Article

Research on Data Service Platforms for Large Language Model Applications

Michael J. Reynolds *

Posted Date: 27 May 2025

doi: 10.20944/preprints202505.2070.v1

Keywords: large language models; data service platforms; data storage and management; real-time and scalability

Article

Research on Data Service Platforms for Large Language Model Applications

Michael J. Reynolds

University of Michigan, USA; mjreynolds0203@gmail.com

Abstract: With the widespread application of large language models (LLMs) such as the GPT series and BERT in Natural Language Processing (NLP), constructing efficient data service platforms to support their large-scale applications and optimize model performance has become a hot research topic. This paper aims to investigate the design and optimization of data service platforms for LLM applications, exploring the challenges and solutions in platform architecture, data storage management, and data processing optimization. First, the data requirements for LLMs, including data volume, quality, real-time processing, and scalability, are analyzed, and the application of multimodal data fusion and distributed data processing technologies is proposed. Secondly, a modular data service platform architecture is designed, addressing issues such as storage, management, interfaces, and API design. Experimental results demonstrate that optimizing data processing workflows and platform architecture significantly enhances LLM training and inference efficiency, driving the implementation of large-scale data-driven AI applications. Finally, the paper discusses the technical challenges in platform design and offers further optimization suggestions, providing theoretical support and practical guidance for the widespread application of LLMs.

Keywords: large language models; data service platforms; data storage and management; real-time and scalability

1. Introduction

In recent years, large language models (LLMs) such as GPT series and BERT have made significant breakthroughs in Natural Language Processing (NLP), with wide applications in machine translation, intelligent Q\&A, automatic text generation, and speech recognition. These models rely on massive datasets for training, enabling them to understand and generate language. As their application scenarios expand, efficiently supporting the storage, processing, and transmission of large-scale data has become a key challenge for LLM applications.Currently, while much research has focused on LLMs and their applications, there is a lack of systematic research on efficient data service platforms specifically designed for these models. Traditional data service platforms typically focus on data storage and processing, but they struggle to meet the real-time, scalability, multimodal fusion, and high-speed processing needs of LLM training and inference. To address these needs, a data service platform must possess an efficient architecture, preprocessing capabilities, and support for distributed computing and parallel processing to ensure optimal operation in complex applications.This paper aims to investigate the design and implementation of data service platforms for LLM applications. It explores how optimizing platform architecture and data processing workflows can support efficient training and inference of LLMs. Additionally, this study analyzes the data requirements for LLMs, focusing on aspects such as data volume, quality, real-time processing, and scalability. The proposed platform design is based on real-world experiments and case studies, providing practical guidance for advancing LLM applications in various fields such as intelligent customer service, content generation, and recommendation systems [1–3].

2. Literature Review

2.1. The Current Status of LLM Applications

Large language models (LLMs) like GPT and BERT have accelerated the growth of artificial intelligence, excelling in understanding and generating high-quality text. These models are widely used across various fields such as intelligent customer service, machine translation, text generation, sentiment analysis, and speech recognition.In intelligent customer service, LLMs enable automated support systems that improve efficiency and reduce labor costs, handling multi-turn dialogues and understanding user needs[4]. However, challenges remain in processing complex dialogues and understanding context. In machine translation, LLMs have surpassed traditional methods by using deep learning to provide multilingual translations, though issues like translation quality for low-resource languages persist.LLMs also play a significant role in text generation, enhancing content creation in areas like news writing, advertising, and literary work[5]. Models like GPT-3 and GPT-4 can generate long-form text from short prompts, boosting productivity. However, issues such as originality, plagiarism, and ethical concerns in generated content are still challenges.Sentiment analysis is another key application, helping businesses analyze social media and customer reviews to understand public sentiment. While this helps in real-time decision-making, understanding sarcasm and complex emotional expressions remains a challenge. In speech recognition, LLM-based systems, such as voice assistants, offer natural human-computer interaction but face difficulties with dialects and noisy environments.Overall, LLMs have transformed several industries, but challenges like data bias, ethical issues, and accuracy in specialized fields remain. With ongoing advancements, LLMs are expected to further optimize and expand their applications[6].

