International Journal of Information Systems and Supply Chain Management
Volume 17 • Issue 1

Design and Application of Legal Information Systems Based on Big Data Technology

Ying Wang, Zhengzhou Technology and Business University, China*

ABSTRACT

China's exploration of the legal application of big data is still far from thorough enough. This article proposes a universal suitable for hierarchical big data storage systems, which can quickly add new cache policies when needed and provide cache scheduling strategies. The neural training algorithm used in legal information systems implements a complete parallel computing framework for large-scale neural network training, supporting distributed storage and management of large-scale sample data. The experimental results show that the framework has good scalability and fault tolerance, and can quickly train legal information systems, improving their efficiency and response speed. This provides new ideas and methods for the design and development of legal information systems.

KEYWORDS

Big Data Storage, Big Data Technology, Legal Information System, Neural Training Algorithm

1. INTRODUCTION

In recent years, with the rapid development of information technology and the digital transformation of the legal field, big data has received increasing attention in legal applications (Joubert et al., 2023). However, the exploration of big data in legal applications in China is still far from being in-depth enough (Cui et al., 2023). This article proposes a universal and scalable cache scheduling framework for layered big data storage systems, which can quickly add new cache policies when needed and provide cache scheduling strategies for different data access modes; thereby, accelerating data read and write access performance in upper layer big data applications. In addition, the neural training algorithm used in the legal information system not only considers the parallelization algorithm of neural network training, but also implements a complete parallel computing framework for large-scale neural network training, supporting distributed storage and management of large-scale sample data. The experimental results show that the framework has good scalability and fault tolerance, which can quickly train legal information systems and improve their efficiency and response speed. This provides new ideas and methods for the design and development of legal information systems. In the future, we will continue to explore the potential of big data in legal applications, while improving and perfecting the proposed frameworks and algorithms to adapt to the constantly evolving environment and needs.

DOI: 10.4018/IJISSCM.338380 *Corresponding Author

2. LITERATURE REVIEW

The current big data processing technology system is not yet mature and perfect. Due to the complexity, diversity, and huge data scale of big data, there are still many technical problems that need to be continuously researched and improved in the existing big data processing technologies and system platforms, including: distributed storage management technologies and systems for Platform, other platforms, efficient algorithm design, and easy-to-use tools (Ngo et al., 2023). For such informatization today, importance of informatization for the legal domain has become more and more prominent (Duggineni, 2023). Many countries believe that legal informatization is another major driving force for legal reform and can be vigorously undergone (Lyu et al., 2023). The United States has been the first to start the process of "law informatization," which clearly expresses the irreplaceable role of information technology in the future legal industry (Himeur et al., 2023). In the country of study, the construction of informatization laws and regulations is synchronized, especially within the country's rule of law (Pathak et al., 2023). Before the reform and opening up, due to the repeated impact of the national rule of law construction, the construction of informatization laws and regulations was relatively backward, lacking basic legal norms, and no systematic concept was proposed (Vasa & Thakkar, 2023). After the reform and opening up, the Chinese government began to realize the importance of information technology and informatization (Bi et al., 2023). The government work report in 1978 proposed to "accelerate the development of research on integrated circuits and electronic computers, and make them widely used in various fields" (Khan et al., 2023). When the "863" plan was formulated in 1986, the country had listed information technology as an important topic and started Research on the country's informatization issues (Teubner et al., 2023). In 1987, the country established the National Information Center and set up a policy research institute within the center, focusing on research on information regulations and policy issues. Additionally, the country organized the "Information and Information Technology Legislative Collections," "China Information Legislation Environment Analysis and Legislative Discussion," "Legislative Framework Suggestions in the Process of Informatization," and other internal materials (Rosati et al., 2023).

At present, the domestic discussion and research on legal issues in the context of big data is still in its infancy, and there is no article that systematically expounds how big data should be applied in legal research (Huang et al., 2023). As for the scientific legislative proposition related to the title of this article, there is currently only the article "Internet, Big Data, Artificial Intelligence and Scientific Legislation" by Jiang Bixin and Zheng Lihua. Proposing future legislative work needs to respect the objective results of statistics (Filgueiras & Lui, 2023). This article takes the current background of China's legislative development as the breakthrough point and discusses several key issues that need to be paid attention to in the application of big data technology in the practice of scientific legislation propositions, but the article does not discuss the combination of scientific legislation and big data (Kumar, 2023). Some basic issues are discussed in detail, such as the analysis of the motivation for the application of big data in scientific legislation. The impact of big data on the development of scientific legislative propositions, of course, is related to the discussion direction selected by the article, but these issues require future research and more attention (Davidson et al., 2023).

Regarding the specific application of big data in various aspects of legislation, Cao Hanyu's "Analysis on the Application of Big Data in Post-legislative Evaluation" starts from the problems existing in the traditional post-legislative evaluation system and evaluates the concept of big data and big data technology (Al-Dmour et al., 2023). It explores the feasibility of applying it to post-legislative evaluation work, and then the article puts forward the necessity of big data technology as the necessary support for post-legislative evaluation technology through necessity analysis (Al-Okaily et al., 2023). When it comes to the impact of the development of big data technology on legal research methods and legal thinking patterns, Fu Yong's "The Reference of Big Data Methods to Legal Research" summarizes the characteristics, functions, and research methods of current big data in detail. Examples of big data application in practice include big data to promote the transformation

of legal research thinking, big data to enhance the method of legal empirical research, the impact of big data application on legislative activities, and the impact of big data application on judicial activities (Karkošková, 2023). On the one hand, it discusses the all-round combination of big data and jurisprudence at the theoretical and practical levels.

Judging from the above research, the current domestic discussion on the application of big data in law is still relatively one-sided and superficial (Saragih et al., 2023). When discussing the impact of big data on legal development, the main perspective is still the protection of personal privacy, data security, information publicity, and other issues. These aspects have been paid attention to by law research institutes in the Internet era, but these are far from all legal issues in the era of big data. The development of law in the era of big data needs to grasp the technical characteristics of big data systems and the influence of these characteristics (Andronie et al., 2023). The resulting new interest pattern can effectively respond to the public's expectations for the innovation and development of law in the era of big data. In particular, some basic issues in the field of legislation are involved. At present, academic research has not paid enough attention to the connection between the functional characteristics of big data and the specific links of legislation. Professor Jiang Bixin's article focuses more on the policy interpretation level, and the argument emphasizes the future (Zhang et al., 2023) and the necessity of legislative development and application of big data. As for how the scientific legislative proposition can effectively apply the functions and values of big data at the theoretical and practical levels, there is still no systematic research results (Sanchez et al., 2023).

