DOI: 10.53469/jtpes.2024.04(03).14

Cloud Computing for Large-Scale Resource Computation and Storage in Machine Learning

Yufu Wang[1,*], Qiaozhi Bao[2], Jiufan Wang[3], Guangze Su[4], Xiaonan Xu[5]

[1]Computer Science & Engineering,Santa Clara University, Santa Clara, CA, USA
[2]Statistics, North Carolina State University, NC, USA
[3]Independent Researcher, William & Mary, Williamsburge,VA, USA
[4]Infomation Science, Trine University, Phoenix, AZ, USA
[5]Independent Researcher, Northern Arizona University, Flagstaff, USA
Corresponding author, stephenwyf1@gmail.com

Abstract: ***With the rapid development of Internet technology, cloud computing technology has gradually entered people's lives. Cloud computing provides users with various IT resources (computing, storage, etc.) in data centers all over the world through the Internet. Currently, there are hundreds of thousands of servers in large-scale data centers, and effective management of resources in such large-scale data centers is a major problem in academia and industry. This article explores the importance and pervasive use of cloud computing and machine learning in today's technology landscape. As the global cloud computing market size and the application of machine learning technologies continue to grow, the demand for computing and storage resources is also increasing. This paper aims to solve the challenges of large-scale resource computing and storage requirements in machine learning, and puts forward a solution how to give full play to the advantages of cloud computing platform and combine machine learning algorithms and technologies. Through practical data and case studies, we highlight the application scenarios, advantages and challenges of cloud computing in machine learning, and look forward to the future development trend.***

Keywords: Cloud Computing; Machine Learning; Resource Computation; Storage

1. INTRODUCTION

Cloud computing and machine learning are increasingly important and widely used in today's technology field. According to statistics, the global cloud computing market has reached hundreds of billions of dollars and is expected to maintain rapid growth in the next few years. At the same time, the application of machine learning technology is also showing an explosive growth trend, according to the International Data Corporation (IDC) report shows that the global machine learning market is expected to reach about 1.6 trillion US dollars in 2025. These figures fully illustrate the important role that cloud computing and machine learning play in driving digital transformation, innovative applications and industrial upgrading.

As the volume of data continues to grow and machine learning algorithms continue to evolve, so does the demand for computing and storage resources. According to the International Data Corporation (IDC), global data storage is growing at a rate of 30% per year and is expected to reach 175 zettabytes (1ZB= 100 million terabytes) by 2025. At the same time, according to PwC survey data, more than 85% of enterprises have or are exploring the application of AI technology, with machine learning being one of the most concerned areas. These data demonstrate the urgency and importance of the need for large-scale resource computing and storage.

This article will explore how to take full advantage of the cloud computing platform and combine machine learning algorithms and techniques to solve the challenges of large-scale resource computing and storage. Through actual data and case analysis, it will focus on the application scenarios, advantages and challenges of cloud computing in machine learning, and propose corresponding solutions and future development trends.

2. RELATED WORK

2.1 Cloud computing and machine Learning literature

Liu et al. 's research demonstrates the great potential of machine learning in materials science, especially in predicting the dielectric constant of spinel microwave dielectric ceramics. However, as materials science research continues to evolve, so does the need to model and predict more complex systems. These complexities often mean larger amounts of data and more complex models, and therefore require powerful computing resources to support them. Traditional computing methods may be limited by computing power and storage space, and it is difficult to deal with large-scale data and complex models effectively. In this context, cloud computing technology provides important support and solutions for materials scientists. By leveraging cloud computing platforms, researchers can easily access large-scale computing resources and storage space, enabling efficient modeling and prediction of complex material systems. The flexibility and scalability of the cloud computing platform allows it to adapt to tasks of different sizes and complexity, providing a flexible working environment for materials scientists.

In combination with machine learning and cloud computing technologies, researchers can leverage large-scale data sets and advanced models to improve the accuracy and efficiency of predicting material properties. By deploying machine learning models on cloud computing platforms, researchers can automate model training and deployment, accelerating the process of material design and engineering optimization. In addition, cloud computing platforms can provide a wealth of data analysis and visualization tools to help researchers better understand the relationship between the structure and properties of materials.