2.2. Related Research on Data Service Platforms

As the application of LLMs expands across various fields, the design and implementation of data service platforms have become increasingly important. These platforms not only handle data storage but also manage tasks such as data collection, cleaning, preprocessing, storage, transmission, and presentation. In the context of LLM applications, data service platforms must possess efficient processing capabilities and scalability to handle massive amounts of data while supporting model training, inference, and other tasks. Current research mainly focuses on platform architecture design, data processing and computation optimization, distributed storage and management, and real-time data processing.The architecture design of data service platforms is fundamental to efficient data processing and LLM support. Many studies adopt microservice architectures to construct data service platforms, decoupling different functional modules into services, allowing platforms to achieve greater flexibility and scalability. By doing so, platforms can decouple tasks like data processing, storage, transmission, and model inference, avoiding bottlenecks in a single system[7–10]. Furthermore, integrating cloud computing and edge computing architectures can enhance platform computing and storage capabilities while reducing data transmission delays, improving real-time performance and response speed. These architectural designs enable LLMs to operate more efficiently in complex application scenarios.Data storage and management is another critical aspect of data service platforms for LLMs. Traditional relational databases cannot meet the storage needs of massive data, so distributed databases and big data storage systems have become the focus of research. Many data service platforms, based on distributed computing frameworks like Hadoop and Spark, can efficiently store and process massive data[10,11]. Meanwhile, NoSQL databases such as Cassandra and MongoDB, with their excellent scalability and efficient read/write performance, are widely used for handling various types of unstructured data required by LLMs. Additionally, research in data management focuses on data cleaning, deduplication, and standardization to improve data quality and provide reliable inputs for LLM training.In terms of computational optimization, since LLM training and inference require substantial computational resources, researchers have proposed several methods to optimize the computation process. Distributed and parallel computing technologies are widely applied to process large-scale data, with tasks distributed

doi:10.20944/preprints202505.2070.v1

across multiple computing nodes for parallel execution, greatly improving computational efficiency. Furthermore, using hardware acceleration technologies like GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) can significantly increase the training speed of deep learning models, reducing computation time and promoting the widespread deployment of LLMs in real-world applications[11–13]. Real-time data processing and stream processing technologies are also crucial for data service platforms in LLMs. To meet the real-time data processing demands of LLMs, many platforms use stream processing frameworks such as Apache Kafka and Apache Flink, which enable low-latency processing of data from various sources. These technologies allow LLMs to perform real-time inference and respond instantly, making them widely applicable in speech recognition, real-time translation, and intelligent Q&A tasks. Additionally, the data pipelines and message queue mechanisms provided by stream processing platforms ensure the real-time transmission and processing of data, offering robust support for real-time LLM applications.In terms of data security and privacy protection, as LLMs are applied across various industries, safeguarding user data privacy and security has become an increasingly important issue[14]. Many studies have proposed techniques such as data encryption, access control, and data anonymization to ensure the security of data during storage and transmission. In industries dealing with sensitive data, such as healthcare and finance, data security and privacy protection are central to platform design. Moreover, the introduction of blockchain technology offers new approaches to data security, providing decentralized data management and immutable data records, which strengthen the trustworthiness of data service platforms[15]. In summary, the importance of data service platforms in LLM applications is growing, with ongoing research and technological advancements. However, existing platforms still face challenges, such as handling growing data volumes, improving computational efficiency, and overcoming bottlenecks in real-time data processing. With continuous technological progress, data service platforms will play an increasingly important role in enhancing LLM performance and application effectiveness, providing a solid foundation for the widespread adoption of artificial intelligence[16].

