sciendo
Control and Cybernetics
vol. **53** (2024) No. 4
pages: 639-672
DOI: 10.2478/candc-2024-0025

Artificial intelligence-based smart cloud computing schema model*

by

Radhakrishnan Kanthavel[1], Ramakrishnan Dhaya[1*], Osamah Ibrahim Khalaf[2] and Abdulsattar Abdullah Hamad[3]

[1] School of Electrical and Communications Engineering, PNG University of Technology, Papua New Guinea
[2]Department of Solar, Al-Nahrain Research Center for Renewable Energy, Al-Nahrain University, Jadriya, Baghdad, Iraq
[3] University of Samarra, College of Education, Department of Physics, Iraq
*corresponding author: dhayavel2005@gmail.com

Abstract: In the contemporary digital era, cloud computing offers an ideal platform for artificial intelligence (AI) applications by providing the necessary computational power, memory, and scalability to handle the massive volumes of data required by intelligent algorithms. AI systems enable computing devices to make expert-level decisions by effectively leveraging information. However, challenges, related to adaptability, efficiency, privacy preservation, and the latent requirement for minimal user intervention remain critical. Notably, error detection efficiency can be improved by distributing data across multiple cloud storage services, akin to spreading data across physical disk drives. Nevertheless, continuously optimizing the performance and cost-efficiency of multiple cloud providers remains a complex task, due to varying pricing models and service quality levels. This paper aims to clarify how rule enforcement for distributed systems can be improved through the use of diverse cloud hosting services guided by authorization patterns. We propose an Effective AI Architecture for File Distribution Enhancement (EAIFDE), which aims to minimize costs and waiting times across various cloud platforms. The proposed architecture is validated using a cloud storage system simulator to evaluate the operational complexity and performance differences among multiple providers.

Keywords: artificial intelligence; cloud storage; predicted cost; authorization; efficacy; waiting time; efficiency and predictive analytics

*Submitted: April 2024; Accepted: February 2025.

1. Introduction

AI plays a vital role in data management for rendering efficient processes and enabling in-depth insights. Integrating AI technology into cloud computing services ensures, in particular, the accuracy of database questions and returns. Effectively, AI cloud services, called AI-as-a-Service (AIaaS), are the cloud-based platforms that provide AI capacities and resources to people and businesses (Tanimoto et al., 2013). These services make AI tools and technologies more functional, quantifiable, and not too costly for several uses. Research is still underway to maximize the latent potential of AI in business and commerce. The significant task for AI is to transform database management for businesses, whether on-premises or in the cloud to provide clients with the complex internet and computer services at a reasonable price (Algarni and Kudenko, 2017).

One of the primary profits of internet usage is the adaptability and flexibility of skills in response to changes in the capability of the assigned workload. Further, the data from redundant arrays of independent discs (RAID) will remain dispersed among several cloud providers, instead of concentrating on a single storage service (Bhardwaj et al., 2021). Hence, it is a good choice to use RAID in conjunction with significant data spread across multiple storage devices in order to avoid problems, caused by a single memory card (Zuluaga, Krause and Puschel, 2016).

Owing to varying levels of service efficiency and selling costs among cloud services, improving the cost and output time of numerous cloud providers becomes quite challenging. Because sending important records through the system can change the cost right away, it is important to look at the performance in terms of long-term costs and efficiency. As a result, it is necessary to create a flexible system that can adapt to the needs of cloud service providers and effectively fix the resulting lag time. With all of that in mind, as well as the need for a real way to keep rules up to date for distributed systems using different cloud hosting services with lower expected costs and shorter waiting times, the main goal of this study is to come up with an effective AI framework for better file delivery and a test in the cloud storage structure.

There are the following parts, composing the present paper, namely a literature review in Section 2, the presentation of the private cloud architecture of the AI-based framework in Section 3, the outline of the AI architecture for improving file distribution in Section 4, followed by an analysis of the results and then the final discussion and conclusions.

2. Literature review

This section presents an overview of key research papers, focusing on the challenges and innovations in applying AI to cloud storage systems. It also highlights the problems inherent in traditional methods and explores novel approaches to overcome them.