Therefore, the combination of cloud computing and machine learning technology can not only make up for the limitations of traditional computing methods, but also bring new ideas and methods to the development of materials science. This integrated approach provides more efficient and precise solutions for material design and engineering optimization, driving continuous progress and innovation in materials science.

2.2 The state of cloud computing and machine learning

Choudhury et al. 's survey not only delves into the potential and limitations of machine learning in solving the computing offload problem in edge computing, but also examines the role and impact of cloud computing. Their research reveals the importance of cloud computing in edge computing systems, especially its role in handling compute offloads and resource management.

Through an analysis of existing technologies, Choudhury et al. point out the challenges machine learning faces in edge computing, such as data privacy and security issues. At the same time, they also explore how cloud computing platforms can be used to solve these challenges, and propose some innovative solutions and future research directions. The flexibility and elasticity of cloud computing platforms enable them to support the performance optimization of edge computing systems and provide them with the necessary computing and storage resources.

Therefore, the research of Choudhury et al. provides an important reference for the design and optimization of edge computing systems, and also highlights the challenges and opportunities of cloud computing in solving edge computing. By combining cloud computing and machine learning technologies, we can better understand and leverage the potential of edge computing systems, providing guidance and support for their performance.

Therefore, these studies provide valuable experience and enlightenment for the combination of cloud computing and machine learning, and point the way for future research and application. In the field of materials science, the application of machine learning has demonstrated its potential in material design, performance prediction, and engineering optimization. By combining cloud computing technologies, researchers can use large-scale data and complex models more efficiently, accelerating the process of materials research. Future research could further explore how to further leverage the role of machine learning in materials science to address the growing demand and complexity of materials design. At the same time, in the field of edge computing, the combination of machine learning also shows great potential, especially in solving computing offload and resource management. Future research can continue to explore how to optimize the performance of edge computing systems and improve their efficiency and reliability. Through these efforts, we can better address the increasingly complex and diverse technological challenges and drive the further development and application of cloud computing and machine learning in the fields of materials science and edge computing.

2.3 Cloud computing large-scale storage

From the research of Gennaro Mellone et al., we can see the advantages of cloud computing in the partitioning of large-scale environmental data. By storing data in the cloud and on-premises storage, and taking advantage of the resiliency and scalability of cloud computing platforms, researchers are able to process and analyze large-scale environmental data more efficiently. Cloud computing provides a highly flexible storage solution that dynamically adjusts storage capacity according to demand, and enables high data availability and fault tolerance to ensure data security and reliability.

At the same time, Hou YaXin et al. demonstrated the potential of optical synapses in neuromorphic computation and information processing. Combined with machine learning techniques, this large-scale and flexible optical synapse can be used to enable efficient neuromorphic computing and integrated perceptive storage processing of visible information. By leveraging the storage and computing power of cloud computing platforms, combined with the advantages of optical synapses, it is possible to achieve faster, more flexible, and more efficient large-scale data processing and analysis. Machine learning technology can further optimize the data processing process and improve the accuracy and efficiency of data analysis, thus bringing more advances and advantages to environmental monitoring, resource management and other application fields.

Therefore, the combination of cloud computing and machine learning technologies can bring important advances and advantages to large-scale environmental data processing and analysis. This combination can improve the speed, flexibility and accuracy of data processing, leading to more innovation and development opportunities in future applications such as environmental monitoring and resource management.

2.4 Limitations of machine learning

According to Anwar Sajid et al., the biggest advantage of machine learning for large-scale storage is its ability to enable intelligent management and optimization of data. Through the application of machine learning algorithms, large-scale and heterogeneous repositories can be analyzed and learned, thereby improving the efficiency of data retrieval, storage and management. Machine learning can automatically discover correlations and patterns between data, and optimize and adjust storage systems based on this information to improve data access speed and resource utilization.

However, the current challenges mainly include the complexity and heterogeneity of data. Large-scale storage systems often contain various types and formats of data, which may have different structures and characteristics, making data analysis and management more complex and difficult. In addition, the amount of data in storage systems is often very large, which can cause traditional machine learning algorithms to encounter performance and efficiency limitations in processing and analysis.