2.3. Technological Development Trends

As large language models (LLMs) are applied across multiple fields, the technology of data service platforms is continually evolving, with future development focusing on enhancing platform computing power, data processing efficiency, real-time capabilities, and security to meet the demands of large-scale model applications.Distributed computing and storage technologies will remain at the core of data service platforms. As LLMs' computational needs and data scales continue to grow, single servers can no longer meet these requirements. Therefore, optimizing distributed computing platforms such as Hadoop and Spark will be a key focus for future development. Additionally, the integration of cloud computing and edge computing will become a trend. Cloud platforms provide powerful computing capabilities, while edge computing can process data closer to the source, reducing latency and enhancing real-time performance[17–19]. This collaborative architecture will support the efficient operation of LLMs.Next, multimodal data processing will be an important direction for future platform development. LLMs need to handle not only text data but also image, audio, video, and other multimodal data. With advancements in deep learning and multimodal technologies, platforms will be able to better integrate different types of data, thereby enhancing model performance and expanding application ranges. For example, combining voice and text, image and text multimodal data can provide richer information for LLMs.Moreover, real-time data processing technologies will further enhance platform response speed and processing capabilities. Stream processing frameworks like Apache Kafka and Apache Flink will be more widely used to handle real-time data streams. This is especially crucial for applications that require low latency, such as speech recognition and real-time translation. Future platforms will focus more on reducing data processing delays[20]. Hardware acceleration technologies will play a key role in future applications. With the widespread adoption of GPUs and TPUs, platforms will be able to accelerate the training and inference processes of LLMs, significantly improving computational efficiency. Additionally, data privacy and security will become a focus of future technologies, especially in industries dealing with sensitive data (such as healthcare and finance). Data encryption, access control, and decentralized storage technologies will ensure data security.In summary, future data service platforms will continue to innovate in distributed computing, multimodal data processing, real-time data processing, hardware acceleration, and data security, providing strong technical support for the widespread application of LLMs[21–23].

3. Data Requirements Analysis for Large Language Models

The performance of Large Language Models (LLMs) heavily relies on the quantity and quality of data. To effectively train and apply LLMs, it is essential to meet their data requirements in terms of volume, quality, real-time processing, and scalability. This paper analyzes the data needs of LLMs in these aspects.LLM training depends on vast datasets to capture the diversity and complexity of language. The larger the dataset, the richer the language features the model can learn, thus improving its accuracy and generalization ability. Typically, training an efficient LLM requires hundreds of GB or even TB of data. For example, GPT-3's training dataset amounts to trillions of words. These massive datasets are usually sourced from various channels such as internet text, books, news articles, and dialogue data. Thus, the size of the data is crucial for enhancing the performance of LLMs[24]. Beyond data volume, the quality of data is also a key to the success of LLMs. High-quality data improves model training effectiveness, preventing the model from learning erroneous or inconsistent information[25–27]. The accuracy, completeness, and diversity of the data are critical standards for evaluating data quality. Data accuracy ensures that input data is free from errors or biases, especially important in specialized applications. Moreover, the diversity of data ensures that the model can learn various language styles, cultural backgrounds, and domain knowledge, enhancing its applicability in multiple scenarios.LLM training data often contains a significant amount of noise, such as duplicate data, spelling errors, grammatical mistakes, or irrelevant content. Therefore, data preprocessing and cleaning are essential. Common preprocessing steps include removing invalid data, filling missing values, standardizing text formats, and deduplication. Additionally, feature extraction and data augmentation are effective ways to improve data quality. Through proper data preprocessing, the input to the model can be cleaner and more useful, thus improving the model's training efficiency and prediction accuracy.With the continuous expansion of LLM application scenarios, especially in real-time speech recognition, real-time translation, and online customer service, the requirements for data real-time processing and scalability have become more demanding. Data service platforms need to support the real-time processing of high-concurrency data streams to ensure that models can respond quickly when handling a large number of requests. To achieve this, distributed computing architectures and stream processing technologies, such as Apache Kafka and Flink, can help the platform efficiently handle real-time data streams, ensuring the response speed and processing capability of LLMs in practical applications[28–30].