And so, Zuluaga, Krause and Puschel (2016) introduced the E-Pareto Active Learning mechanism, an effective approach for selecting the most informative samples from large pools of unlabeled data. Their model improves performance and reduces labeling costs. Similarly, Papaioannou et al (2012) developed a cloud-broking architecture that evaluates different cloud service providers based on document consumption data. This architecture not only saves storage costs but also enhances reliability and mitigates security threats. However, their analysis overlooks the impact of the platform on latency. Xu et al (2012) proposed a cohesive approach for configuring hybrid computers, enabling real-time data center auto-configuration by application service providers. This cohesive approach adjusts virtual machine (VM) source distributions and equipment restrictions based on cloud features and workloads, helps to achieve a balance between improving Service Level Agreement (SLA) goals and framework utilization. Their tests on Xen VMs with varied workloads demonstrated the strategy's effectiveness.

Zhang et al. (2021) proposed AI-driven techniques for managing resources in cloud-based data services, aiming to enhance efficiency, scalability, and adaptability in dynamic cloud environments. Zhou et al. (2015) developed a novel quantitative objective function to assess and optimize strategies in dynamic environments. Zhou and Liu (2018) explored the use of Graphics Processing Unit (GPU) based authorization frameworks, which leverage the parallel processing power of GPUs to accelerate authorization processes in systems with high throughput and intensive computational demands. These frameworks often integrate machine learning or cryptographic algorithms to enhance security and efficiency. Common areas of application include identity and access management, real-time data processing, and cryptographic operations. Zheng and Wen (2021) investigated how AI technologies are utilized in cloud computing resources and proposed a reliability evaluation method for assessing the dependability of cloud services within the Infrastructure-as-a-Service (IaaS) layer. Their method enables users to evaluate the reliability of their purchased cloud services quickly and easily. Robertson et al. (2021) contributed a rigorous approach to tackling system dynamics problems in AI applications. Hagemann and Katsarou (2020) outlined the main areas of AI theory, including deep learning (DL), inferential statistics, and machine learning (ML), providing an overview of how these models can be used to classify images as a cloud authentication method. Parashar et al. (2022) proposed a four-tier, vertically connected Cloud-

assisted Small Factory (CaSF) architecture, which also classifies and defines the primary AI technologies utilized within the CaSF.

Lin et al. (2019) introduced a cloud capability planning tool, featuring a comprehensive framework that includes a Relocation Type Comprehensive Making Plan and an AI Controller. This framework aims to streamline cloud resource management. Van Hasselt, Guez and Silver (2016) introduced feedback automation for continual agent critique, a method that can handle ongoing variable changes in interactive environments. Prasad et al. (2024) designed a cloud-based organizational management system, categorizing content based on its sensitivity. Approximately 18% of the information was stored in private clouds, while the rest was stored in public clouds, reinforcing the importance of privacy in cloud-based setups.

Caglar et al. (2023) further developed the ChArIoT cloud-based web platform, utilizing AI and the MERN Stack framework to identify mutations in Python code. Their solution demonstrated efficiency and adaptability across various scales, improving performance while saving time and resources. Cloud computing enables the efficient provision of a wide range of resources, such as software, storage, and analytics, fostering rapid innovation and adaptability. DL (Deep Learning), a subset of AI, is considered a crucial element of the fourth industrial revolution. The rise of cloud-based services has led to increased business collaboration, with AI significantly enhancing these processes. Research shows that organizations benefit substantially from integrating AI into public cloud strategies, improving the efficiency of cloud services and devices. In this context, Mohamed et al. (2022) described a methodological approach for managing cloud environments. Dhaya et al. (2022) used Gaussian Mixture Models for object detection, applying a Kalman filter to monitor movement and store the data in a private cloud data center.

AI-as-a-Service (AIaaS) combines AI and cloud computing, offering organizations easy access to AI capabilities without the need for complex on-site solutions. Numerous small and medium-sized enterprises (SMEs) view AIaaS as a promising option for adopting AI technologies. Griesch, Rittelmayer and Sandkuhl (2024) explored the significance of AIaaS, discussing the factors influencing its implementation and comparing AIaaS alternatives with on-site solutions. Kaur et al. (2021) examined the obstacles to and potential of AI devices for diverse populations, highlighting improved production efficiency, service quality, and customer support. Their work emphasizes the economic potential of AI and cloud computing for large organizations, particularly those with a significant consumer base. Belgaum et al. (2019) explored ML (Machine Learning) potential to describe and approximate complex nonlinear equations, though the practical application remains unclear. Bowers, Juels and Oprea (2009) applied the RAID principle to enhance data security in the cloud, but further results on

its impact were not provided. Bessani et al. (2011) developed DepSky, a tool that uses cryptography and steganography to enhance cloud data reliability, though its cost is higher than that of single online storage solutions. Finally, Bala, Rashid and Ismail (2024) focused on leveraging AI and ML in new computing frameworks, like cloud, fog, edge, and quantum-based computing to reduce costs while maintaining high performance.