According to David M. Eyers et al., one way to address these challenges is to use middleware in combination with machine learning. Through the introduction of middleware, the machine learning algorithm can be effectively integrated with the storage system, and the automatic management and optimization of the storage system can be realized. Middleware can provide data abstraction and interfaces that make it easier for machine learning algorithms to access and analyze data in storage systems, thereby improving the efficiency of data management and storage. Although machine learning has important advantages for large-scale storage, there are still challenges to overcome such as the complexity and heterogeneity of data. The combination of middleware and machine learning technology can effectively solve these challenges and achieve intelligent management and optimization of large-scale storage systems.

3. METHODOLOGY

In a typical distributed file system, metadata services play a crucial role because directory file metadata operations account for a large proportion of the overall file system operations. With the popularity of applications such as large-scale machine learning, big data analytics, and enterprise-class data lakes, the data scale of distributed file systems has expanded from the petabyte level to the EB level. At present, many distributed file systems (such as HDFS) face the challenge of metadata scalability.

Some leading tech companies, such as Google, Facebook, and Microsoft, have basically implemented distributed file systems capable of managing exabytes of data. A common architectural feature of these systems is that they rely on the underlying distributed database capability to achieve horizontal scaling of metadata performance. For example, Google's Colossus is based on BigTable, Facebook is based on ZippyDB, and Microsoft's ADLSv2 is based on Table Storage. In addition, some open source file systems, such as CephFS and HopsFS, also basically implement the ability to scale horizontally.

In summary, the impact of machine learning algorithms in cloud computing on storage system performance needs to take into account the metadata management and operation of file systems. For large-scale distributed file systems, the application of machine learning algorithms can help optimize the performance of metadata services and improve the scalability and efficiency of file systems. However, attention needs to be paid to the degree to which different file systems support machine learning algorithms, as well as semantic differences that may result from the limitations of the underlying distributed database.

3.1 Metadata evolution

DanceNN is a directory tree information storage system developed by the company, which is committed to solving the directory tree requirements of all distributed storage systems (including but not limited to HDFS, NAS, etc.), greatly simplifying the complexity of directory tree operations relied on by upper-layer storage systems, including but not limited to atomic Rename, recursive deletion, etc. Solve problems such as scalability, performance, and global unified namespace between heterogeneous systems in hyperscale directory tree storage scenarios, and build the world's leading universal distributed directory tree service. Currently, DanceNN provides directory tree metadata services for the online ByteNAS and offline HDFS distributed file systems.

NameNode

Initially, the company used HDFS native NameNode, and despite a lot of optimization, it still faced the following problems:

- Metadata (including directory trees, files, and Block copies) is stored in full memory, and the carrying capacity of a single machine is limited
- Based on Java language implementation, GC pauses for a long time in large memory scenarios, which seriously affects SLA
- When a global read/write lock is used, the read/write performance is poor

3.2 Data collection and preprocessing

Below is a simple example of a raw data table for collecting data from different distributed storage systems, including information such as metadata and operation logs. The data records the creation and modification of files and directories, as well as the corresponding time and storage system. By cleaning, transforming, and standardizing this data, it can provide a basis for subsequent analysis and modeling.

Table 1: Distributed Storage Operations Log

Storage System	File/Directory Name	File/Directory Type	Creation Time	Modification Time	File Size	Operation Type	Operation Time
HDFS	/data/file1	File	2023-01-01 10:15:00	2023-01-02 08:30:00	1024 MB	Create	2023-01-01 10:15:00
HDFS	/data/dir1	Directory	2023-01-01 10:20:00	2023-01-02 09:00:00	-	Create	2023-01-01 10:20:00
NAS	/docs/report1	File	2023-01-03 14:20:00	2023-01-04 11:45:00	512 MB	Create	2023-01-03 14:20:00
NAS	/docs/dir2	Directory	2023-01-03 14:30:00	2023-01-04 12:00:00	-	Create	2023-01-03 14:30:00
HDFS	/data/file1	File	2023-01-05 09:30:00	2023-01-05 10:45:00	2048 MB	Modify	2023-01-05 10:45:00
NAS	/docs/report1	File	2023-01-06 09:40:00	2023-01-06 12:15:00	768 MB	Modify	

The table includes fields such as:

- Storage System: Indicates the distributed storage system from which the data originates (e.g., HDFS or NAS).
- File/Directory Name: Path name of the file or directory.
- File/Directory Type: Indicates whether the record refers to a file or a directory.