LLMs have very high data volume and quality requirements, and with the expansion of their application scenarios, real-time processing and scalability requirements are becoming even higher. To support the efficient training and application of LLMs, it is crucial to provide massive, high-quality data while incorporating real-time data processing and distributed computing technologies to ensure data processing efficiency and scalability[31].

4. Architecture Design of Data Service Platform

The application of Large Language Models (LLMs) requires an efficient, scalable, and flexible data service platform. This platform not only needs to support the storage and processing of large-scale data but also ensure real-time processing, data security, and scalability. Below is the core architecture design of an efficient data service platform[32]. Firstly, the data collection and input module is the foundation of the platform, responsible for acquiring data from various sources. These data sources include text, speech, images, and other multimodal data, as well as real-time data streams. To improve data processing efficiency, platforms typically use stream processing technologies such as Apache Kafka or Apache Flink, ensuring the real-time processing of high-frequency data[33]. The data storage and management module is responsible for the persistent storage of data. As data volume grows, platforms generally use distributed storage technologies such as HDFS or NoSQL databases (e.g., Cassandra, MongoDB) to handle large-scale data storage requirements. Additionally, to improve data access efficiency, the platform implements efficient data indexing mechanisms to ensure fast querying. In the data processing and computing module, the platform must possess strong computational capabilities. The training process for large language models typically requires high-performance distributed computing frameworks such as Apache Spark, TensorFlow, or PyTorch. Moreover, by utilizing hardware acceleration technologies like GPUs or TPUs, the platform can significantly improve training and inference efficiency[34]. The model training and inference module is the core part of the platform, responsible for training and inference of large language models. The platform needs to support distributed training and leverage large-scale computational resources to accelerate model training. The inference module must ensure efficient response speed, ensuring real-time processing of user requests. Data interfaces and API design are crucial for ensuring smooth integration with external systems. The platform typically provides RESTful APIs to support data exchange and functionality calls from external systems. At the same time, the platform should be able to handle the input and output of different modalities of data, ensuring efficient data flow transmission and processing. Finally, the security and privacy protection module is crucial for ensuring the safety of user data. The platform should adopt encryption, authentication, and access control technologies to protect data privacy and security, particularly when handling sensitive data, adhering to relevant regulations such as GDPR. In summary, the data service platform needs to provide powerful data collection, storage, processing, and inference capabilities, and support distributed computing, hardware acceleration, and security protection technologies to efficiently meet the demands of large language models in various applications.

5. Data Platform Optimization and Model Support

To efficiently support the application of Large Language Models (LLMs), data platforms not only need to provide robust data storage and processing capabilities but also require optimization at multiple levels to ensure the efficiency and accuracy of model training and inference. Below are key strategies for optimizing data platforms and supporting models.Firstly, data preprocessing and feature engineering are crucial for improving the performance of large language models[35]. Data preprocessing includes steps like data cleaning, denoising, standardization, and filling missing values to ensure the quality of input data for model training. Additionally, feature engineering, by extracting important features, enhances the model’s learning efficiency and prediction accuracy. For example, using word embeddings or TF-IDF for feature extraction in text data helps the model better understand the semantics of the text.Next, distributed data processing and computation acceleration are core to enhancing platform performance. Training large language models requires processing massive data, so using distributed computing frameworks like Apache Spark, TensorFlow, and PyTorch allows data processing tasks to be distributed across multiple computing nodes, enabling parallel computation and significantly improving processing speed. By leveraging hardware acceleration such as GPUs and TPUs, the platform can substantially reduce computation time during both training and inference, supporting larger models and datasets. Regarding model support, multi-task learning and transfer learning are essential for optimizing platform performance[36]. Multi-task learning allows the model to learn multiple related tasks simultaneously during the same training process, enhancing the model’s generalization ability and efficiency. Transfer learning, on the other hand, enables the transfer of knowledge from existing models to new tasks, reducing training time and data requirements, thus allowing large language models to adapt more quickly to new application scenarios. Furthermore, data flow optimization and real-time processing are important areas of platform optimization. With increasing demand for LLM applications, platforms need to support real-time data stream processing to ensure the model responds efficiently in real-time applications. For instance, using stream processing frameworks like Apache Kafka and Apache Flink helps the platform efficiently process real-time data from different sources and quickly pass it to the model for inference, meeting low-latency and high-throughput requirements. Finally, automated tuning and intelligent management are crucial directions for future platform optimization. By using automated tools for hyperparameter optimization, resource scheduling, and performance monitoring, the platform can automatically adjust computational resources based on task demands, ensuring efficient operation. In conclusion, optimizing data platforms and supporting models require a combination of data preprocessing, distributed computing, hardware acceleration, transfer learning, and other technologies to enhance platform performance and improve model application effectiveness. These optimization measures can significantly improve the training efficiency of large language models, supporting more complex and diverse application scenarios[37].