3. Private cloud architecture of an AI-based framework

Existing methods have some flaws, such as underestimating the costs and delay times when using multiple cloud sources. This makes it more difficult to efficiently distribute data and ensure privacy while storing it (Dhaya and Kanthavel, 2022a). To address these issues, we propose an AI-based framework for private cloud computing that is both simpler and more effective. The goal of this study is to optimize the distribution and compression of documents across various private cloud data providers using AI. The AI architecture allows clients to distribute files across multiple cloud storage providers, focusing on AI-enhanced document availability features. The primary aim of AI-based systems in this context is cost optimization.

AI architecture is particularly suitable for this framework because AI plays a crucial role in the efficiency and flexibility of cloud computing models. Specifically:

- **Optimization efficiency:** AI algorithms can analyze and optimize resource allocation in real time, ensuring that workloads are balanced, resources are utilized efficiently, and cloud services operate seamlessly.
- **Analytics:** AI-based models can predict hardware failures, assist in capacity planning, and anticipate changes in user demand, enabling more proactive and smoother operations.
- **Automation:** The AI-driven cloud model minimizes the need for manual interventions, reducing human error and allowing the system to react quickly and automatically to issues.
- **Enhanced security:** AI improves security within cloud environments by detecting anomalies and potential threats more effectively than traditional methods.
- **Data management:** AI technologies facilitate the classification, indexing, and retrieval of data, making it easier to manage and use large datasets effectively.
- **User experience:** AI-driven improvements allow for smarter recommendations and personalized experiences, enhancing the usability and functionality of cloud applications.

- **Cost management:** AI-based models optimize costs by analyzing usage patterns and resource allocation, ensuring more cost-effective cloud operations.

The total cost of the proposed strategy comprises maintenance, storage, and network bandwidth. A critical factor influencing efficiency is the file-sharing mechanism between privatized services and client latency (Dhaya and Kanthavel, 2022b). Although retailer-side maintenance is not required, distributing AI-enhanced data across multiple private cloud providers can significantly improve the system's data quality and reliability (Voas and Zhang, 2009). The proposed model incorporates two AI techniques, as this is illustrated in Fig. 1. In this context, we address the multi-objective optimization problem, which is outlined below.

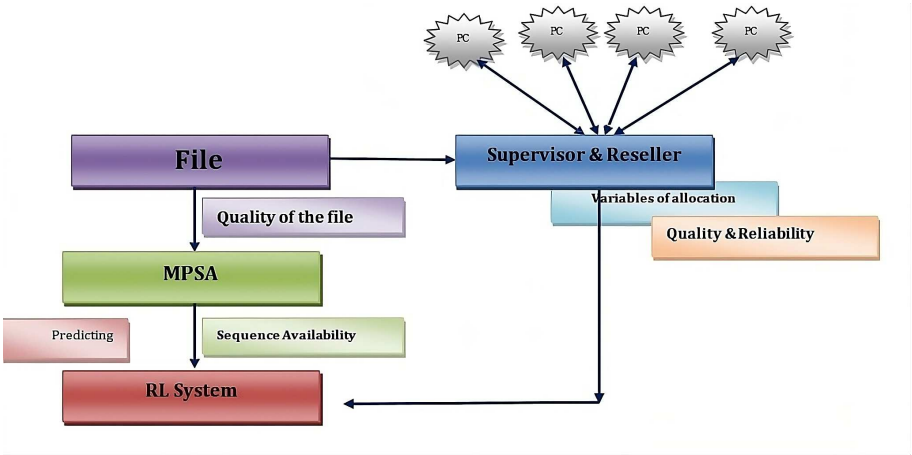

Figure 1. General sketch of the AI-based framework architecture

So, the system depicted in Fig. 1 addresses a multi-objective optimization problem that aims to ensure efficient resource allocation while maintaining high standards of quality and reliability. At the core of this system, the Supervisor & Reseller module is responsible for distributing files to multiple PCs based on a set of allocation variables. These variables are influenced by several key factors, including the quality of the file, the sequence availability predicted by the Mechanism for Predicting Sequence Availability (MPSA) module, and insights from an RL system. The RL system continuously predicts and updates optimal allocation strategies to improve performance. The overall objective is to simultaneously optimize multiple conflicting criteria such as file quality, allocation reliability, and availability of sequences, making the resource allocation process both intelligent and adaptive. Thus, Quality of the file (Q_f), Reliability

of allocation (R), Sequence availability and Prediction accuracy of resource requirements (P), Availability of PC nodes and Supervisor & Reseller's allocation rules (A_s) are to be maximized, subject to constraints on resource allocation. This can be formally written as:

$$\begin{aligned} &\text{Maximize } [Q_f(x), R(x), A_s(x), P(x)] \\ &\text{subject to } x \in X. \end{aligned} \tag{1}$$