- Creation Time: Timestamp indicating when the file or directory was created.
- Modification Time: Timestamp indicating the last modification time of the file or directory.
- File Size: Size of the file in megabytes (MB). This field is empty for directories.
- Operation Type: Type of operation performed on the file or directory (e.g., Create, Modify).
- Operation Time: Timestamp indicating when the operation was performed.

3.3 Feature engineering

Based on the collected data, we can design and extract relevant features that reflect various aspects of the storage system, including performance, workload, and metadata structure.

- File/Directory Count: The total count of files and directories in the storage system, indicating its size and complexity.
- Average File Size: The average size of files in the system, providing insights into storage utilization.
- File Modification Frequency: The frequency of file modifications over a certain period, indicating the system's activity level.
- Directory Depth: The maximum depth of directories in the system, reflecting its hierarchical structure.
- Storage System Type: A categorical feature indicating the type of storage system (e.g., HDFS, NAS).
- Time Since Last Modification: The duration since the last modification of any file or directory, indicating system stability.
- Operation Type Distribution: The distribution of different operation types (e.g., create, modify) in the system, providing insights into usage patterns.
- Creation-to-Modification Time Ratio: The ratio of the time taken between creation and the last modification of files, indicating how quickly files are modified after creation.
- File Size Distribution: The distribution of file sizes in the system, providing insights into data distribution and potential hotspots.
- Storage System Load: A derived feature indicating the overall load on the storage system, calculated based on factors such as file count, size, and modification frequency.

These features capture various aspects of the storage system's performance, workload, and metadata structure, which can be used to optimize model performance and make informed decisions about system management and resource allocation.

3.4 Model fundamentals

1) Random sampling: In a random forest, the training samples of each decision tree are obtained by random sampling. Random sampling refers to taking some samples from the original training set to form a new training set. The purpose of this is to make the training samples of each decision tree slightly different, increasing the diversity between decision trees.

2) Random feature selection: On each node of the decision tree, the random forest algorithm will randomly select a part of the features from all the features for segmentation. The goal is to increase the diversity between each decision tree and prevent certain features from becoming too dominant in the decision making process of the entire random forest.

3) Decision tree construction: Use randomly sampled data and randomly selected features to build multiple decision trees. In the construction process of decision tree, the usual decision tree algorithm (such as ID3, CART, etc.) is used.

4) Prediction of random forest: When a new sample is input into the random forest, it will go through the prediction process of each decision tree, and finally get the final prediction result according to the way of decision integration. For classification problems, the most common integration method is to adopt majority voting, that is, to vote according to the classification results of each decision tree, and select the category with the most votes as the final prediction result. For regression problem, we can adopt the method of average prediction, that is, average the predicted value of each decision tree as the final prediction result.

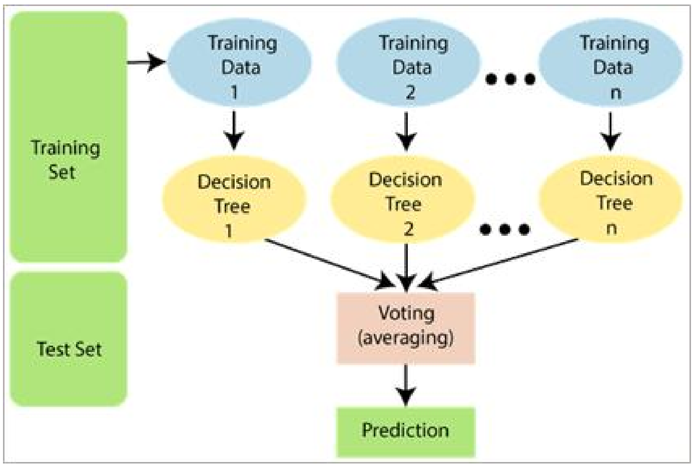

Figure 1. Random forest model

Through random sampling and random feature selection, the random forest algorithm can reduce the overfitting risk and improve the generalization ability of the model. At the same time, by integrating the prediction results of multiple decision trees, the random forest can obtain more stable and accurate predictions.