6. Experiment and Result Analysis

To evaluate the effectiveness of the data service platform in supporting Large Language Models (LLMs), we designed a series of experiments covering data processing, model training, inference response speed, and other aspects. The goal of the experiment is to validate the platform’s performance in handling large-scale data, assess the impact of various optimization strategies on training efficiency and model accuracy, and compare the results before and after platform optimization.The experiments are divided into two parts: data processing efficiency and model inference speed. The first part of the experiment measures data processing time as the evaluation metric, testing the platform's processing capability under different data volumes. The second part focuses on the inference response time, testing the platform's inference performance under high concurrency conditions.

As shown in the Figure 1, as the data volume increases, the time for data preprocessing is significantly reduced after adopting distributed processing, resulting in a substantial performance improvement. Through distributed computing frameworks and GPU acceleration, the platform effectively shortens the processing time for large data volumes, improving the data stream processing efficiency.From the experimental results, the platform shows a significant improvement in preprocessing time, particularly when dealing with data volumes above 100GB. The distributed processing approach, compared to traditional single-machine processing, improves efficiency by nearly 62%. This indicates that the platform’s design offers a clear advantage in data processing efficiency, providing reliable support for LLM training.

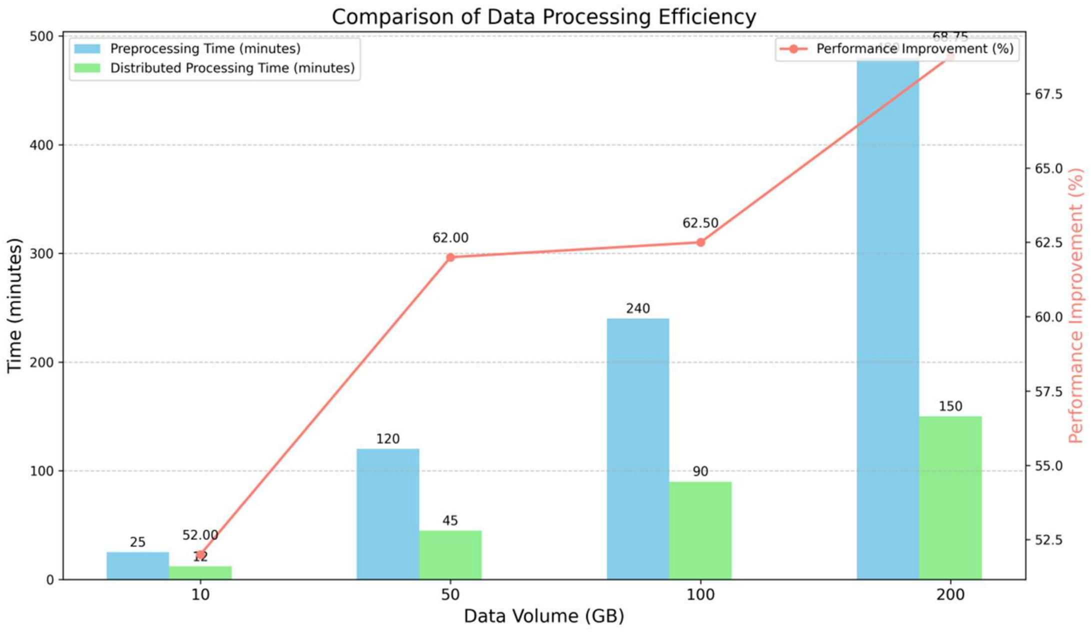

Figure 1. Comparison of Data Processing Efficiency.

