(IJACSA) International Journal of Advanced Computer Science and Applications,
Vol. 15, No. 8, 2024

The Application of Anti-Collision Algorithms in University Records Management

Ying Wang[1]*, Ying Mi[2]
Archives of the Party and Administration Office, North China Electric Power University, Baoding, 071000, China[1]
Department of Economic Management, Baoding Vocational and Technical College, Baoding, 071000, China[2]

***Abstract*—University records management has grown in importance as a result of the quick growth of big data, artificial intelligence, and other technologies. However, university archives management is prone to data loss, redundancy, and errors. Moreover, the use of scientific management systems and algorithms can effectively improve such problems. To create an effective and secure archive management system and run simulation tests, the study suggests an RFID-based archive management system and uses nested random time slot ALOHA (RS0) and binary tree (BT) anti-collision algorithms to solve the collision problem between tags in the created system. The test results showed that the average query coefficient, recognition efficiency, and communication volume of the proposed algorithm were 1 and 1.2 times, 95% and 90%, 50 Bit and 180 Bit in two scenarios, continuous and uniform, respectively. 0.91% and 3.92%, 24.21% and 31.14% of the system CUP and memory occupation were achieved when the number of clients was 10 and 100, respectively. The average response time of the system was 0.112s and 1.244s when 100 and 1000 users were accessed, respectively. The information extraction accuracy of the system was 94% at 1000 accessed users. This suggests that the approach used in the study can significantly improve the operational effectiveness of the records management system and the accuracy of information extraction, as well as provide technical support for improving the university records management system.**

Keywords—Anti-collision algorithms; archive management systems; information networks; RFID technology

I. INTRODUCTION

The number of archives has greatly increased as the twenty-first century has progressed due to the growth in both students and faculty in higher education. As a result, many schools have established electronic information-based records management systems [1-2]. Additionally, radio frequency identification technology can be extensively employed in the gathering and extraction of archival information, among other things, due to its benefits of high recognition efficiency and long usage time [3]. However, one of the key issues with radio frequency recognition technology is the collision problem. In this technology, the collision problem is further broken down into collisions between readers and collisions between tags. In most cases, it is the collision between tags that is encountered. While common RFID tag anti-collision algorithms are effective in improving the system's automatic identification performance, they also have several shortcomings. In high collision situations, each time slot or query will be subject to a significant amount of duplication, which can lead to a reduction in system efficiency or lag. Additionally, these algorithms are not well-suited for big data applications. When the data volume is very high and the number of tags exceeds the system's processing capacity, the system may experience performance issues and substantial delays [4-6]. The standard anti-collision technique is therefore no longer able to satisfy the requirements of archive management in the big data era. Therefore, to maintain the security, integrity, efficiency, and accuracy of archive management, it is important to upgrade the conventional anti-collision algorithm. Based on this background, it is proposed that when a tag collision occurs, a little time slot is first released to handle the tags that have collided. Depending on the number of tags that have collided, the nested random time slot ALOHA (SR) algorithm and the nested choice binary tree (BT) anti-collision algorithm are used to process them in anticipation of further improving the recognition efficiency and accuracy of the system.

II. OVERVIEW

Data management of archives is one of the key issues in the construction of universities today. The huge number and variety of types of archives in universities require a standardized management system to ensure the security, integrity and accuracy of archives management as well as the efficiency and accuracy of archives access. Numerous specialists have studied data management extensively in this area. Uka K. K. et al. created an online system for document sharing and data management at higher education institutions. The programme was a decentralized cloud-based file management and sharing programme. The results showed that the proposed system was able to enable file sharing, replication and data management permission transactions between accessing users in higher education institutions [7]. To make viewing of platform picture data more safe, Gamido et al. created an image file management system. The system was encrypted and kept on the server, and image files were encrypted using the AES technique to give the file owner more security. The outcomes demonstrated that the encryption of image files on the server was successful, and the technique for exchanging images was now reliable and efficient [8]. Veena et al. constructed a hybrid image and document archiving system in order to increase the volume of document storage and make document retrieval more efficient. The system used a client-server architecture, which utilized Opencv to identify objects in images and Tesseract to identify text in images and generate labels. The results showed that the system could be used to store and retrieve large numbers of documents [9]. Zhou et al. proposed a metadata service approach to the problem of dealing with large amounts of data requiring long, continuous and uninterrupted data access. The strategy made use of a novel shared storage pool and main-standby fault-tolerant design. The outcomes demonstrated that the method significantly decreased the average recovery time of the data service and enhanced the availability of the file system [10]. In the context of currently utilized distributed file systems that differentiate between the file system and network layers, Zhu et al. proposed a distributed persistent memory file system with RDMA support. The system introduced self-identifying remote procedure calls and offered direct access to a shared pool of persistent memory. The results demonstrated that the file system outperformed other distributed file systems by several orders of magnitude [11].

The anti-collision algorithm, which is extensively employed in many sectors, can significantly enhance the system's automatic recognition performance. Choi et al. proposed an anti-collision algorithm for mobile robots to address the problem that robots were prone to collision during the distribution process. The algorithm will reset a new path and motion when the robot collides. The outcomes demonstrated that the system could successfully prevent a robot-robot collision [12]. The multi-tag collision problem in RFID systems was addressed by the hybrid ALOHA and tree algorithm (HAMT) proposed by Zhou et al. For tag recognition, the method employed the DFSA algorithm. The outcomes demonstrated that the algorithm's recognition efficiency could increase to about 0.72 [13]. For the collision problem of RFID tag detection in the Internet of Things, Qiu et al. suggested an improved group adaptive query tree technique (IGAQT). The approach fixed the issues with the original algorithm's temporal complexity and communication overload. The outcomes demonstrated that the proposed strategy had a significantly higher recognition efficiency [14]. For the RFI multi-tag collision avoidance problem, Qu J et al. offered an adaptive frame-time slot ALOHA collision avoidance algorithm based on IGA. The algorithm counted the amount of tags before grouping them to identify tags. According to the findings, the algorithm was approximately 71% efficient, which is 90% more effective than the conventional ALOHA algorithm [15]. To address the collision issue, Jing C et al. devised an anti-collision method based on blind source separation (BSS). FastICA, PowerICA, ICA_p, and SNR_MAX were among the BSS algorithms that were used to separate and test the mixed signals in RFID systems. The ICA_p method had the best overall performance among the aforementioned algorithms, according to the results [16].

From the research of numerous scholars mentioned above, anti-collision algorithms have been widely applied in multiple fields, most of which are applied in the Internet of Things, robot path research, and other fields. Few scholars have combined anti-collision algorithms with data archive management systems. Therefore, this research innovatively introduces anti-collision algorithms, a widely used technology in fields such as the Internet of Things, into university archive management systems, thus filling the technical gap in RFID tag recognition in this field. At the same time, the existing anti-collision algorithms are improved and optimized, and a combination scheme of nested SR algorithm and BT anti-collision algorithm is proposed. This scheme has shown significant advantages in reducing tag collisions, improving recognition efficiency and accuracy, and providing new ideas and methods for the application of RFID technology in the field of records management.