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

3.5 Layered Architecture

The latest HDFS distributed file system adopts a hierarchical architecture, which mainly includes three layers:
Data layer: Used to store file contents and handle block-level IO requests
Datanodes provide services
The Namespace layer: handles metadata related to directory trees and requests for creating, deleting, Rename, and authenticating directories and files
Provided by Distributed DanceNN Cluster
File Block layer: responsible for file related metadata, file and Block mapping and Block copy location information, processing file creation and deletion, file block addition and other requests
A BSGroup manages the metadata of some file blocks in a cluster and consists of multiple DanceBS to provide high availability services

You can dynamically expand the capacity of a BSGroup to adapt to the cluster load. When a BSGroup reaches its performance limit, write data can be controlled.

3.6 Storage Format

There are two general metadata formats based on distributed storage:

Solution 1 is similar to Google Colossus, which uses full path as key and metadata as value storage, and has the following advantages:

path resolution is very efficient, reading the metadata of the corresponding inode directly from the underlying KV storage through the path requested by the user

The scan directory can be scanned for KV storage by prefix.

3.7 Conclusion analysis

Based on the experiment described, the conclusion and analysis of implementing large-scale storage optimization based on machine learning can be summarized as follows:

1. Effectiveness of DanceNN: The introduction of DanceNN as a universal distributed directory tree service appears promising in addressing scalability, performance, and global namespace challenges across heterogeneous storage systems. The ability to simplify directory tree operations and provide metadata services for both online and offline distributed file systems like ByteNAS and HDFS showcases its potential impact on improving storage system efficiency.

2. Challenges with Traditional NameNode: The limitations of traditional HDFS NameNode, such as memory constraints, Java-based implementation leading to garbage collection issues, and poor read/write performance due to global locks, underscore the need for innovative solutions like DanceNN to overcome these shortcomings.

3. Data Collection and Preprocessing: The presented data collection and preprocessing methods offer a structured approach to extract valuable insights from distributed storage operation logs. By standardizing and transforming raw data into meaningful features, it becomes possible to analyze various aspects of storage system performance and workload.

4. Feature Engineering: The identified features capture essential characteristics of storage systems, including file and directory counts, average file size, modification frequency, and storage system load. These features provide a foundation for optimizing model performance and making informed decisions about resource allocation and system management.

5. Model Fundamentals: The explanation of random forest model fundamentals highlights its ability to address overfitting risks, improve generalization, and provide stable and accurate predictions. By leveraging random sampling, feature selection, and ensemble learning techniques, random forest models can effectively optimize storage system performance.

6. Layered Architecture and Storage Format: The discussion on the layered architecture of the HDFS distributed file system and different metadata storage formats provides insights into the underlying infrastructure supporting large-scale storage systems. Understanding the architecture and storage formats is crucial for designing and implementing optimization strategies.

7. Conclusion: In conclusion, the integration of machine learning algorithms in large-scale storage optimization shows promise in addressing challenges related to metadata scalability and improving the efficiency of distributed file systems. By leveraging technologies like DanceNN and random forest models, organizations can enhance system performance, scalability, and overall management of massive data sets.

8. Future Directions: Future research could focus on further refining the DanceNN system, exploring additional machine learning algorithms for storage optimization, and investigating the impact of different architectural designs on system performance. Additionally, evaluating the scalability and effectiveness of these approaches in real-world deployments would be essential for practical implementation.

4. Conclusion

In summary, the integration of machine learning techniques with large-scale storage optimization presents a promising solution to the challenges posed by metadata scalability and the efficient management of distributed file systems. By leveraging innovations such as DanceNN for universal directory tree services and employing random forest models for performance optimization, organizations can effectively enhance system efficiency, scalability, and overall resource management in the face of massive data sets. This approach aligns with the pervasive use of cloud computing and machine learning in addressing the growing demand for computing and storage resources, paving the way for further advancements in digital transformation and industrial innovation.