However, the current big data processing technology system is not yet mature and perfect. Due to the complexity, diversity, and huge data scale of big data, there are still many technical problems that need to be continuously researched and improved in the existing big data processing technologies and system platforms. The country of study entered the information age for the 20th century and began the process of "law informatization" (Ryalat et al., 2023). To sum up, the development of legal informatization in this country is slower, and it is necessary to speed up the design and application of legal informatization systems in the country, in order to benefit all walks of life. We must set up a legal information system belonging to the country to promote the country's scientific legislation, standard legislative process, and judicial system progress. The study will focus on comprehensive big processing technology, builds the legal information system, and conducts research on the distributed storage management, the performance optimization of the parallel computing system for legal information system, and the optimization of the training method. And draws the following conclusions: (1) Since the legal information system has ones, it needs memory to save, and different caching strategies and different data access modes will affect the performance of the legal information system. But even in real big data application scenarios, where data access patterns are not very regular, ARC and LIRS caching strategies can still be used to speed up big data applications with irregular access patterns. (2) In the distributed data access experiment of the legal information system, the memory-centric distributed file system can speed up about 100 times the performance of the disk-based distributed file system. (3) The cNeural training algorithm adopted by the legal information system not only considers the neural network training parallelization algorithm, but also researches and implements a complete parallel computing framework for large-scale neural network training, which supports distributed storage management from large-scale sample data to the integrated processing of parallelized training calculation and has good system scalability and fault tolerance, which can realize the rapid training of legal information systems.

3. METHODOLOGY

3.1 Research on Big Data Distributed Storage Technology and Systems

This paper proposes a generalized and scalable cache scheduling framework for hierarchical big data storage systems for the design of legal information systems. The framework allows new cache strategies to be added quickly when needed in a pluggable manner. When discussing the impact of

big data on legal development, the main perspective is still the protection of personal privacy, data security, information publicity, and other issues.

Based on the above framework, this paper implements and provides a set of cache scheduling strategies covering different data access modes to accelerate the data read and write access performance in upper-layer big data applications. The provided caching strategies include LRFU (LeastRecently/Frequently Used), LIRS (Low Inter-reference Recency Set) and ARC (AdaptiveReplacement Cache) strategy.

For Recency-friendly data access mode, the basic form is:

$$\left(a_1, a_2, \cdots, a_{k-1,} a_k, a_{k-1}, \cdots, a_2, a_1\right)^N \tag{1}$$

Among them, k represents the number of data blocks, and N represents the number of loop visits.

Among them, k will be the bricks of the visited numbers, while A will be the time of the data bricks. M will be the number of the data, which is for the one visit, and p will be the m ones for the rate being visited. N1 will be the term of the all data number.

The basic form is:

$$\left(\left(a_1, a_2, \cdots, a_{k-1,} a_k\right)^A P_\varepsilon \left(b_1, b_2, \cdots, b_m\right)\right)^N \tag{2}$$

Among them, k will be the loops. When k < cache size, the data blocks accessed more than once can be completely stored in the cache. When k + m > cache size, the cache cannot store all the data blocks, and some data blocks are replaced by the current cache. A good replacement strategy in this scenario should preferentially replace data blocks that are only accessed once.

In this access mode, data blocks in the system are accessed with an uneven frequency. Caching more frequently accessed data blocks will lead to greater access performance improvements.

The basic form is:

$$\left(a_1, a_2, \cdots, a_{k-1,} a_k\right)^N \tag{3}$$

Among them, k represents the number of data blocks accessed by the loop, and N represents the number of loops.

The data access mode means that the data block is continuously accessed in a circular fashion. This access pattern is common in big data iterative computing applications such as K-Means and PageRank. In this data access mode, replacing the most recently accessed data block will achieve better results, while replacing the data block that has not been accessed for the longest time will cause each accessed data block to be out of memory.

For mixed data access mode, the basic form is:

$$\left(\left(a_1, a_2, \cdots, a_{k-1,} a_k, a_{k-1}, \cdots, a_2, a_1\right)^{N_1} \left(\left(a_1, a_2, \cdots, a_{k-1,} a_k\right)^A P_\varepsilon (b_1, b_2, \cdots, b_m)\right)^{N_2}\right)^N \tag{4}$$

Among them, k will be the bricks of the visited numbers, while A will be the time of the data bricks. M will be the number of the data, which is for the one visit, and p will be the m ones for the rate being visited. N1 will be the term of the all data number.

Figure 1 depicts the overall framework of a generalized tiered storage system cache scheduling. The goal of this framework is to design and provide a generalized and extensible hierarchical cache scheduling framework for hierarchical big data storage systems, through which different data access modes and cache strategies and upper-layer big data can be integrated. The applications are integrated and provide users with a set of efficient cache scheduling strategies covering a variety of different data access modes; thereby, accelerating the data read and write access performance of upper-layer big data applications.

The above cache scheduling framework itself has good versatility and is platform independent, and its architecture, access mode, and cache strategy can be applied to any hierarchical storage system. With a set of efficient cache scheduling strategies covering a variety of different data access modes, it can accelerate the data read and write access performance of upper-layer big data applications. However, with the aim to specifically verify the correctness of the framework, for such work, the most typical Alluxio will be used as the verification implementation system and test bed.

Figure 2 describes the layered distributed storage architecture of the Alluxio system, in which each worker data storage server worker node includes a three-tier storage system of MEM-SSD-HDD.

LRFU (Least Recently/Frequently Used) combines the LRU strategy. When it occurs, LRFU replaces the data block with the smallest one.

$$CRF_{t_{base}}(b) = \sum_{i=1}^{K} F(t_{base} - t_{b_i}) \tag{5}$$

In the above formula, CRF combines the contribution of it F(t) is defined in equation (6). F(t) represents it:

$$F(t) = \left(\frac{1}{attenuation}\right)^{step \times t} \tag{6}$$

Figure 1. Cache scheduling framework of hierarchical storage system covering multiple access modes and caching strategies