III. DESIGN OF AN RFID-ANTI-COLLISION ALGORITHM BASED ARCHIVE MANAGEMENT SYSTEM FOR UNIVERSITIES

This chapter first elaborates on the functions and design scheme of a university archives management system based on RFID technology. In response to the issues of data loss, redundancy, and errors that are inherent to university archives management, a nested SR algorithm and BT anti-collision algorithm have been developed to enhance the efficiency and accuracy of tag collision recognition, mitigate the risk of collisions between tags, and ultimately improve the operational efficacy of the university archives management system.

A. *Model Design of RFID Technology-based University Archive Management System*

In colleges and universities, there are many distinct kinds of records management, and each college has various records management requirements. The basic functions required for a general university records management system are shown in Fig. 1.

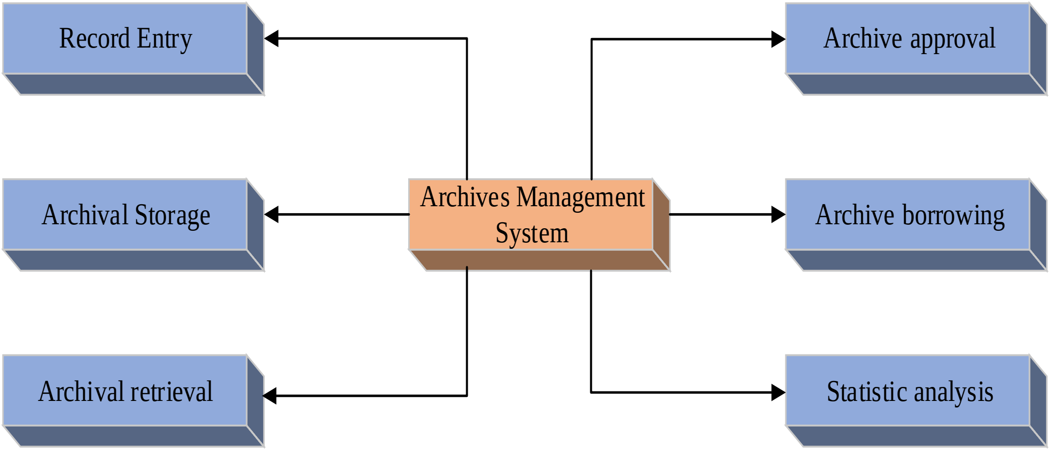


Fig. 1. Basic functions of university archives management system.

The basic operations of the university's records management system, including record entry, record storage, record retrieval, record approval, record borrowing, and statistical analysis, are shown in Fig. 1. According to the different needs of major universities for records management, the corresponding functions can be added or deleted. According to the needs of general universities for archive management systems, a network connection structure of university archive management systems based on RFID technology is studied and designed, as shown in Fig. 2.

As shown in Fig. 2, the network structure of the system uses a reader to first read the file information collected by the antenna, a WEB server to send it to the database server for backup, and a terminal server to implement file information management for students and teachers [17]. Four key components make up the function of the file management system developed for the project, which are shown in Fig. 3.

As demonstrated in Fig. 3, the archive management system designed in the study mainly includes four modules: user management, archive management, information collection and arrangement, and database management. The user management module mainly focuses on the access rights of different users to the archives. The database module mainly backs up and destroys the archives. The information collection and arrangement module mainly collects and processes information from different sources of archives. The archive management mainly manages the information of archives such as access, borrowing and returning, and whether they are in place. The database module is used for backing up and destroying the archives. The information collection and arrangement module is used for collecting and processing information from different sources. The archives management module is used for managing the information on access, borrowing and returning, and whether the archives are in place [18]. Six major divisions and fourteen subsections have been created based on the needs of the system's users to better serve their needs. The specific structure of the system functions for the management of archives by users is shown in Fig. 3.

As shown in Fig. 4, the basic functions of archive management in the university archive management system include user rights management, organization management, classification management, archive management, and borrowing management. According to the different user rights, the system divides users into three categories: system administrator, file administrator, and ordinary user. The system administrator has the highest authority and can perform all the functions of the system and modify the user rights to modify the file information. The file administrator is responsible for the daily maintenance of the system. Ordinary users can only query their own file information. The study employs a three-tier architecture for the software system structure in order to meet the many functional needs of the users and to make the management system convenient and effective with lower running costs. Fig. 5 shows the exact technical design of this system.

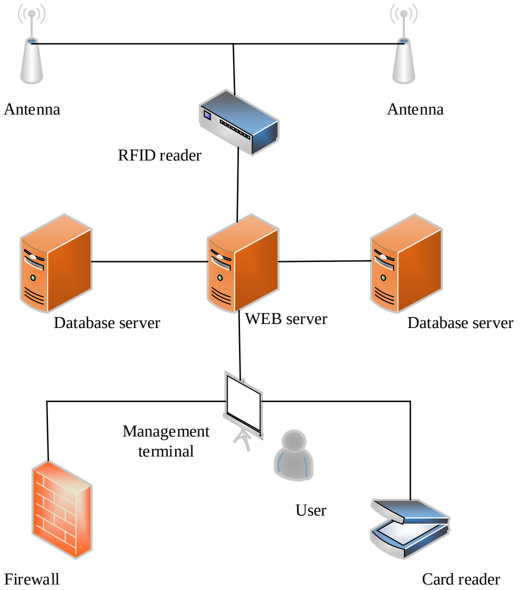


Fig. 2. System for managing university archives connections to the internet.

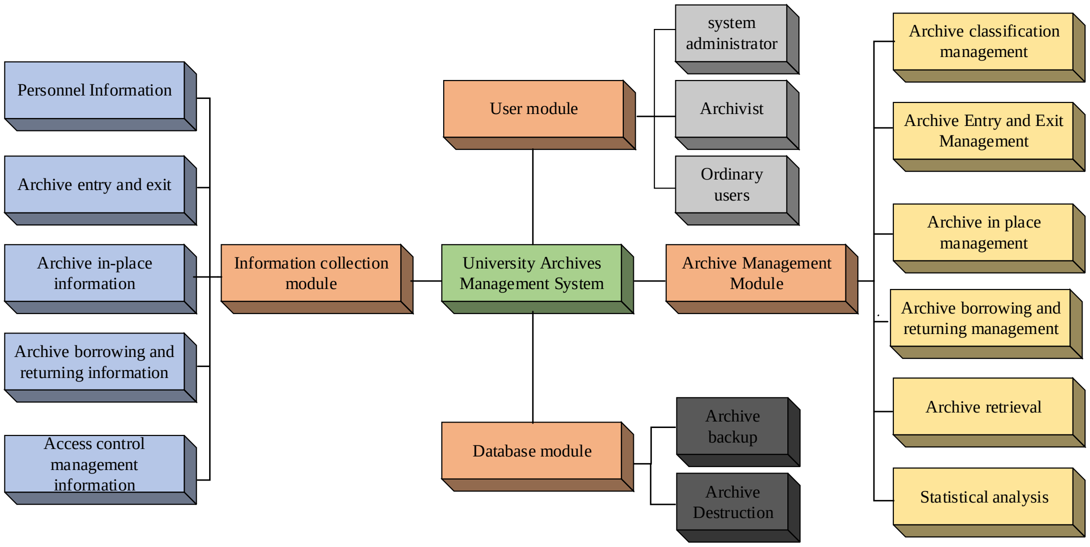


Fig. 3. System for managing university archives, showing its functional structure.