In conclusion, the synergy between machine learning and large-scale storage optimization not only addresses the pressing need for efficient resource management in cloud computing but also underscores the importance of innovative solutions in driving digital transformation. By harnessing the capabilities of technologies like DanceNN and random forest models, organizations can effectively navigate the complexities of metadata scalability and system performance, thus unlocking new opportunities for innovation and growth in the era of big data and machine learning.

ACKNOWLEDGEMENT

We are grateful to Huo, Shuning et al for the deep learning approach proposed in their research, which provides important inspiration and reference for our work. They are in Deep Learning Approaches for Improving Question Answering Systems in Hepatocellular Carcinoma In the paper "Research", the question and answer system in the study of hepatocellular carcinoma is deeply discussed and analyzed, which provides valuable theoretical basis and methodology support for our research.

In particular, I would like to thank the first authors of the study for their hard work and unique insights. Their work provides valuable references for our research and provides important ideas and directions for our research.
In addition, we would also like to thank all the institutions and individuals who support our work, their support and help to provide important resources and conditions for our research. Thank you for your support and encouragement, so that we can successfully complete this research work..

5. REFERENCES

[1] Wan, Weixiang, et al. "Progress in artificial intelligence applications based on the combination of self-driven sensors and deep learning." arXiv preprint arXiv:2402.09442 (2024).
[2] Wang, Yong, et al. "Construction and application of artificial intelligence crowdsourcing map based on multi-track GPS data." arXiv preprint arXiv:2402.15796 (2024).
[3] Zheng, Jiajian, et al. "The Random Forest Model for Analyzing and Forecasting the US Stock Market in the Context of Smart Finance." arXiv preprint arXiv:2402.17194 (2024).
[4] Yang, Le, et al. "AI-Driven Anonymization: Protecting Personal Data Privacy While Leveraging Machine Learning." arXiv preprint arXiv:2402.17191 (2024).
[5] Cheng, Qishuo, et al. "Optimizing Portfolio Management and Risk Assessment in Digital Assets Using Deep Learning for Predictive Analysis." arXiv preprint arXiv:2402.15994 (2024).
[6] Zhu, Mengran, et al. "Utilizing GANs for Fraud Detection: Model Training with Synthetic Transaction Data." arXiv preprint arXiv:2402.09830 (2024).
[7] Wu, Jiang, et al. "Data Pipeline Training: Integrating AutoML to Optimize the Data Flow of Machine Learning Models." arXiv preprint arXiv:2402.12916 (2024).
[8] Yu, Hanyi, et al. "Machine Learning-Based Vehicle Intention Trajectory Recognition and Prediction for Autonomous Driving." arXiv preprint arXiv:2402.16036 (2024).
[9] Huo, Shuning, et al. "Deep Learning Approaches for Improving Question Answering Systems in Hepatocellular Carcinoma Research." arXiv preprint arXiv:2402.16038 (2024).
[10] Chen, Jianhang, et al. "One-stage object referring with gaze estimation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[11] Duan, Shiheng, et al. “Prediction of Atmospheric Carbon Dioxide Radiative Transfer Model Based on Machine Learning”. Frontiers in Computing and Intelligent Systems, vol. 6, no. 3, Jan. 2024, pp. 132-6, https://doi.org/10.54097/ObMPjw5n.
[12] Chen , Jianfeng, et al. “Implementation of an AI-Based MRD Evaluation and Prediction Model for Multiple Myeloma”. Frontiers in Computing and Intelligent Systems, vol. 6, no. 3, Jan. 2024, pp. 127-31, https://doi.org/10.54097/zJ4MnbWW.
[13] “Machine Learning Model Training and Practice: A Study on Constructing a Novel Drug Detection System”. International Journal of Computer Science and Information Technology, vol. 1, no. 1, Dec. 2023, pp. 139-46, https://doi.org/10.62051/ijcsit.v1n1.19.
[14] Duan, Shiheng, et al. "THE INNOVATIVE MODEL OF ARTIFICIAL INTELLIGENCE COMPUTER EDUCATION UNDER THE BACKGROUND OF EDUCATIONAL INNOVATION." The 2nd International scientific and practical conference “Innovations in education: prospects and challenges of today”(January 16-19, 2024) Sofia, Bulgaria. International Science Group. 2024. 389 p.. 2024.
[15] Shen, Zepeng, et al. "EDUCATIONAL INNOVATION IN THE DIGITAL AGE: THE ROLE AND IMPACT OF NLP TECHNOLOGY." OLD AND NEW TECHNOLOGIES OF LEARNING DEVELOPMENT IN MODERN CONDITIONS (2024): 281.
[16] Gong, Yulu, et al. "RESEARCH ON A MULTILEVEL PRACTICAL TEACHING SYSTEM FOR THE COURSE'DIGITAL IMAGE PROCESSING." OLD AND NEW TECHNOLOGIES OF LEARNING DEVELOPMENT IN MODERN CONDITIONS (2024): 272.