In the inference response time experiment, the response time is significantly reduced after platform optimization. As shown in Figure 2, Particularly in scenarios with high request volumes, the optimized platform greatly enhances inference speed, reducing response time by over 60%. This indicates that the platform optimization, including distributed computing and hardware acceleration, effectively improves the concurrency processing capacity of the inference process, meeting the demands of large-scale requests.The experimental results demonstrate that the data service platform shows significant optimization effects in both data processing and inference performance. By employing distributed computing, hardware acceleration, and data flow optimization technologies, the platform effectively enhances the performance of LLMs in real-world applications, meeting the demands of high concurrency and high data volume environments. These optimizations provide strong support for the application of LLMs and offer practical insights for further development and improvement of the platform.

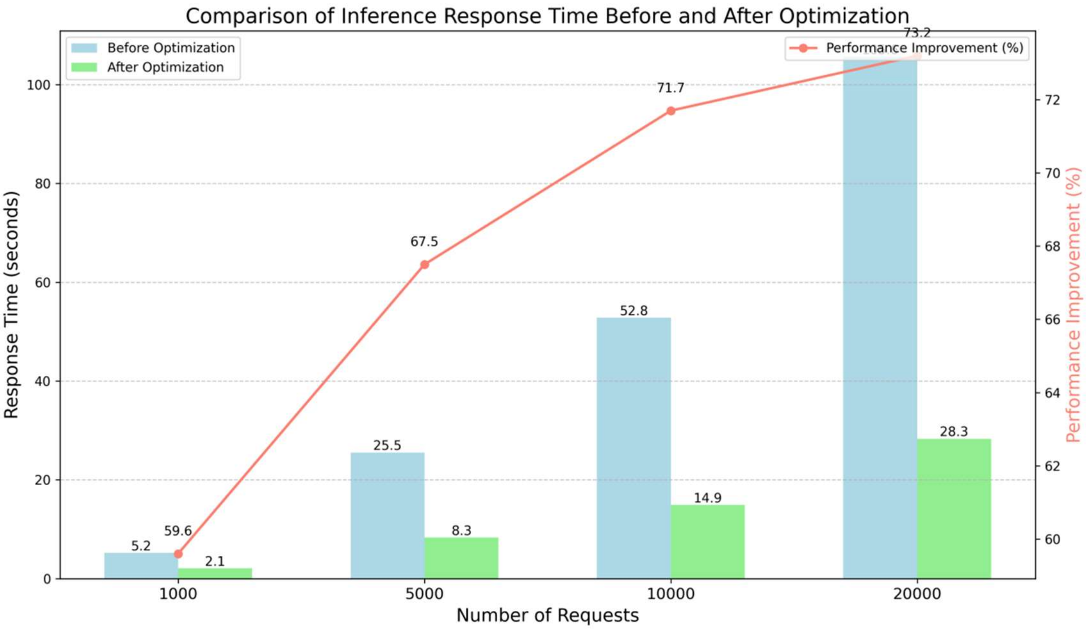

Figure 2. Comparison of Inference Response Time.