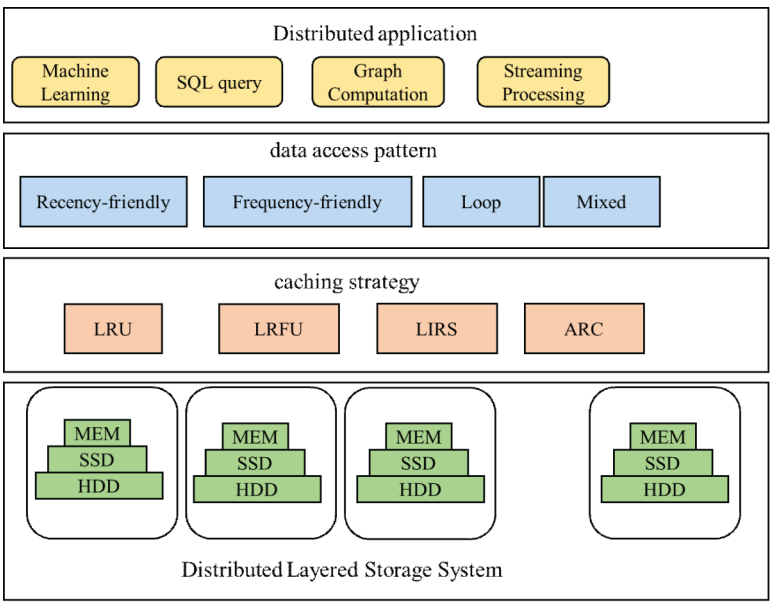
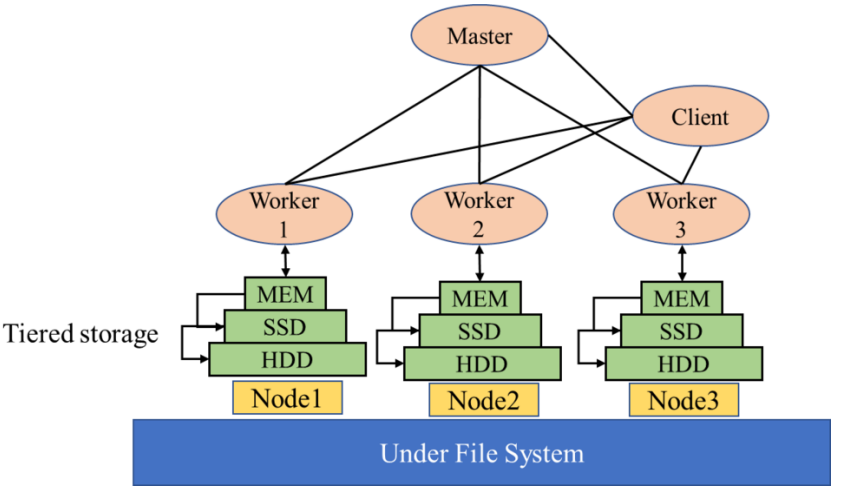


Figure 2. The overall architecture of Alluxio



Among them, step is a weight adjustment parameter. If its value is equal to 0, then LRFU degenerates into LFU; if step=1, then LRFU evolves into LRU. Therefore, step controls whether the behavior of LRFU is closer to LRU or LFU.

3.2 Research on Performance Optimization of Big Data Parallel Computing Systems

Parallel computing is the process of decomposing a task into several small tasks and executing them together to complete the solution. It is an effective way to enhance the ability to solve complex problems and improve performance. Deep learning, knowledge graphs, and robots are all inseparable from distributed computing resources. Big data and parallel computing bring countless opportunities. These knowledge models, as external brains, provide intelligent decisions and actions for humans and robots, making artificial intelligence a reality. Computational resources can be shared, collaborated, and complete tasks together, and they constitute the strongest brain. The Internet of Things provides input and action, big data provides knowledge models and intelligence, and parallel computing provides tools. They complement each other and achieve the ultimate artificial intelligence.

The amount of data in the legal information system is huge, and parallel computing is required. This article focuses on the scenarios where the mainstream big data parallel computing system Hadoop MapReduce executes short jobs, and the big data parallel computing system Spark has high consumption of JVM heap storage carried out related performance optimization research.

3.3 Research on Neural Network Training Algorithms

The legal information system needs to be trained with a large amount of data to achieve the expected effect. This paper adopts the neural network training algorithm to achieve the effect.

Backpropagation algorithm is a common algorithm used to train neural networks (Fernando et al., 2023). It calculates the gradient of network parameters, backpropagates errors from the output layer to the hidden layer, and updates the parameters using gradient descent. The key to backpropagation algorithm lies in updating parameters through gradient descent, gradually adjusting the network, and reducing errors. It can effectively train multi-layer neural networks and achieve good performance in various tasks. In addition, backpropagation algorithms can also be combined with other techniques, such as regularization and batch normalization, to further improve the performance and robustness of the model. The backpropagation algorithm uses gradient descent to update the weights of the neural network, which can efficiently calculate and update weights when training on large-scale datasets,

accelerating the learning process. The backpropagation algorithm can accelerate the training process through parallel computing. Due to the independent gradient calculation and weight update of each neuron, the ability of parallel computing can be utilized to improve computational efficiency. The backpropagation algorithm can provide information about the contribution of each weight and bias to the network output; thereby, explaining the decision-making process of the network. This is very helpful for understanding and analyzing the behavior of the network.

The following are the basic steps of the backpropagation algorithm:

(1) Forward Propagation: The input samples are propagated forward through a neural network to calculate the output values of each layer. Starting from the input layer, non-linear transformation is performed on the weighted inputs of each neuron through activation functions to obtain the output values of each layer.
(2) Compute Error: Compare the network output with the target value and calculate the error between the predicted value and the true value. Usually, a loss function (such as mean square error or cross entropy loss) is used to measure the size of the error.
(3) Backpropagation: Starting from the output layer, calculate the gradient for each layer according to the chain rule. Firstly, calculate the gradient of the output layer, and then sequentially calculate the gradient of the hidden layer forward. Obtain the gradient for each neuron based on the gradient and the derivative of the activation function.
(4) Update Parameters: Based on the gradient descent method, use the calculated gradient to update the network parameters. By multiplying the parameters with the learning rate and subtracting the product of gradients, parameter updates are achieved. The learning rate determines the step size for each update.
(5) Iteration: Repeat the steps of forward propagation, error calculation, backpropagation, and parameter updates until the preset stopping conditions are reached (such as reaching the maximum number of iterations or meeting the error requirements).

But a specific parallel optimization design should be carried out according to specific algorithms. In this study, through the research, design, and implementation of several typical and more complex big data analysis parallel algorithm design cases, the author seeks to discuss the parallel design and computing performance optimization methods of machine learning and data analysis algorithms based on mainstream big data processing platforms.

First, the process (Backpropagation Algorithm) is introduced.