[17] Qian, Wenpin, et al. "NEXT-GENERATION ARTIFICIAL INTELLIGENCE INNOVATIVE APPLICATIONS OF LARGE LANGUAGE MODELS AND NEW METHODS." OLD AND NEW TECHNOLOGIES OF LEARNING DEVELOPMENT IN MODERN CONDITIONS (2024): 262.

[18] W. Sun, W. Wan, L. Pan, J. Xu, and Q. Zeng, “The Integration of Large-Scale Language Models Into Intelligent Adjudication: Justification Rules and Implementation Pathways”, Journal of Industrial Engineering & Applied Science, vol. 2, no. 1, pp. 13–20, Feb. 2024.

[19] Zhou, Yanlin, et al. "Utilizing AI-Enhanced Multi-Omics Integration for Predictive Modeling of Disease Susceptibility in Functional Phenotypes." Journal of Theory and Practice of Engineering Science 4.02 (2024): 45-51.

[20] Liang, Penghao, et al. "Enhancing Security in DevOps by Integrating Artificial Intelligence and Machine Learning." Journal of Theory and Practice of Engineering Science 4.02 (2024): 31-37.

[21] Zhang, Chenwei, et al. "SegNet Network Architecture for Deep Learning Image Segmentation and Its Integrated Applications and Prospects." Academic Journal of Science and Technology 9.2 (2024): 224-229.

[22] Wang, Yong, et al. "Autonomous Driving System Driven by Artificial Intelligence Perception Fusion." Academic Journal of Science and Technology 9.2 (2024): 193-198.

[23] Zhang, Quan, et al. "Application of the AlphaFold2 Protein Prediction Algorithm Based on Artificial Intelligence." Journal of Theory and Practice of Engineering Science 4.02 (2024): 58-65.

[24] Wei, Kuo, et al. “Strategic Application of AI Intelligent Algorithm in Network Threat Detection and Defense”. Journal of Theory and Practice of Engineering Science, vol. 4, no. 01, Jan. 2024, pp. 49-57, doi:10.53469/jtpes.2024.04(01).07.

[25] Su, Jing, et al. "Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature R eview." arXiv preprint arXiv:2402.10350 (2024).

[26] Wang, Yong, et al. "Construction and application of artificial intelligence crowdsourcing map based on mul ti-track GPS data." arXiv preprint arXiv:2402.15796 (2024).

[27] K. Tan and W. Li, "Imaging and Parameter Estimating for Fast Moving Targets in Airborne SAR," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 126-140, March 2017, doi: 10.1109/TCI.2016.2634421.

[28] K. Tan and W. Li, "A novel moving parameter estimation approach offast moving targets based on phase extraction," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 2015, pp. 2075-2079, doi: 10.1109/ICIP.2015.7351166.

[29] Tan, Kai, et al. “Integrating Advanced Computer Vision and AI Algorithms for Autonomous Driving Systems”. Journal of Theory and Practice of Engineering Science, vol. 4, no. 01, Jan. 2024, pp. 41-48, doi:10.53469/jtpes.2024.04(01).06.

[30] H. Zhu and B. Wang, "Negative Siamese Network for Classifying Semantically Similar Sentences," 2021 International Conference on Asian Language Processing (IALP), Singapore, Singapore, 2021, pp. 170-173, doi: 10.1109/IALP54817.2021.9675278.