Here, x ∈ X denotes the decision variables governed by the Supervisor & Reseller, balancing the allocation based on quality metrics and system predictions. The RL system continuously updates predictions to support dynamic allocation decisions. This structure allows for simultaneous minimization of costs and latency while maximizing reliability, which is ideal for AI-enhanced private cloud distribution strategies. An AI-based smart cloud computing schema model integrates AI techniques with cloud computing to enhance performance, efficiency, scalability, and automation. The functional description of the model is as follows:

Cloud Infrastructure: The foundational layer includes cloud servers, storage, and networking to provide the essential resources for deploying applications and managing data. The I Engine (RL System) layer, based on reinforcement learning, incorporates AI algorithms and models. This layer can perform data analysis, predictive modeling, decision-making, and automation.

Art Middleware (MPSA): This layer serves as an interface between cloud infrastructure and AI by optimizing resource usage, ensuring security, and automating processes such as scaling and load balancing.

RL determines how resources are allocated among several private cloud storage systems. This determination is based on a combination of factors, including each provider's efficiency, billing models, and directory activity patterns, which refer to how often files are created, accessed, modified, or deleted within each storage environment. During the training phase of the model, i.e., when the ANN is learning from historical data patterns and simulated operations, an ANN was used to support the RL mechanism in improving the performance and profitability of private cloud services. Pricing procedures refer to mechanisms that define how and when data operations occur, primarily focusing on cost-based decision-making rather than task scheduling, in the context of economic and market-based resource allocation strategies in cloud computing (Armbrust et al., 2010; Buyya et al., 2009). Throughout this training period, the ANN guided the RL system in learning optimal strategies for distributing files in a way that minimized cost and improved efficiency. The learning technique incorporates a novel mechanism that translates input state conditions, such as file access frequency, sensitivity, expected lifespan, and read/write rates into specific storage actions, like choosing the optimal cloud provider or determining

the file's replication level (Dhaya et al., 2019). In order to assess the long-term availability and relevance of a file, referred to here as its "survival", the MPSA analyzes the file's metadata and usage patterns. Based on these insights, the RL system predicts when the file is likely to be accessed again. It uses this information to make informed decisions on where and how much of each file to store across various providers. Using these RL predictions, the distribution engine (referred to as the "receiver") divides each file into fragments and assigns them proportionally to different private cloud providers. The resource manager (or storage orchestrator) is responsible for managing the logic of file segmentation and allocation, ensuring that each merchant receives an appropriate portion of the file based on predicted demand and storage efficiency. This orchestrator operates on the premise that files do not need to be fully stored by a single provider. Instead, each cloud provider stores a portion of the file, allocated according to their performance, security, and cost-efficiency characteristics. In the illustration of the proposed AI-based framework architecture, the following key terms are used:

- **File**: A file in this context refers to a digital data unit stored in the cloud. It contains the content that must be maintained and classified based on its quality and relevance. Assessing the quality refers to evaluating a file's value or importance based on its content structure, frequency of access, and update patterns, all of which influence storage decisions.
- **MPSA**: This mechanism analyzes a document's lifecycle and predicts how often it will be accessed or updated over time. The goal is to optimize file placement and reduce storage costs in private cloud environments by understanding usage patterns. The MPSA links a file's behavioral features with its structural attributes. These predictions allow the reinforcement learning system to pre-emptively adjust storage and access strategies.

The key features assessed by MPSA include:

- **Lifespan**: The duration of the period, during which a file remains active or relevant (Dhiman and Alghamdi, 2024).
- **Frequency of Reads**: The number of times a file is accessed over its entire lifespan.
- **Frequency of Writes**: This refers to how frequently the file is modified during its active life.