In the forward stage, the input layer takes the input signal and passes it to each neuron in the hidden layer. The hidden layers then process these signals and pass the processing results to the output layer. For an input vector X, the input and output signals of each neuron in the hidden layer are labeled uj and hj, and these two signals can be calculated by formula (7) and formula (8), respectively:

$$u_j = \sum_{i=1}^{m} W_{ij} x_i + \theta_j,\ j = 1,2,\ldots,m \tag{7}$$

$$h_j = f(u_j) = \frac{1}{1+\exp(-u_j)},\ j = 1,2,\cdots,q \tag{8}$$

After the output layer obtains the signal from the hidden layer, subsequent processing is also required. The input signal lk and output signal ck of the neurons in the output layer are calculated by formula (9) and formula (10), respectively.

$$l_k = \sum_{j=1}^{q} V_{jk} h_j + \gamma_k, \ k = 1,2,\ldots,n \tag{9}$$

$$c_k = f(l_k) = \frac{1}{1+\exp(-l_k)}, \ k = 1,2,\ldots,n \tag{10}$$

So far, the neural network model weights W, V, and bias θ, y do is called Error. In the backward process, formula (11) is used to calculate the dk, and then formula (12) is used to calculate the deviation ej.

$$d_k = (y_k - c_k)c_k(1-c_k), \ k = 1,2 \tag{11}$$

$$e_j = \left(\sum_{k=1}^{n} d_k V_j\right) h_j(1-h_j), \ j = 1,2 \tag{12}$$

Bias is fed back from the output layer to the hidden layer. Through this bias back-propagation method, the connection weights of the output layer and the hidden layer are updated using formulas (13) and (14). Further, use formula (15) to update the connection weight between the hidden layer and the input layer.

$$V_{jk}(N+1) = V_{jk}(N) + \alpha_1 d_k(N) h_j \tag{13}$$

$$W_{ij}(N+1) = W_{ij}(N) + \alpha_2 e_j(N) \tag{14}$$

$$\theta_j(N+1) = \theta_j(N) + \alpha_2 e_j(N) \tag{15}$$

In the above formula, i=1,2,..,m, j=1,2..,q; k = 1,2,..,n. α1 and α2 are learning rates ranging from 0 to 1. N represents the number of the current training round. The backpropagation for accumulated △W will be there.

4. RESULTS AND ANALYSIS

4.1 Evaluation of Storage Performance of Legal Information Systems

Because the legal information system has a large amount of data, it needs a large-capacity memory for storage. Different caching strategies and different data access modes will affect the performance of the legal information system.

Therefore, this paper uses Alluxio-Perf to evaluate the performance of different caching strategies under different data access modes. The experiment implements various data access patterns in Alluxio-Perf, adding test cases Recency-friendly, Frequency-friendly, Loop, and Mixed.

As shown in Figure 3, the LIRS caching strategy achieves the best hit rate in the Loop test case, 20% to 40% higher than other caching strategies, while saving up to 50 seconds of execution time compared to other strategies. This is because data blocks in LR space will not be replaced to the bottom layer in Loop access mode. Therefore, the data blocks in the LR space will always be hit, so the LRS has a higher hit rate for the Loop access mode. In addition, LRS, like LRFU (step=0.02, step=0) and ARC, has achieved good results in Frequency-friendly test cases.

ARC achieves the best results in the Mixed test case, with a 20% to 50% higher hit rate than other caching strategies. As can be seen from Figure 4, ARC saves up to 150 seconds of execution time compared to other caching strategies. The reason why ARC achieves the best results in the Mixed test case is that ARC automatically adjusts the allocation to Recency-friendly access mode and

Figure 3. The hit rate of the cache strategy in the Alluxio-Perf test case

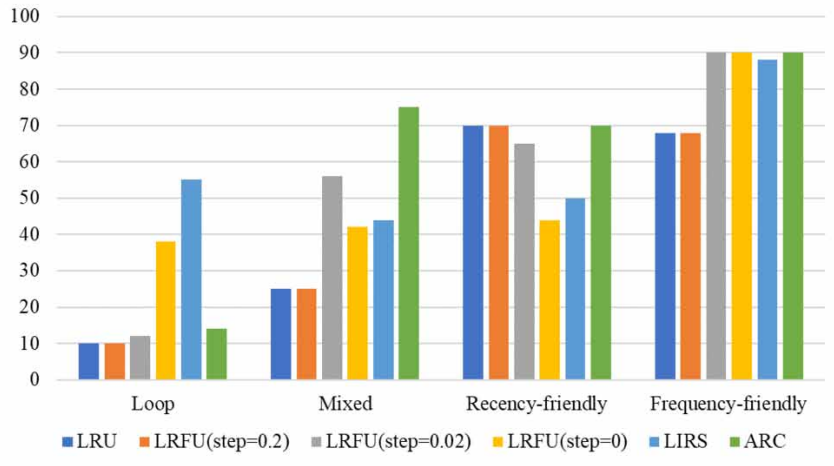
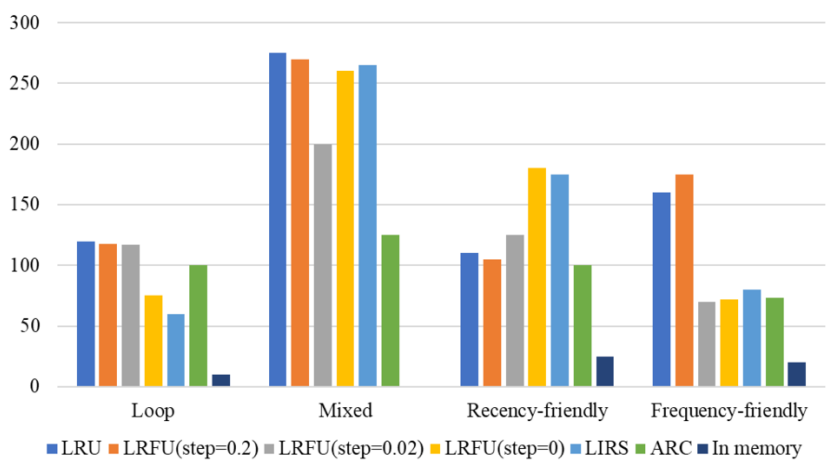


Figure 4. Execution time of caching strategy in Alluxio-Perf test case



Frequency-friendly access mode space to improve ARC's current hit rate. In addition to the Mixed access mode, ARC, LRU, LRFU (step-0.2) also achieved good results in the Recency-friendly test case, and the same as LRFU (step=0.02, step-0), LIRS obtained in the Frequency-friendly access mode better effect.

Therefore, it can be concluded that even in the real big data application scenarios where the data access patterns are not very regular, these caching strategies can still be used to well accelerate big data applications with irregular access patterns.

In addition to how the cache strategy will affect the performance of the legal information system, the selection of the distributed file data access mode will also affect the system performance.

The operations included in the distributed file data access mode are: metadata operations, sequential reads, sequential writes, and random reads. Figure 5 shows the proportion of these operations in different applications. As can be seen from the figure, the amount of data read by various applications is much larger than the amount of data written to the file. Both K-Means and Bayes tend to read data sequentially, while data query applications include a large number of random read operations.

There are usually two different system architectures for metadata management of distributed file systems: centralized architecture and decentralized architecture. A distributed file system with a centralized architecture has a similar working method. The application always first obtains the location of the target file from the master node that manages metadata, and then goes to the corresponding slave node to access the specific file data, representing a system such as HDFS. In a distributed file system, with a distributed architecture, the file location is usually determined by the client, and all clients share a unified hash algorithm. The location of the target file is obtained by calculating the hash value, and then the corresponding node accesses the specific file data. In a centralized architecture, since all metadata operations are done in the memory of the master node, it is more conducive to metadata operations. The distributed architecture needs to synchronize data in the cluster when performing metadata operations, which has a large network overhead, and will affect the operation performance of the distributed architecture metadata.

