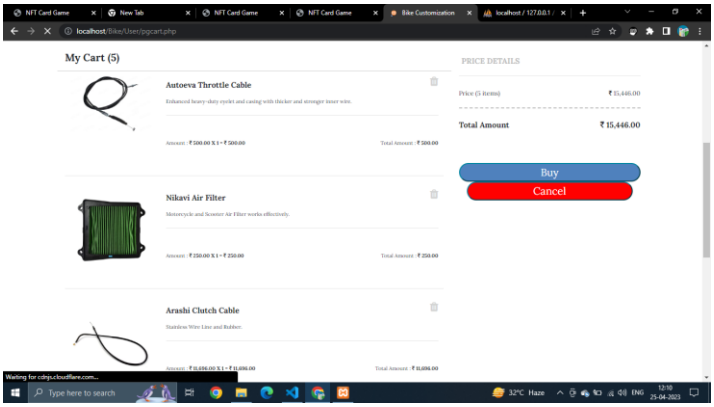

Fig. Cart

Then the buy now click redirects the users to the razorpay transaction gateway where the payment can be made by any method accepted by the gateway or simply click cash on delivery option.

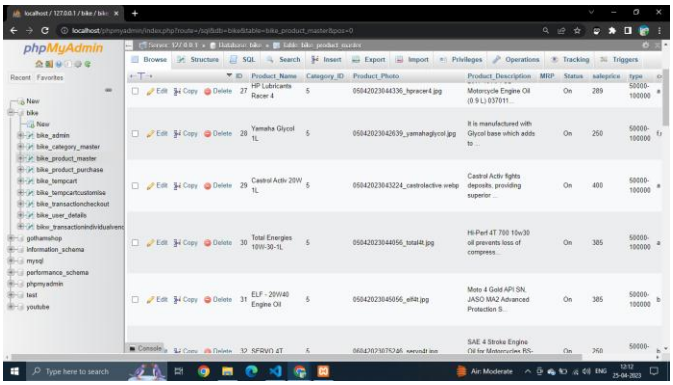

Fig. Database

All of the customer data is stored/retrieved from the sql database accessible from the phpmyadmin page. There are different tables with multiple attributes.

10. Future Scope

Adding more vendors is key to giving users a better choice among them. We also wish to add a feedback form for grievances related to the vendors and a rating system based on 5 star grading after completion of each order. The number of orders completed by each vendor will also be displayed and discount coupon application will be added. The product list will be expanded to more products and categories so users will have an even larger pool of products to choose from. Customer support in the form of a chat bot for simple queries and human interaction in case of larger queries will be added. The training of the data will be done via fp-tree algorithm to manage with the scaling of the project.

11. CONCLUSIONS

The project was successfully implemented and all the initial requirements were fulfilled. The smart cart recommendations are retrained every time there is a significant increase in the number of transactions and the recommendations are helpful. The website is responsive and fast, users can use it through their phone too. The order tracking is accurate and there are no active bugs.

12. REFERENCES

[1] T. Deekshitha, M. Venkatesan, and S. Vijayalakshmi, "Design and Implementation of a Web Portal for Purchasing Bike Parts with Smart Cart," International Journal of Engineering and Advanced Technology, vol. 10, no. 4, pp. 1653-1659, 2021.

[2] S. V. S. S. Satyanarayana, P. S. Reddy, and P. V. S. S. S. G. Gupta, "Smart Shopping Cart System for Efficient Shopping," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6S, pp. 1049-1052, 2019.

[3] M. V. Kumar, R. Vinodh Kumar, and B. Anjaneyulu, "Development of a Web-based E-commerce System for Bike Parts," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 9, no. 3, pp. 443-447, 2019.

[4] S. Kumar and A. Kumar, "Design and Development of Online Bike Parts Purchase Portal with Smart Cart," International Journal of Computer Sciences and Engineering, vol. 9, no. 6, pp. 383-389, 2021.

[5] H. Garg and N. Gupta, "A Comprehensive Study on E-commerce Websites and Customer Satisfaction," Journal of Applied Engineering Research, vol. 13, no. 4, pp. 328-333, 2018.

[6] R. Kaur and S. Dhiman, "A Study of User Perception and Expectation towards Online Shopping," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 6, no. 1, pp. 24-28, 2021.

[7] N. N. Al-Madi and M. A. Almulla, "Factors Affecting Online Shopping Behavior: A Study of Kuwaiti Consumers," Journal of Internet Banking and Commerce, vol. 25, no. 2, pp. 1-19, 2020.

[8] Priority-Based Scheduling Algorithm for Real-Time Systems" by R. S. Panda and R. K. Sahoo, published in the International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No. 4, April 2008.

[9] Liu, B., Hsu, W., & Ma, Y. (1999). Integrating classification and association rule mining. In Proceedings of the Fourth

International Conference on Knowledge Discovery and Data Mining (KDD-99) (pp. 80-86).

[10] Savasere, A., Omiecinski, E., & Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. In Proceedings of the 21st International Conference on Very Large Data Bases (VLDB) (pp. 432-444).

[11] Park, J. S., Chen, M. S., & Yu, P. S. (1995). An effective hash-based algorithm for mining association rules. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (pp. 175-186).

[12] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 1-12).

[13] Zhang, H., & Ramakrishnan, R. (2002). Optimal grid layout for data mining: a case study of the Apriori algorithm. In Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (pp. 567-578).

[14] Priority-Based Scheduling Algorithm for Real-Time Systems" by R. S. Panda and R. K. Sahoo, published in the International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No. 4, April 2008.

[15] A Dynamic Priority Queue Algorithm for Real-Time Embedded Systems" by H. Abeni, M. Caccamo, and L. Palopoli, published in the Proceedings of the IEEE Real-Time Systems Symposium, 2003.

[16] Priority-Based Scheduling in Real-Time Systems: A Survey" by C. C. Hsu and W. K. Shih, published in the Journal of Systems and Software, Vol. 81, No. 9, September 2008.

[17] A Priority-Based Algorithm for Resource Allocation in Cloud Computing" by H. Wang, Y. Zhang, and X. Li, published in the Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems.

[18] An Improved Priority-Based Scheduling Algorithm for Real-Time Systems" by M. H. Jafri and M. S. Bhatia, published in the Proceedings of the 2013 IEEE International Conference on Advanced Communication Technology (ICACT).