7. Conclusions

This study explored the design and optimization of data service platforms for LLM applications, presenting key technological solutions for platform architecture, data storage management, and data processing optimization. Through modular design, distributed computing, hardware acceleration, and real-time data stream processing optimizations, the platform can effectively enhance the training and inference efficiency of large language models, meeting the performance demands across various application scenarios.The experimental results show that after adopting distributed processing and hardware acceleration, the platform exhibits significant performance improvements in data preprocessing and inference response time. Especially in high data volume and high concurrency scenarios, the optimized platform significantly reduces processing time and improves response speed. These optimizations make the platform more efficient in real-world applications, particularly in tasks requiring real-time responses, such as intelligent customer service, speech recognition, and real-time translation.However, despite the significant progress made in data processing and inference performance, there is still room for further optimization. Future research could continue to explore how to enhance the platform's computational capabilities, optimize multimodal data fusion and processing, further reduce system latency, and strengthen data security and privacy protection to accommodate more complex and diverse application scenarios.In conclusion, the data service platform provides robust technical support for large language models, promoting their widespread deployment in real-world applications. With further optimization and technological advancements, future platforms will be better equipped to address the challenges of LLM applications, providing a more reliable infrastructure for intelligent services.

References

1. Zhao Y, Hu B, Wang S. Prediction of brent crude oil price based on lstm model under the background of low-carbon transition[J]. arXiv preprint arXiv:2409.12376, 2024.
2. Yang, Haowei, et al. "Optimization and Scalability of Collaborative Filtering Algorithms in Large Language Models." arXiv preprint arXiv:2412.18715 (2024).
3. Xiang A, Qi Z, Wang H, et al. A multimodal fusion network for student emotion recognition based on transformer and tensor product[C]//2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE). IEEE, 2024: 1-4.
4. Diao, Su, et al. "Ventilator pressure prediction using recurrent neural network." arXiv preprint arXiv:2410.06552 (2024).
5. Shi X, Tao Y, Lin S C. Deep Neural Network-Based Prediction of B-Cell Epitopes for SARS-CoV and SARS-CoV-2: Enhancing Vaccine Design through Machine Learning[J]. arXiv preprint arXiv:2412.00109, 2024.
6. Wang T, Cai X, Xu Q. Energy Market Price Forecasting and Financial Technology Risk Management Based on Generative AI[J]. Applied and Computational Engineering, 2024, 100: 29-34.
7. Yin Z, Hu B, Chen S. Predicting employee turnover in the financial company: A comparative study of catboost and xgboost models[J]. Applied and Computational Engineering, 2024, 100: 86-92.
8. Yin J, Wu X, Liu X. Multi-class classification of breast cancer gene expression using PCA and XGBoost[J]. Theoretical and Natural Science, 2025, 76: 6-11.
9. Gao, Dawei, et al. "Synaptic resistor circuits based on Al oxide and Ti silicide for concurrent learning and signal processing in artificial intelligence systems." Advanced Materials 35.15 (2023): 2210484.
10. Lv, Guangxin, et al. "Dynamic covalent bonds in vitrimers enable 1.0 W/(m K) intrinsic thermal conductivity." Macromolecules 56.4 (2023): 1554-1561.
11. Min, Liu, et al. "Financial Prediction Using DeepFM: Loan Repayment with Attention and Hybrid Loss." 2024 5th International Conference on Machine Learning and Computer Application (ICMLCA). IEEE, 2024.
12. Yu Q, Wang S, Tao Y. Enhancing anti-money laundering detection with self-attention graph neural networks[C]//SHS Web of Conferences. EDP Sciences, 2025, 213: 01016.
13. Shen, Jiajiang, Weiyan Wu, and Qianyu Xu. "Accurate prediction of temperature indicators in eastern china using a multi-scale cnn-lstm-attention model." arXiv preprint arXiv:2412.07997 (2024).
14. Huang B, Lu Q, Huang S, et al. Multi-modal clothing recommendation model based on large model and VAE enhancement[J]. arXiv preprint arXiv:2410.02219, 2024.
15. Wang, H., Zhang, G., Zhao, Y., Lai, F., Cui, W., Xue, J., ... & Lin, Y. (2024, December). Rpf-eld: Regional prior fusion using early and late distillation for breast cancer recognition in ultrasound images. In 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2605-2612). IEEE.
16. Mo K, Chu L, Zhang X, et al. Dral: Deep reinforcement adaptive learning for multi-uavs navigation in unknown indoor environment[J]. arXiv preprint arXiv:2409.03930, 2024.
17. Guo H, Zhang Y, Chen L, et al. Research on vehicle detection based on improved YOLOv8 network[J]. arXiv preprint arXiv:2501.00300, 2024.
18. Yang, H., Lu, Q., Wang, Y., Liu, S., Zheng, J., & Xiang, A. (2025). User Behavior Analysis in Privacy Protection with Large Language Models: A Study on Privacy Preferences with Limited Data. arXiv preprint arXiv:2505.06305.
19. Ziang H, Zhang J, Li L. Framework for lung CT image segmentation based on UNet++[J]. arXiv preprint arXiv:2501.02428, 2025.
20. Shih K, Han Y, Tan L. Recommendation System in Advertising and Streaming Media: Unsupervised Data Enhancement Sequence Suggestions[J]. arXiv preprint arXiv:2504.08740, 2025.