Figure 6 tests the basic read and write performance of different distributed file systems. The operating system's cache was emptied before each experiment in order to accurately assess the performance of each operation. It can be observed that a memory-centric distributed file system can perform about 100 times faster than a disk-based distributed file system. The read and write

Figure 5. The proportion of data involved in different file operations

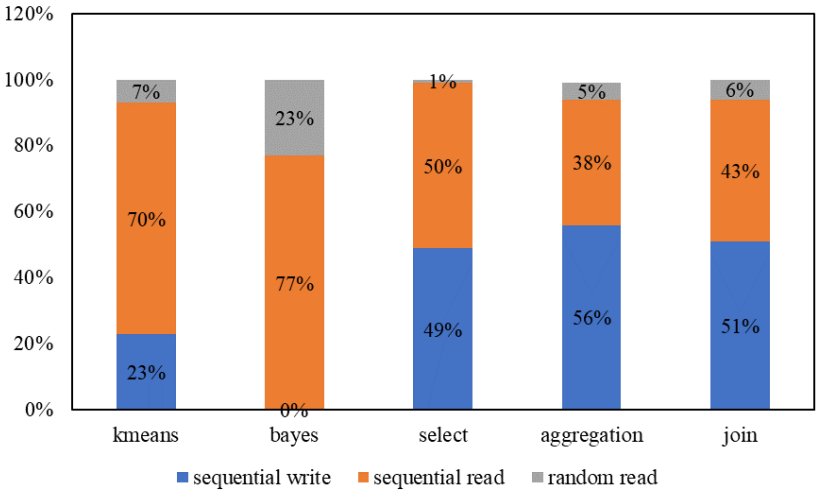
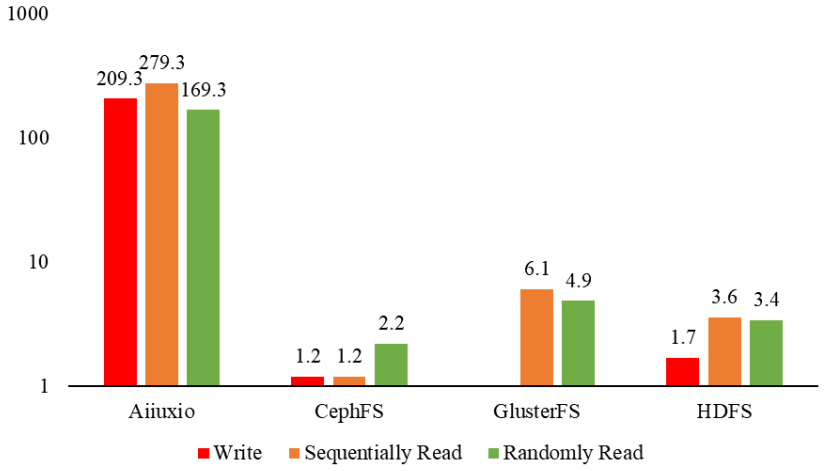


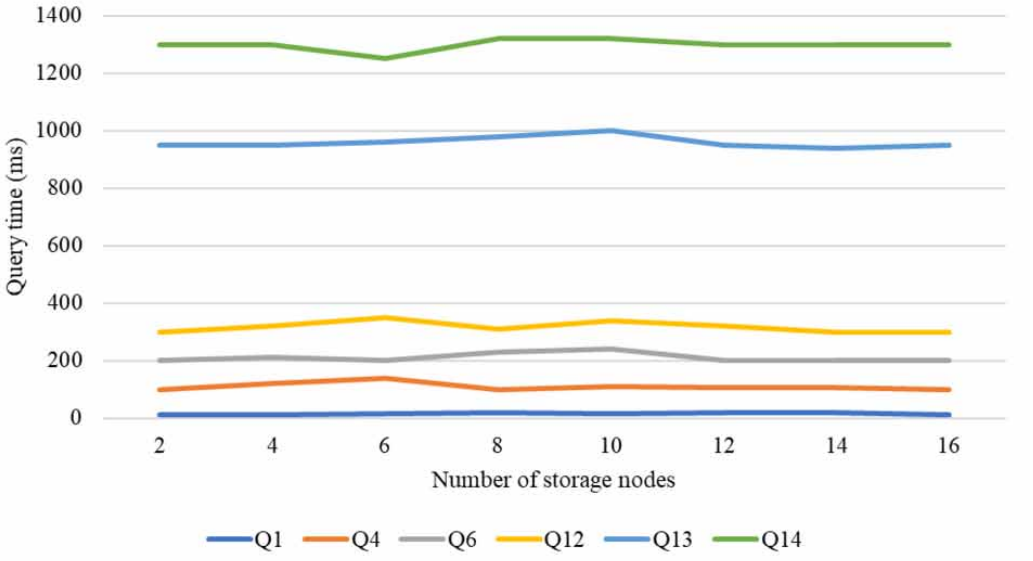
Figure 6. Basic read and write performance (Overall cluster throughput)



performance of Alluxio is better than other comparative systems because all its read and write operations are in local memory. In addition, different fault tolerance mechanisms also have a great impact on read and write performance. CephFS, GlusterFS, and HDFS require multiple copies of data to the cluster to complete fault tolerance, while Aluxio can choose to use lineage and checkpoint mechanisms for fault tolerance.

The scalability of Rainbow-IM is evaluated with different node numbers on the LUBM-100 dataset. The experimental results are shown in Figure 7. For all queries, the time-consumption of Rainbow-M is almost unchanged. This is because Rainbow-M's index can be kept in memory using

Figure 7. Query time-consuming changes when adjusting the number of storage nodes



a small number of nodes. The new node increases the distributed memory space of Rainbow-IM and increases the potential of the system to store and manage larger-scale RDF data.

4.2 Execution Performance Optimization Analysis of Legal Information Systems

In order to test and optimize the execution performance of the legal information system, this experiment is carried out on Hadoop 1.0.3 and SHadoop. The experimental cluster contains one master node and 36 slave nodes. The master node is configured with two 6-core 2.8 GHz Intel Xeon processors, 36 GB of memory, and two 2 TB 7200 RPM SATA hard drives. Each slave node is configured with two 4-core 2.4 GHz Intel Xeon processors, 24 GB of memory, and two 2TB 7200 RPM SATA hard drives. All these nodes are interconnected via a 1 Gb/s Ethernet. They are all installed with the RedHat Enterprise Linux 6 operating system with kernel version 2.6.32 and the file system of Ext3.