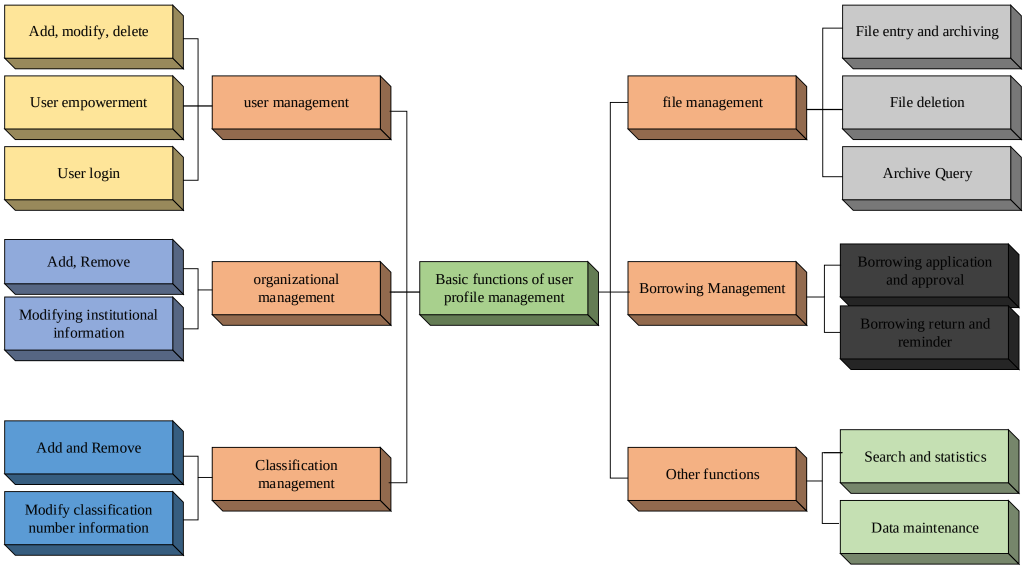


Fig. 4. Schematic representation of the fundamental file management operations carried out by users.

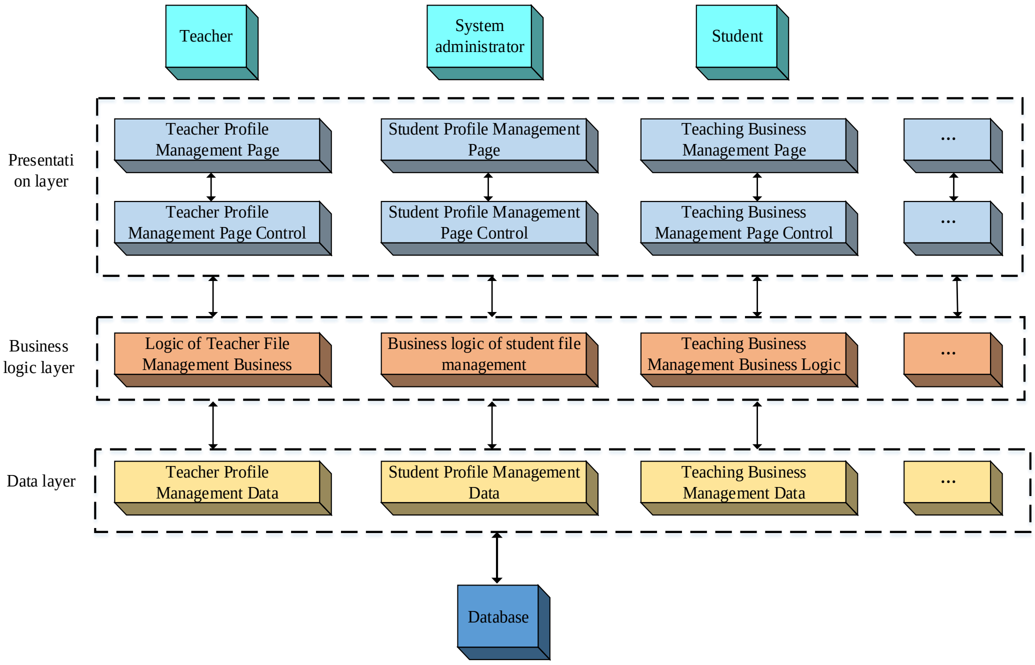


Fig. 5. Technical architecture of university archives management system.

Fig. 5 illustrates the three-layer B/S architecture used by the university file management system, with the presentation layer serving as the user operation interface via which users can manage, query, and alter files as needed. The business logic layer is the system operation logic processing, the main function of which is to parse the information sent by the user operation interface and execute the parsed instructions, and then send the instructions to the data layer. The data layer serves as the system's data processing layer. Its primary function is to filter out the necessary data in accordance with the instructions from the business logic layer, after which it sends the data back to the business logic layer for processing before sending it to the presentation layer for the users to see. Then, according to the needs of this file management system, the corresponding running tools and development environment are selected, as shown in Table I.

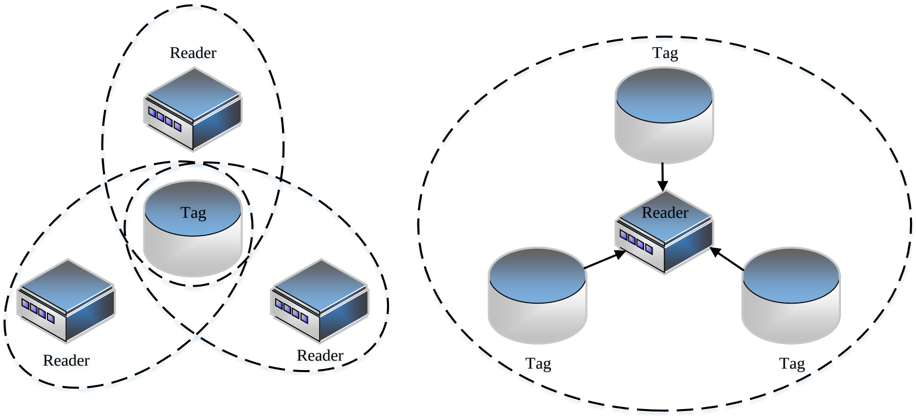
TABLE I. OPERATING CONDITIONS AND TOOLS OF ARCHIVE MANAGEMENT SYSTEM

Operating system	WIN 8
Database	MYSQL
B/S architecture development tool	Eclipse

B. Design of Improved Anti-Collision Algorithms Based on RS and BT

The university's RFID-based file management system has a sizable number of tags. When data is transmitted between these tags and the reader, collision of data information is likely to occur, resulting in failure of information transmission [19]. There are two situations in which data information transmission collision occurs, as shown in Fig. 6.