21. Ge, Ge, et al. "A review of the effect of the ketogenic diet on glycemic control in adults with type 2 diabetes." Precision Nutrition 4.1 (2025): e00100.
22. Wang, Junqiao, et al. "Enhancing Code LLMs with Reinforcement Learning in Code Generation." arXiv preprint arXiv:2412.20367 (2024).
23. Lin, Xueting, et al. "Enhanced Recommendation Combining Collaborative Filtering and Large Language Models." arXiv preprint arXiv:2412.18713 (2024).
24. Yi, Qiang, et al. "Score: Story coherence and retrieval enhancement for ai narratives." arXiv preprint arXiv:2503.23512 (2025).
25. Mao, Yiting, et al. "Research and Design on Intelligent Recognition of Unordered Targets for Robots Based on Reinforcement Learning." arXiv preprint arXiv:2503.07340 (2025).
26. Wu, Siye, et al. "Warehouse Robot Task Scheduling Based on Reinforcement Learning to Maximize Operational Efficiency." Authorea Preprints (2025).
27. Yang, Haowei, et al. "Research on the Design of a Short Video Recommendation System Based on Multimodal Information and Differential Privacy." arXiv preprint arXiv:2504.08751 (2025).
28. Yu, Dezhi, et al. "Machine learning optimizes the efficiency of picking and packing in automated warehouse robot systems." 2024 International Conference on Computer Engineering, Network and Digital Communication (CENDC 2024). 2024.
29. Tang, Xirui, et al. "Research on heterogeneous computation resource allocation based on data-driven method." 2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS). IEEE, 2024.
30. Tan, Chaoyi, et al. "Real-time Video Target Tracking Algorithm Utilizing Convolutional Neural Networks (CNN)." 2024 4th International Conference on Electronic Information Engineering and Computer (EIECT). IEEE, 2024.
31. Tan, Chaoyi, et al. "Generating Multimodal Images with GAN: Integrating Text, Image, and Style." arXiv preprint arXiv:2501.02167 (2025).
32. Li, Xiangtian, et al. "Artistic Neural Style Transfer Algorithms with Activation Smoothing." arXiv preprint arXiv:2411.08014 (2024).
33. Qi, Zhen, et al. "Detecting and Classifying Defective Products in Images Using YOLO." arXiv preprint arXiv:2412.16935 (2024).
34. Yang, Haowei, et al. "Analysis of Financial Risk Behavior Prediction Using Deep Learning and Big Data Algorithms." arXiv preprint arXiv:2410.19394 (2024).
35. Xiang A, Huang B, Guo X, et al. A neural matrix decomposition recommender system model based on the multimodal large language model[C]//Proceedings of the 2024 7th International Conference on Machine Learning and Machine Intelligence (MLMI). 2024: 146-150.
36. Xiang A, Zhang J, Yang Q, et al. Research on splicing image detection algorithms based on natural image statistical characteristics[J]. arXiv preprint arXiv:2404.16296, 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.