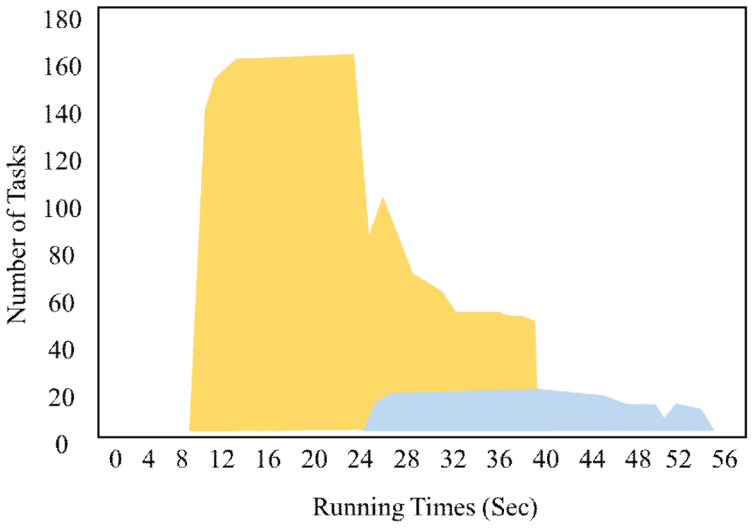
Each slave node runs Hadoop's TaskTracker/DataNode process, and the master node runs Hadoop's JobTracker/NameNode process. In the Hadoop software configuration, the number of Map/Reduce slots for each node is 8, and the rest use the default configuration. Both standard Hadoop and SHadoop run on OpenJDK 1.6, and the JVM heap size is 2GB.

SHadoop adds two optimizations over the standard Hadoop MapReduce framework. Therefore, the goal of the first set of experiments is to evaluate the effect of each optimization method, and the evaluation metric is the job execution time.

First, run the evaluation optimization through the classic WordCount test program. In order to keep the job running time relatively short, the size of the input data is 4.5 GB, including about 200+ data blocks. This group of experiments uses 20 slave nodes with a total of 160 task slots, and runs 16 Reduce tasks in Hadoop and SHadoop. During the execution of the entire job, the JobTracker side collects and records the load status of all TaskTrackers at various times. The relevant experimental results are shown in Figure 8.

SHadoop mainly includes two important optimizations: optimizing the process of starting and ending jobs and providing a messaging mechanism for instant communication to quickly communicate urgent events during task execution. The first optimization can reduce the time consumption of the job startup and cleanup phases, which is especially effective for short jobs. The second optimization works well for most short jobs with many tasks.

Figure 8. Running wordcount standard test case on standard Hadoop



Compared with standard Hadoop, SHadoop can achieve an average performance improvement of 25% on various Hadoop benchmarking programs and Hive applications. SHadoop also has good data and system scalability. In addition, SHadoop retains the standard HadoopMapReduce programming interface, so SHadoop is compatible with existing applications developed based on standard Hadoop. SHadoop optimization work has also been tested at Intel's internal production level and integrated into the Intel Hadoop distribution.

In the next step of work, it is planned to explore more optimization methods to further improve the execution performance of MapReduce jobs, including dynamically setting the number of slots in the Hadoop cluster according to the actual load and rationally allocating the nodes for task execution according to the characteristics of the job and task, etc., so as to achieve execution performance optimization of legal information systems.

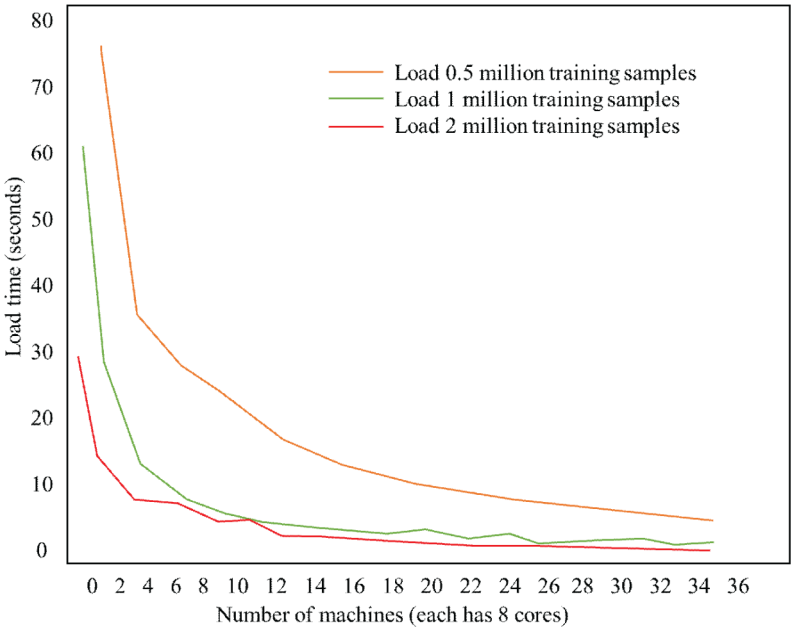
4.3 Performance Evaluation of Legal Information System Training Methods

The legal information system needs to be trained before it can really play a role in practice. This experiment is used to estimate the impact of the cNeural training algorithm on the performance of the legal information system. The experiments were performed in the 1.6.

The curves in Figure 9 show it failed to be successfully executed due to the limited resources of the single machine configuration. In the distributed data access experiment of the legal information system, the memory-centric distributed file system can speed up about 100 times the performance of the disk-based distributed file system.

The cNeural training algorithm adopted by the legal information system not only considers the neural network training parallelization algorithm, but also researches and implements a complete parallel computing framework for large-scale neural network training, which supports distributed storage management from large-scale sample data to the integrated processing of parallelized training