Fig. 6(a) shows a tag-to-reader collision, which is caused by a tag being unable to accept command requests from multiple readers at the same time. A tag-to-tag collision is depicted in Fig. 6(b), where a single reader is unable to extract the data from several tags at once. While the former collision scenario can be avoided by adjusting transmit power, the latter scenario requires strengthening the anti-collision algorithm. The ALOHA-based algorithm and the tree-based algorithm are the two anti-collision algorithms between tags that are most frequently utilized. ALOHA algorithms include the pure ALOHA algorithm (PA algorithm), the time slot ALOHA algorithm (SA algorithm), and the frame time slot ALOHA algorithm (FSA algorithm) [20]. The FSA algorithm decomposes the recognition process into separate recognition frames. Since the FSA algorithm can break down the recognition process into individual frames and utilize the time slot ALOHA algorithm one at a time, this method has greater recognition accuracy. The SR algorithm is devised since the amount of time slots per frame cannot be altered. This algorithm has the ability to change the total number of time slots in a recognition cycle and allow unrecognized tags to select a different time slot. The algorithm flow is shown in Fig. 7.



(a). Reader - Tag collision

(b). Tag - Tag collision

Fig. 6. Collision situation.

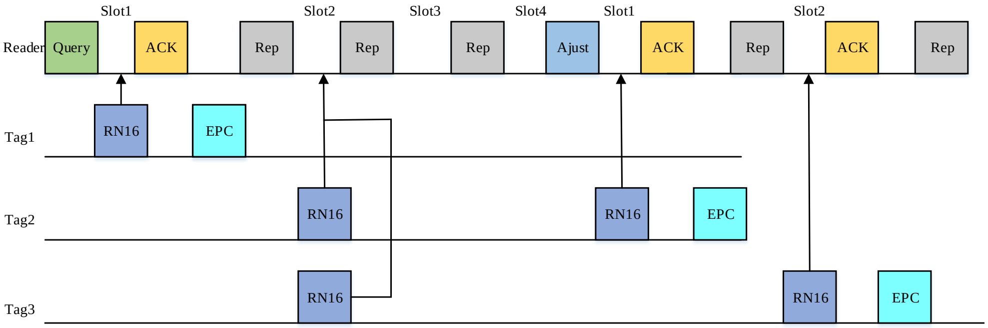


Fig. 7. SR algorithm flowchart.

The SR algorithm's main flow is depicted in Fig. 7. The reader transmits a command to the tag with the parameter Q, which the tag receives and uses to create a random number as the response command when the time period is up. Finally, the reader sends a separate instruction and moves on to the following time slot based on the corresponding amount of tags. The success rate of a specific tag identification in a particular time slot of the cycle is displayed in Eq. (1).

$$Pe = C_n^1 \frac{1}{2^Q}\left(1-\frac{1}{2^Q}\right)^{(n-1)} \tag{1}$$

In Eq. (1), Q is the parameter in the command. n is the total number of tags. When the reader accepts signals from several tags at the same time, the probability of collision between tags is shown in Eq. (2).

$$P_i = C_n^i \left(\frac{1}{2^Q}\right)^i \left(1-\frac{1}{2^Q}\right)^{(n-i)} \left(i \geq 2\right) \tag{2}$$

Let the total number of time slots be 2^Q , then the total expected value of tag collisions over the entire recognition cycle is calculated as shown in Eq. (3).

$$E_0 = n\left(1-\frac{1}{2^Q}\right)^{(n-1)} \tag{3}$$

In Eq. (3), E_0 is the total expected value of collisions for a given cycle of tags, based on which the algorithm efficiency of SR can be calculated in Eq. (4)

$$\eta_{SR}=\frac{n\left(1-\frac{1}{2^Q}\right)^{(n-1)}}{2^Q} \tag{4}$$

In Eq. (4), η_{SR} is the efficiency of SR algorithm. The tree structure-based collision prevention algorithms include the BT algorithm, the query algorithm (QT algorithm), and the spanning tree algorithm. Fig. 8 displays the schematic diagram of the BT anti-collision algorithm among them.

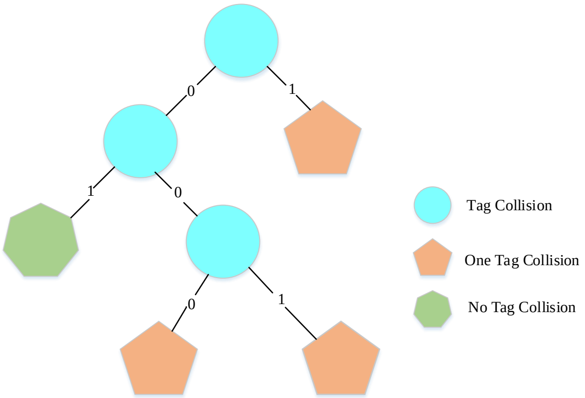


Fig. 8. Schematic diagram of binary tree anti-collision.

As shown in Fig. 8, the algorithm starts with a counter for each reader and tag, each with a value of 0. The tag then sends a message to the reader, and the reader adjusts the value of the counter according to the message to drive the algorithm. Finally, the algorithm terminates when the reader's counter value reaches 0. The formula for the h level of the binary tree, which has 2^k cycles at the h level, is given in Eq. (5).

$$\begin{cases} S_{BT}(n,h)=2^h\left(1-\frac{1}{2^h}\right)^n \\ A_{BT}(n,h)=n\left(1-\frac{1}{2^h}\right)^{(n-1)} \\ H_{BT}(n,h)=\sum_{h=0}^{\infty}2^h\left[1-\left(1-\frac{1}{2^h}\right)^{(n-1)}-n\frac{1}{2^h}\left(1-\frac{1}{2^h}\right)^{(n-1)}\right] \end{cases} \tag{5}$$

In Eq. (5), $S_{BT}(n,h)$ is the number of idle cycles. $A_{BT}(n,h)$ is the number of readable cycles. $H_{BT}(n,h)$ is the number of collision cycles for the h layer. In this algorithm, the number of queries $L(n)$ and the amount of communication $S(n)$ for the reader to recognize the tag are shown in Eq. (6).