Figure 9. Training data loading performance of the cNeural system



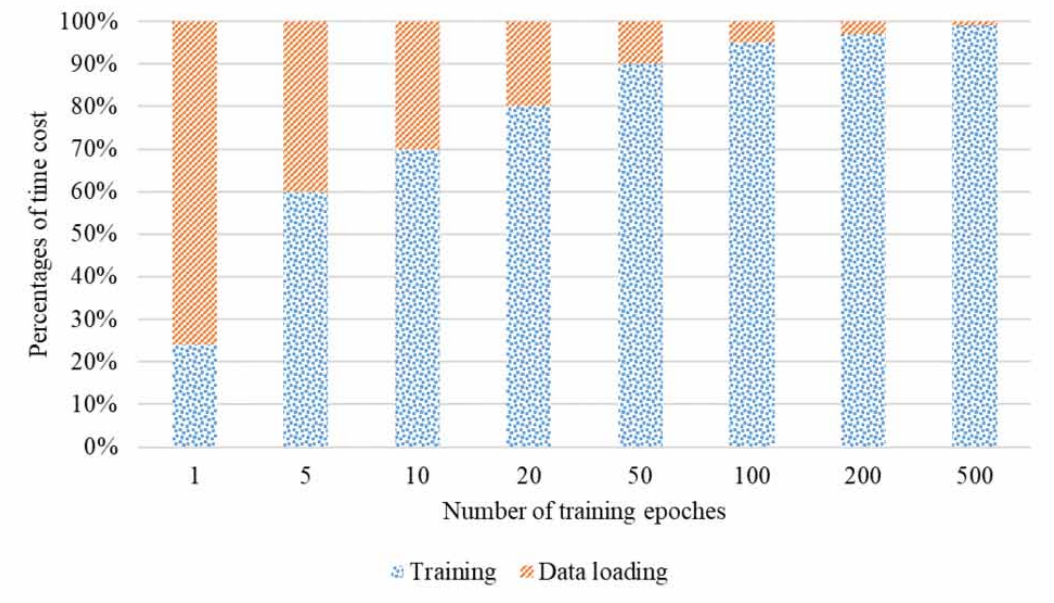
calculation, and has good system scalability and fault tolerance, which can realize the rapid training of legal information system as shown in Figure 10.

4.4 Analysis of Practical Applications

With the rapid development of information technology, legal information systems are playing an increasingly important role in modern legal work. However, due to the particularity of the legal field and the continuous growth of data scale, effective management and efficient utilization of a large amount of legal data have become an urgent problem to be solved. In order to improve the performance and efficiency of legal information systems, this paper proposes an innovative method. This method improves the processing capability of legal documents, knowledge graphs, and large-scale legal data by optimizing data access and query processes, and supports applications such as intelligent legal recommendations. Next, we will further introduce the practical application of this method in legal information systems.

(1) Legal document management: Legal information systems require the management and retrieval of a large number of legal documents, including judgments, legal provisions, cases, etc. By using this framework, the speed of reading and retrieving legal documents can be improved, and the efficiency of legal professionals can be enhanced.
(2) Construction of legal knowledge graph: A legal knowledge graph is an important legal information resource that can assist legal professionals in case analysis, legal policy research, and other related work. This framework can accelerate access to legal knowledge graphs, providing fast graph queries and inference functions.
(3) Legal data analysis and mining: The application of big data technology in the legal field is becoming increasingly widespread, such as predicting case outcomes and discovering legal rules by analyzing large-scale legal case data. This framework can improve the efficiency of processing and analyzing large-scale legal data, supporting faster and more accurate legal data mining.

Figure 10. Time-Consuming ratio of data loading and training execution under different iterations



(4) Legal intelligent recommendation system: Based on user needs and historical data, legal intelligent recommendation system can provide personalized legal services and suggestions for users. This framework can provide efficient data access and query functions, support real-time and personalized recommendation algorithms, and accelerate the operation of legal intelligent recommendation systems.

In summary, the method proposed in this paper can help improve the application efficiency of data management, knowledge graph construction, data analysis and mining, and intelligent recommendation in legal information systems, providing better work support and decision-making basis for legal professionals.

The method of this paper still has some limitations in practical application:

(1) Uneven data distribution: Due to the particularity of legal data, some categories of data may have fewer quantities, while others may have more quantities, resulting in uneven data distribution. This may impact the effectiveness and accuracy of model training.
(2) Difficulty in parameter tuning: This method requires selecting appropriate parameters and model architecture, but it is not easy to obtain the appropriate parameter selection. In addition, the optimal parameter settings may also vary on different datasets.

To address the above limitations, the following measures can be taken:

(1) Data augmentation and balancing processing: By using data augmentation techniques, the dataset can be expanded, the problem of imbalanced data distribution can be improved, and the generalization ability of the model can be enhanced. Meanwhile, data balance processing techniques can be used to replicate or synthesize data from categories with fewer samples, in order to achieve data balance.
(2) Model tuning and evaluation: Through techniques such as cross validation, the performance and accuracy of models under different parameter settings can be evaluated, and the optimal parameter settings and model architecture can be selected. In addition, to avoid overfitting on specific datasets, techniques, such as regularization, can be used to prevent model overfitting.
(3) Continuous improvement and optimization: With the continuous development of data and technology, the demand for legal information systems is also constantly changing. Therefore, it is necessary to continuously monitor new data and technological trends and to optimize and improve methods to meet constantly changing needs.

Through the above measures, we can better address the limitations of the method proposed in this paper, improve the performance and applicability of the model, and promote the successful implementation of legal information systems in practical applications.

5. CONCLUSION

In recent years, with the rapid development of information technology and the digital transformation of the legal field, the application of big data in law has received increasing attention. However, the discussion on the application of big data in law in China is still relatively one-sided and superficial. This article proposes a universal and scalable hierarchical big data storage system cache scheduling framework for the design of legal information systems. This framework allows for the quick addition of new caching policies in a pluggable manner when needed. The cache scheduling strategies

provided in this article for different data access modes can accelerate the data read and write access performance in upper level big data applications. The neural training algorithm used in the legal information system of this article not only considers the parallelization algorithm of neural network training, but also studies and implements a complete parallel computing framework for large-scale neural network training, supporting distributed storage and management of large-scale sample data. The experimental results demonstrate that it has good system scalability and fault tolerance, and can achieve rapid training of legal information systems. This article improves the efficiency and response speed of legal information systems, providing new ideas and methods for the design and development of legal information systems. Due to time and resource constraints, the paper may not have conducted sufficient real case verification to fully demonstrate the effectiveness and feasibility of the proposed framework in practical applications. In the future, we will consider further verifying real cases, collecting and analyzing actual data, and evaluating the actual performance and effectiveness of the proposed framework in different contexts. We will continue to monitor the development trends in the fields of information technology and law, and combine them with practical application needs to continuously improve and perfect the proposed framework and algorithms to adapt to constantly changing environments and needs.

DATA AVAILABILITY

The figures used to support the findings of this study are included in the article. The data analyzed during the current research period can be obtained from the communication author according to reasonable requirements.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

FUNDING STATEMENT

This work was not supported by any funds.

ACKNOWLEDGMENT

The authors would like to show sincere thanks to those techniques who have contributed to this research.

REFERENCES

Al-Dmour, H., Saad, N., Basheer Amin, E., Al-Dmour, R., & Al-Dmour, A. (2023). The influence of the practices of big data analytics applications on bank performance: Filed Filed study. *VINE Journal of Information and Knowledge Management Systems*, *53*(1), 119–141. doi:10.1108/VJIKMS-08-2020-0151

Al-Okaily, M., Alkhwaldi, A. F., Abdulmuhsin, A. A., Alqudah, H., & Al-Okaily, A. (2023). Cloud-based accounting information systems usage and its impact on Jordanian SMEs' performance: The The post-COVID-19 perspective. *Journal of Financial Reporting and Accounting*, *21*(1), 126–155. doi:10.1108/JFRA-12-2021-0476

Andronie, M., Lăzăroiu, G., Iatagan, M., Hurloiu, I., Ștefănescu, R., Dijmărescu, A., & Dijmărescu, I. (2023). Big Data Management Algorithms, Deep Learning-Based Object Detection Technologies, and Geospatial Simulation and Sensor Fusion Tools in the Internet of Robotic Things. *ISPRS International Journal of Geo-Information*, *12*(2), 35. doi:10.3390/ijgi12020035

Bi, Z., Jin, Y., Maropoulos, P., Zhang, W. J., & Wang, L. (2023). Internet of things (IoT) and big data analytics (BDA) for digital manufacturing (DM). *International Journal of Production Research*, *61*(12), 4004–4021. do i:10.1080/00207543.2021.1953181

Cui, Y., Ma, Z., Wang, L., Yang, A., Liu, Q., Kong, S., & Wang, H. (2023). A survey on big data-enabled innovative online education systems during the COVID-19 pandemic. *Journal of Innovation & Knowledge*, *8*(1), 100295. doi:10.1016/j.jik.2022.100295

Davidson, E., Wessel, L., Winter, J. S., & Winter, S. (2023). Future directions for scholarship on data governance, digital innovation, and grand challenges. *Information and Organization*, *33*(1), 100454. doi:10.1016/j.infoandorg.2023.100454

Duggineni, S. (2023). Impact of controls on data integrity and information systems. *Science and Technology*, *13*(2), 29–35.

Fernando, Y., Tseng, M. L., Wahyuni-Td, I. S., de Sousa Jabbour, A. B. L., Chiappetta Jabbour, C. J., & Foropon, C. (2023). Cyber supply chain risk management and performance in industry 4.0 era: Information Information system security practices in Malaysia. *Journal of Industrial and Production Engineering*, *40*(2), 102–116. doi :10.1080/21681015.2022.2116495

Filgueiras, F., & Lui, L. (2023). Designing data governance in Brazil: An An institutional analysis. *Policy Design and Practice*, *6*(1), 41–56. doi:10.1080/25741292.2022.2065065

Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., Bensaali, F., & Amira, A. (2023). AI-big data analytics for building automation and management systems: A A survey, actual challenges and future perspectives. *Artificial Intelligence Review*, *56*(6), 4929–5021. doi:10.1007/s10462-022-10286-2 PMID:36268476

Huang, Y., Li, Y. J., & Cai, Z. (2023). Security and privacy in metaverse: A comprehensive survey. *Big Data Mining and Analytics*, *6*(2), 234–247. doi:10.26599/BDMA.2022.9020047

Joubert, A., Murawski, M., & Bick, M. (2023). Measuring the big data readiness of developing countries–index development and its application to Africa. *Information Systems Frontiers*, *25*(1), 327–350. doi:10.1007/s10796-021-10109-9

Karkošková, S. (2023). Data governance model to enhance data quality in financial institutions. *Information Systems Management*, *40*(1), 90–110. doi:10.1080/10580530.2022.2042628

Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A. M., & Luo, B. (2023). SQL and NoSQL database software architecture performance analysis and assessments—A systematic literature review. *Big Data and Cognitive Computing*, *7*(2), 97. doi:10.3390/bdcc7020097

Kumar, S., Lim, W. M., Sivarajah, U., & Kaur, J. (2023). Artificial intelligence and blockchain integration in business: Trends Trends from a bibliometric-content analysis. *Information Systems Frontiers*, *25*(2), 871–896. PMID:35431617

Lyu, X., Jia, F., & Zhao, B. (2023). Impact of big data and cloud-driven learning technologies in healthy and smart cities on marketing automation. *Soft Computing*, *27*(7), 4209–4222. doi:10.1007/s00500-022-07031-w

Ngo, V. M., Duong, T. V. T., Nguyen, T. B. T., Dang, C. N., & Conlan, O. (2023). A big data smart agricultural system: Recommending Recommending optimum fertilisers for crops. *International Journal of Information Technology : an Official Journal of Bharati Vidyapeeth's Institute of Computer Applications and Management*, *15*(1), 249–265. doi:10.1007/s41870-022-01150-1

Pathak, S., Krishnaswamy, V., & Sharma, M. (2023). Big data analytics capabilities: A A novel integrated fitness framework based on a tool-based content analysis. *Enterprise Information Systems*, *17*(1), 1939427. doi:10.1080/17517575.2021.1939427

Rosati, R., Romeo, L., Cecchini, G., Tonetto, F., Viti, P., Mancini, A., & Frontoni, E. (2023). From knowledge-based to big data analytic model: A A novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. *Journal of Intelligent Manufacturing*, *34*(1), 107–121. doi:10.1007/s10845-022-01960-x

Ryalat, M., ElMoaqet, H., & AlFaouri, M. (2023). Design of a smart factory based on cyber-physical systems and Internet of Things towards Industry 4.0. *Applied Sciences (Basel, Switzerland)*, *13*(4), 2156. doi:10.3390/app13042156

Sanchez, T. W., Shumway, H., Gordner, T., & Lim, T. (2023). The prospects of artificial intelligence in urban planning. *International Journal of Urban Sciences*, *27*(2), 179–194. doi:10.1080/12265934.2022.2102538

Saragih, A. H., Reyhani, Q., Setyowati, M. S., & Hendrawan, A. (2023). The potential of an artificial intelligence (AI) application for the tax administration system's modernization: The The case of Indonesia. *Artificial Intelligence and Law*, *31*(3), 491–514. doi:10.1007/s10506-022-09321-y

Teubner, T., Flath, C. M., Weinhardt, C., van der Aalst, W., & Hinz, O. (2023). Welcome to the Era of ChatGPT et al. *Business & Information Systems Engineering*, *65*(2), 95–101. doi:10.1007/s12599-023-00795-x

Vasa, J., & Thakkar, A. (2023). Deep learning: Differential privacy preservation in the era of big data. *Journal of Computer Information Systems*, *63*(3), 608–631. doi:10.1080/08874417.2022.2089775

Zhang, J., Li, S., & Wang, Y. (2023). Shaping a smart transportation system for sustainable value co-creation. *Information Systems Frontiers*, *25*(1), 365–380. doi:10.1007/s10796-021-10139-3

Ying Wang, born in Jiangxi, China in 1985, studied at East China Jiaotong University from 2002 to 2006, and received her bachelor's degree in 2006. From 2007 to 2009, she studied at East China Jiaotong University and obtained a master's degree in 2009. Since 2009, she has been working in Zhengzhou Technology and Business University. 12 papers have been published, 5 of which are in Chinese core journals. Research interests include legal philosophy and society, marriage, and family law.