$$\begin{cases} L(n)=n(\log_2 n+1) \\ S(n)=Y\square n(\log_2 n+1) \end{cases} \tag{6}$$

Based on the basic properties of bifurcation numbers, the total period identified by the algorithm can be obtained as $1+H_{BT}$. Both of these algorithms are common anti-collision algorithms for handling tag collisions, but both have shortcomings, so the study proposes a sub-nesting algorithm to improve them. In the event of a collision between tags when a command is sent in the SR algorithm, the reader processes the tag directly into the subsequent time slot, thereby bypassing the tag that collided. To address this problem, the study proposes that instead of going to the next gap immediately after a collision occurs, the colliding tag is processed first and the SR algorithm is nested with SR. Let the number of time slots to be released by this algorithm be m, and the SR nested SR algorithm is analyzed by the formula, the probability of success of identifying a tag per time slot in this algorithm is shown in Eq. (7).

$$Pe_1=\left(1-\frac{1}{m}\right)^{(m-1)} \tag{7}$$

Since there is a m time slot, the expectation of successful tag recognition at m is expressed in Eq. (8).

$$E_1=m\left(1-\frac{1}{m}\right)^{(m-1)} \tag{8}$$

In contrast, the traditional SR algorithm's expectation of successful tag recognition in m time slots is shown in Eq. (9).

$$\frac{n}{2}\left(1-\frac{1}{2^Q}\right)^{(n-1)} \tag{9}$$

Therefore, the nested SR algorithm label recognition success expectation optimization rate can be obtained as shown in Eq. (10).

$$\eta=\frac{\frac{n}{2}\left(1-\frac{1}{2^Q}\right)^{(n-1)}}{m\left(1-\frac{1}{m}\right)^{(m-1)}}\times 100\% \tag{10}$$

Moreover, for the improvement of the BT algorithm, the study uses the SR nested BT algorithm to improve it. Let the number of tags to be recognized be L ,the depth of the binary tree is $\left\lceil \log_2^L \right\rceil$, combined with the traditional BT algorithm, Eq. (11) can be used to determine the time required to recognize the tags and the formula for calculating expectations.

$$\begin{cases} E_A=\sum_{a=1}^{\left\lceil \log_2^L \right\rceil}\left(1-\frac{1}{2^a}\right)^{L-1} \\ T_A=\sum_{a=1}^{\left\lceil \log_2^L \right\rceil}2^a \end{cases} \tag{11}$$

In Eq. (11), E_A is the expectation of successful tag recognition. T_A is the time taken to recognize the tag. The efficiency of the algorithm can be obtained from the expectation and the time taken. Next, the SR nested SR algorithm and the SR nested BT algorithm are compared. Let the total number of tags identified by the SR nested BT algorithm be E_A and the time taken be T_A. For the same usage time, the total number of tags identified by the SR nested SR algorithm is E_{SR}, and the tag identification expectation is given in Eq. (12).

$$E_{SR} = T_A\left(1-\frac{1}{T_A}\right)^{(T_A-1)} \tag{12}$$

The label recognition expectations according to the two algorithms are compared in size using the do-quotient method, as shown in Eq. (13).

$$\lambda = \frac{T_A\left(1-\frac{1}{T_A}\right)^{(T_A-1)}}{\sum_{a=1}^{\lceil \log_2^L \rceil}\left(1-\frac{1}{2^a}\right)^{L-1}} \tag{13}$$

The time used in Eq. (13) can be brought in with the time used in Eq. (10) and calculated to give Eq. (14).

$$\lambda = \frac{\sum_{a=1}^{\lceil \log_2^L \rceil}\left[2^a\left(1-\frac{1}{\sum_{a=1}^{\lceil \log_2^L \rceil}2^a}\right)^{\left(\sum_{a=1}^{\lceil \log_2^L \rceil}2^a\right)-1}\right]}{\sum_{a=1}^{\lceil \log_2^L \rceil}\left(1-\frac{1}{2^a}\right)^{L-1}} \tag{14}$$

According to Eq. (14), it is concluded that the SR nested SR algorithm has more tag recognition when $L \geq 5$ and the SR nested BT algorithm has more tag recognition when $L < 5$ [21]. Based on this conclusion, the nested RS and BT anti-collision algorithm flow is shown in Fig. 9.

The upgraded algorithm's reader delivers a command with a Q value, as seen in Fig. 9, which kick-starts the cycle of tag recognition. Each tag receiving the command randomly generates its own tag number and selects a corresponding time slot. When there is no tag response, the reader issues a command to move on to the next time slot. When there is one tag response, the reader issues a command to read and write the tag before moving on to the next time slot. Moreover, when there are multiple tag responses, the reader issues a nested BT algorithm command if there are $L < 5$ or $L \geq 5$ tag collisions, respectively.

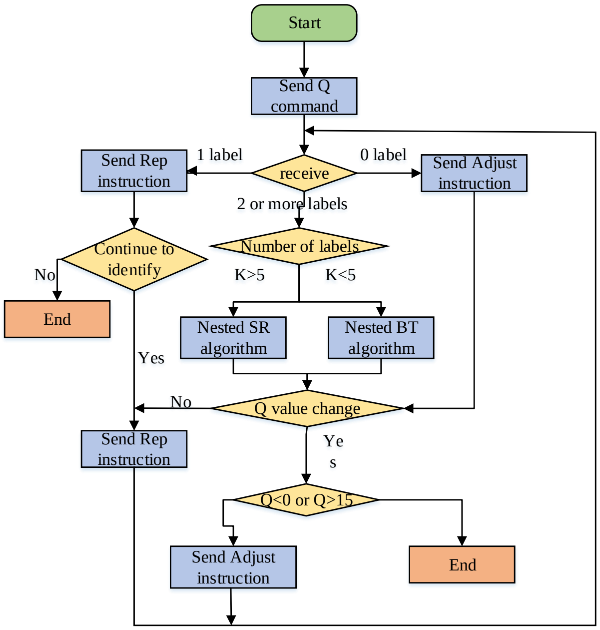


Fig. 9. Flow diagram of nested RS and BT anti-collision algorithm.

IV. Performance Analysis and Simulation Application of RFID-Based Anti-Collision Algorithm for Archive Management System

This chapter mainly elaborates on the performance of nested RS and BT anti-collision algorithms and the design of file management system detection experiments.

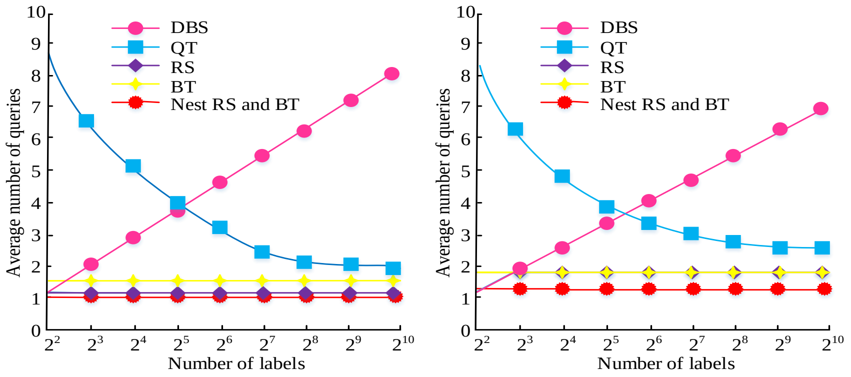
A. *Performance Analysis of Nested RS and BT Anti-Collision Algorithms*

The study divides the tag identity codes into two distribution scenarios—continuous and uniform—to test the effectiveness of the recommendation algorithms developed in this study. The performance of the five QT algorithms, DBS algorithm, RS algorithm, BT algorithm, nested RS and BT algorithms are simulated and tested in these two scenarios respectively. Table II displays the experimental parameter settings.

TABLE II. Experimental Parameter Settings

Number of readers	1
Label length	100
Transfer rate	100kbps
Number of labels	4~2048
Software	Matlab
Number of experiments	100

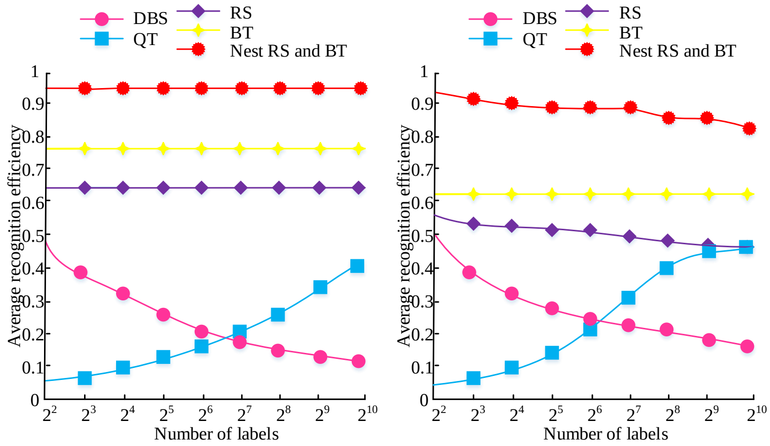
The average number of queries for each algorithm is first tested in two distribution scenarios for the tag identity code. The results of which are shown in Fig. 10.



(a)Continuous distribution of label UID (b)Uniform distribution of label UID

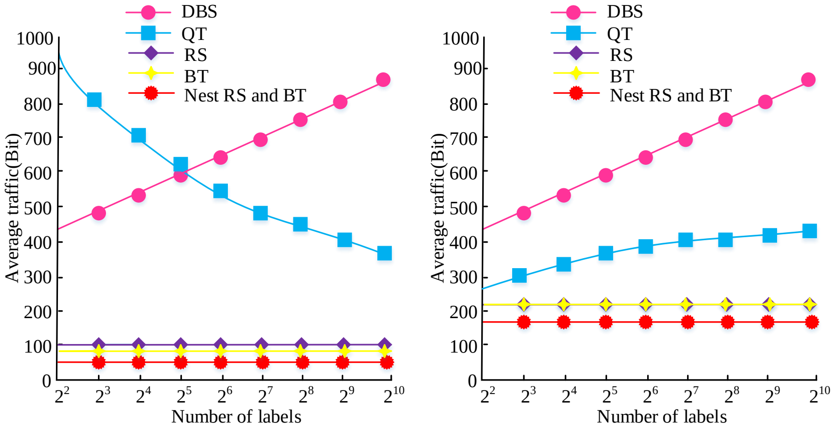
Fig. 10. Average query times of different algorithms in two scenarios.

As shown in Fig. 10(a), the average number of queries for the five algorithms in the continuous scenario is shown. It can be concluded that the average query counts of the three algorithms, nested RS and BT algorithms, RS algorithm and BT algorithm, remain stable. Therefore, the average number of times to identify a tag is 1, 1.2 and 1.5 times, respectively. The average query count of DBS algorithm increases with the number of tags. The average query count of QT algorithm decreases with the number of tags. The average number of inquiries for the five methods in the uniform scenario are shown in Fig. 10(b). It can be concluded that the DBS algorithm and QT algorithm query counts vary similarly to (a). The RS algorithm and BT algorithm both have the same average query count of 1.7. The nested RS and BT algorithms have the lowest query count of 1.2. The average recognition efficiency of the various algorithms is then tested in two distribution scenarios of tag identity codes, and the specific results are shown in Fig. 11.



(a)Continuous distribution of label UID (b)Uniform distribution of label UID

Fig. 11. Average recognition efficiency of different algorithms in two scenarios.



(a)Continuous distribution of label UID (a)Continuous distribution of label UID

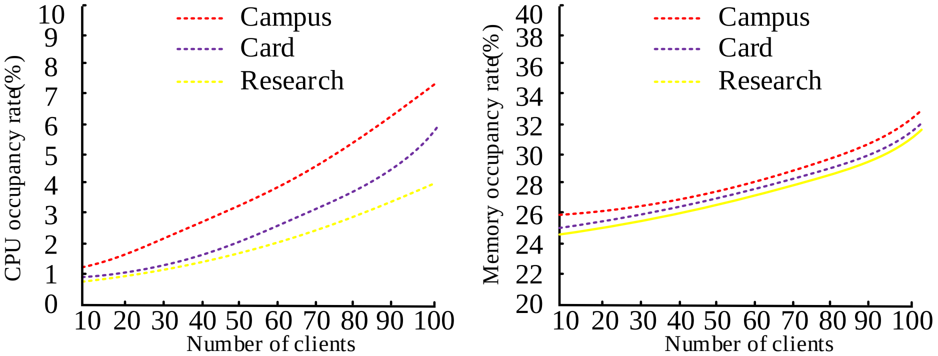
Fig. 12. Average traffic of different algorithms in two scenarios.

As shown in Fig. 11(a), the average recognition efficiency of the five algorithms in the continuous scenario is shown. As can be shown, the average recognition efficiency of the three algorithms—nested RS and BT algorithms, RS algorithm, and BT algorithm—remains stable as the number of tags increases, at 95%, 64%, and 78%, respectively. The average recognition efficiency of the DBS algorithm and QT algorithm is constantly changing, with the highest being around 50%. Fig. 11(a) shows the average recognition efficiency of the five algorithms in the uniform scenario. As can be observed, the average recognition efficiency of the BT algorithm stays constant at 62%, whereas that of the nested RS and BT + RS algorithms varies between 90% and 55%, respectively. The average recognition efficiency of the DBS and QT algorithms is similar to the case in (a), with a maximum of 50%. Finally, the average communication of the various algorithms is tested in two distribution scenarios of tag identity codes, and the specific results are shown in Fig. 12.

As shown in Fig. 12(a), the average communication volume of the five algorithms in the continuous scenario is shown. It can be concluded that the average communication volume of three algorithms, namely nested RS and BT algorithm, RS algorithm, and BT algorithm, remains stable, with 50, 100, and 80 Bit, respectively. While the average communication volume of the QT algorithm is inversely proportional to the number of tags and has a minimum communication volume of 400 Bit that of the DBS method is proportionate to the number of tags and has a minimum communication volume of 430 Bit. Fig. 12(b) shows the average communication volume of the five algorithms in the continuous scenario. The average communication volume of the five algorithms is shown in Fig. 12(b). The average communication volume of the nested RS and BT algorithms, RS algorithm and BT algorithm remains stable but increases slightly to 180, 220 and 220 Bit respectively. The average communication volume of the DBS algorithm is the same as in scenario (a). The average communication volume of the QT algorithm increases slowly to a minimum of 280 Bit and uniform distribution scenarios.

B. Simulation Application Analysis of a File Management System Based on Nested RS and BT Anti-Collision Algorithms

To further verify the system designed by the research in practical applications, a university is selected to conduct simulation experiments with the research system. To provide a point of comparison, a traditional campus network file management system and a card-based file management system are also set up. The number of clients is set to 100, and the CPU occupancy and memory occupancy of the three systems are first tested. The results are shown in Fig. 13.



(a)CUP occupancy rate (b)Memory occupancy

Fig. 13. CPU usage and memory usage of three systems.

As shown, Fig. 13(a) shows the change of CP U occupancy with client growth for the three systems. Fig. 13(b) shows the change of memory occupancy with client growth for the three systems. It can be concluded that the CPU occupancy and memory occupancy of the study system are smaller than those of the other two systems throughout the client growth. When the number of clients is 10, the CPU occupancy of the research system is 0.91% and the memory occupancy is 24.21%, while the CPU occupancy and memory occupancy of the other systems are as low as 0.97% and 25.13%. When the number of clients is 100, the CPU occupancy of the research system is 3.92% and the memory occupancy is 31.14%, while the CPU occupancy and memory occupancy of the other systems are as low as 5.98% and 32.13%. The average response time of the three systems is then evaluated once more, with the lowest values being 5.98% and 32.12%. Fig. 14 displays the average findings from 100 tests on the three systems' average reaction times with 100, 500, 1000, and 2000 users.

Fig. 14 shows that when the number of users increases, the average response time of the three algorithms increases as well. The average response time of the three systems is roughly the same at 0.112 seconds when there are 100 users being accessed? The average response time of the studied system is 1.244 seconds, which is 0.723 seconds faster than the card-based archive management system campus network archive management system and 1.036 seconds faster than the campus network archive management system. However, as the total amount of users increased, the difference in average response time between the three systems grow increasingly larger. When the number of access users reaches 2000, the studied system's average response time is 1.244 seconds. Finally, the accuracy of archive information extraction is tested for the three systems. A total of 1,000 users are permitted to access the system at any given time, with the system operating continuously for a period of either 12 or 24 hours per day. The results are shown in Fig. 15.

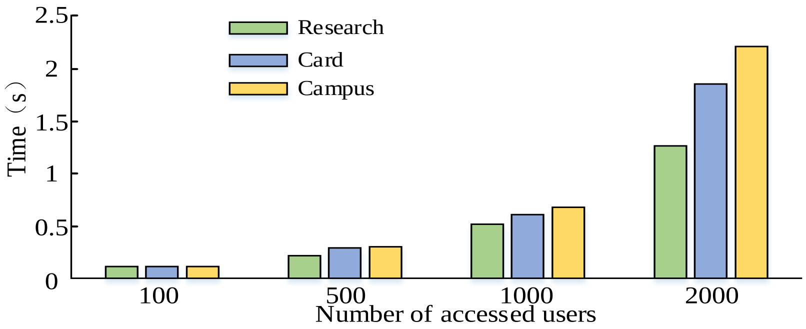
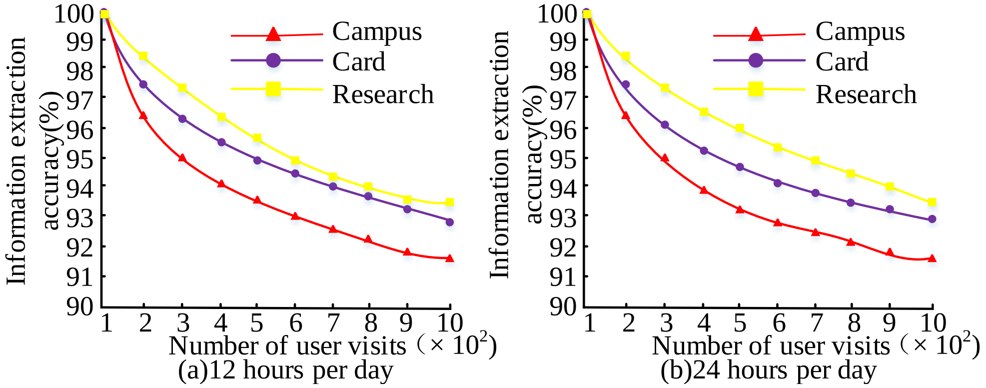


Fig. 14. Average response time of three systems.



(a)12 hours per day

(b)24 hours per day

Fig. 15. Accuracy of extracting archive information for three systems.

Fig. 15(a) depicts the correlation between the number of users visiting the archives and the accuracy of information extraction from the archives for the three systems operating for 12 hours each day. It is clear that for all three systems, information extraction accuracy is high when the number of access users is low. The accuracy rate starts to decline as the user base grows, but the research system ages the least quickly. When the number of users is 1000, the accuracy of the research system algorithm is 94%, while the other two algorithms are accurate up to 93%. Fig. 15(b) shows the accuracy of archive information extraction versus the number of users accessed for the three systems running for 24h a day. The comparison shows that there is only a slight difference between the two scenarios. The duration of operation of all systems does not affect this result. In conclusion, it is clear that the research system offers quick performance, excellent information correctness, and minimal memory utilization for managing university records.

V. DISCUSSION

Research on the effective integration of RFID technology, Internet big data and artificial intelligence technology is an important trend in the field of Internet of Things. Internet big data provides a rich database for RFID system, while AI technology further improves the intelligence level of RFID system through data analysis, prediction and optimization. This technological integration can not only improve the efficiency of university archives management, but also improve the system's adaptability and decision-making ability. In the process of RFID tag recognition, nested random slot ALOHA and binary tree anti-collision algorithms were introduced and optimized, significantly improving the efficiency and accuracy of tag recognition. The combination of these two algorithms not only leveraged the advantages of the random slot ALOHA algorithm in reducing collision probability, but also utilized the characteristics of the binary tree algorithm in quickly locating and solving collision problems. This dual optimization of recognition efficiency and communication volume is achieved through the effective integration of these two algorithms. From the test results, this study outperformed other algorithms in terms of average query coefficient, recognition efficiency, and communication volume, which fully demonstrated the effectiveness and superiority of the proposed algorithm. Especially in continuous and uniform scenarios, its performance advantages were more pronounced, which was crucial for stability and reliability in practical applications. This study is of great significance for fields such as university archive management that require high-precision and high-efficiency record management.

VI. CONCLUSION

An archive management system based on RFID technology was researched, designed and implemented, effectively integrating big data and artificial intelligence technologies to improve the intelligence level of archive management. The study introduced nested SR and binary tree anti-collision algorithm into the archive management system to solve the collision problem in the RFID tag recognition process. The test results showed that in both continuous and uniform scenarios, the proposed algorithm outperformed other algorithms in terms of average query coefficient, recognition efficiency, and communication volume, with values of 1.2

times and 1.2 times, 95% and 90%, and 50Bit and 180Bit, respectively. In practical application testing, when the number of clients was 10 and 100, the CPU and memory occupancy of the research system were 0.91% and 3.92%, 24.21% and 31.14%, respectively. At different numbers of clients, the CPU and memory usage of the system remained at a low level, indicating high resource utilization and the ability to meet the needs of large-scale file management. When accessing 100 and 1000 users, the average response time of the research system was 0.112 seconds and 1.244 seconds, respectively. The average response time of the system was short. Especially in high concurrency access scenarios, it still maintained good response speed and improved user experience. When accessing 1000 users, the information extraction accuracy of the research system was 94%. This indicated that the method could reduce recognition practice, improve recognition efficiency, and have high recognition accuracy. However, this method has not fundamentally solved the label collision problem, and the optimization level of information extraction accuracy is not obvious, which needs further improvement. Future research should concentrate on the fundamental resolution of the RFID tag collision issue. This should be achieved through the implementation of innovative algorithms and technological solutions, which will significantly enhance the accuracy of information extraction. Furthermore, continuous optimization of system performance is essential to guarantee the efficiency, accuracy, and stability of the archive management system.

REFERENCES

[1] Watanabe O, Delmo C, Ridho T. Analysis Analysis of Mobile-Based Archive Management Information System Design. Journal of Information Systems and Technology Research, 2023, 2(2): 91-101.

[2] Reynaldo G. Alvez. “The Development of A Cloud-Based University Research Repository Software Using A Configurable Subscription Model.” Acta Informatica Malaysia. 2022; 6(1): 07-12.

[3] Alfath A N, Taningngsih I, Suharto E. “Designed an Inactive Archives Management Information System Using Visual Basic 2010.” The Department of Journal of Applied Engineering and Technological Science (JAETS), 2022, 3(2): 105-115.

[4] Ai Y, Bai T, Xu Y, Zhang, W. “Anti-collision algorithm based on slotted random regressive-style binary search tree in RFID technology.” IET Communications, 2022, 16(10): 1200-1208.

[5] Zan J. “Research on robot path perception and optimization technology based on whale optimization algorithm.” Journal of Computational and Cognitive Engineering, 2022, 1(4): 201-208.

[6] Fan Zhang. “Research On Keyword Mining of Academic Library Subject Service. Acta Informatica Malaysia.” 2023; 7(2): 97-100.

[7] Uka K K, Oguoma S I, Chuma-Uba U P. “Analysis of Blockchain Architecture in File Sharing Management for Tertiary Institution.” Intelligent Information Management, 2020, 12(3):88-104.

[8] Gamido H V, Gamido M V, Sison A M. “Developing a secured image file management system using modified AES.” Bulletin of Electrical Engineering and Informatics, 2019, 8(4):1461-1467.

[9] Veena B, Dhiraj S. “A Personalized and Scalable Machine Learning-Based File Management System.” Tehnički glasnik, 2022, 16(2): 288-292.

[10] Zhou J, Chen Y, Wang W, He S, Meng, D. “A highly reliable metadata service for large-scale distributed file systems.” IEEE Transactions on Parallel and Distributed Systems, 2019, 31(2): 374-392.

[11] Zhu B, Chen Y, Wang Q, Lu, Y., & Shu, Jl. “Octopus+: An rdma-enabled distributed persistent memory file system.” ACM Transactions on Storage (TOS), 2021, 17(3): 1-25.

[12] Choi Y I, Cho J H, Kim Y T. “Collision Avoidance Algorithm of Mobile Robots at Grid Map Intersection Point.” International Journal of Fuzzy Logic and Intelligent Systems, 2020, 20(2):96-104.

[13] Zhou W, Jiang N. “Research on hybrid of ALOHA and multi-fork tree Anti-collision algorithm for RFID.” Procedia Computer Science, 2021, 183(5):389-394.

[14] Qiu Y, Lu J, Yang L, Xu, Y. “Adapted RFID anti-collision algorithm and its application in sharing economy.” International Journal of Internet and Enterprise Management, 2020, 9(3): 248-260. Enterprise Management, 2020, 9(3): 248-260.

[15] Qu J, Wang T. “An adaptive frame slotted ALOHA anti-collision algorithm based on tag grouping.” Cognitive Computation and Systems, 2021, 3(1): 17-27.

[16] Jing C, Luo Z, Chen Y, et al. “Blind anti-collision methods for RFID system: a comparative analysis.” Infocommunications Journal, 2020, 12(3):8-16.

[17] Fahmi H, Fadli S, Ashari M, Ramadhon, M. S. Development of Mail Archive Management Information System at Lombok Tengah District Education Office. JISA (Jurnal Informatika dan Sains), 2022, 5(2): 165-172.

[18] Watanabe O, Delmo C, Ridho T. Analysis Analysis of Mobile-Based Archive Management Information System Design. Journal of Information Systems and Technology Research, 2023, 2(2): 91-101.

[19] Choudhuri S, Adeniye S, Sen A. “Distribution Alignment Using Complement Entropy Objective and Adaptive Consensus-Based Label Refinement for Partial Domain Adaptation.” Artificial Intelligence and Applications, 2023, 1(1): 43-51.

[20] Bertram J, Wei P, Zambreno J. A fast Markov decision process-based algorithm for collision avoidance in urban air mobility[J]. IEEE transactions on intelligent transportation systems, 2022, 23(9): 15420-15433.

[21] Shen K, Shivgan R, Medina J, Dong Z, Rojas-Cessa R. Multidepot drone path planning with collision avoidance. IEEE Internet of Things Journal, 2022, 9(17): 16297-16307.