The MPSA was developed and trained using iterative learning algorithms applied to real-world data collected from large-scale storage systems, such as enterprise cloud platforms or data warehouses. These data sources include filing procedures related to various domains such as holidays, employment records, financial assessments, customer inquiries, and business planning documents (Ghavamipoor et al., 2020). The datasets simulate realistic and diversified storage scenarios. The log files used in model training contained metadata like ad-

min user activity, client location, file creation date, file formats, file categories, and document segmentation details. These entries provided the structural properties of each file, which the MPSA uses to predict future access behavior in terms of metadata organization, helping to inform how files are classified and distributed in the cloud.

MPSA Evaluation: Predictive modeling and ML are used to assess how well a model performs in forecasting target values. Sathyaraj et al. (2022) use predictive models to measure how accurately MPSA can predict key file behaviors, such as the frequency of reads/writes or lifespan, based on the file's metadata and historical patterns. To evaluate the accuracy and reliability of the MPSA, three well-established statistical and regression evaluation metrics were used.

- **Pearson's Correlation Coefficient (r)**: This statistical measure evaluates the linear relationship between the predicted values (e.g., predicted read/write frequency or lifespan) and the actual observed values from the dataset. It returns a value between -1 and 1, where:
  - +1 implies a perfect positive correlation,
  - 0 indicates no correlation, and
  - -1 signifies a perfect negative correlation.

In this context, a higher positive r value would indicate that MPSA predictions better follow the actual data trends.

Root Mean Squared Error (RMSE): RMSE calculates the square root of the average squared differences between the predicted and actual values. It reflects the magnitude of prediction error and is sensitive to large deviations. Lower RMSE values imply better predictive performance, meaning MPSA predictions are close to the real outcomes for file behaviors.

Mean Absolute Error (MAE): MAE computes the average of the absolute differences between predicted and actual values. Unlike RMSE, it treats all errors equally without amplifying larger ones. In the context of MPSA, a low MAE indicates that on average, the model's prediction errors are small, which is crucial for reliably planning file allocation and estimating storage efficiency.

Application in MPSA Context: These three metrics collectively evaluate how well the MPSA model forecasts: the number of times a file will be accessed or modified, how long the file will remain active, and whether the predicted usage pattern aligns with actual user behavior.

By applying Pearson's r, RMSE, and MAE, the framework ensures that the MPSA can Accurately predict file usage trends, improve resource allocation via the RL model, reduce unnecessary storage or access overhead, and ultimately enhance cost efficiency and system responsiveness.

For the Pearson correlation to be valid, both variables must be measured on either an interval scale or a ratio scale, where the differences between values are meaningful and consistent. Additionally, there should be a continuous relationship between the variables, meaning that changes in one variable correspond to measurable and consistent changes in the other. The Pearson correlation is sensitive to outliers, as extreme data points can disproportionately influence the results, potentially leading to inaccurate conclusions. Therefore, careful consideration is necessary of data points and ensuring that the relationship between the variables is at least sufficiently close to linear for a reliable interpretation of the correlation coefficient (Wan et al., 2018).

RMSE is a commonly used metric for assessing the accuracy of a model's predictions, particularly when dealing with quantitative data. This measure is often used in standard regression analysis:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(k_i - y_i)^2} \tag{2}$$

where k_i is the predicted value for the i^{th} observation and these are the values generated by the model or forecasting method; y_i is the actual measured value for the i^{th} observation, usually originating from some empirical dataset. n is the total number of observations.

Mean absolute error (MAE) is another common metric used to measure the accuracy of a model's predictions, in particular, in specific regression tasks. Unlike RMSE, which penalizes larger errors more due to squaring, MAE treats all errors equally:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|k_i - y_i| \tag{3}$$

where $| k_i - y_i |$ is the absolute error, arising between the predicted and actual value for an observation, indexed i.

The file system in the AI-based cloud framework offers enhanced accessibility, allowing files to be accessed from anywhere, which benefits file distribution. This improvement is also linked to cost-effectiveness, time reduction, file recovery, and universal file access. To measure these benefits, metrics such as RMSE, MAE, and Pearson's correlation can be used to assess the accuracy of predictions regarding cost, time consumption, and accessibility. The primary challenges in implementing the framework lie in ensuring optimization, predictive analytics, and enhanced security. For example, low RMSE values could indicate that the framework's time and cost predictions are accurate, while Pearson's correlation

could reveal how well file accessibility correlates with cloud service performance, helping to further optimize the system.

4. Effective AI Architecture for File Distribution Enhancement (EAIFDE)

RL frameworks incorporate ANNs, where the respective state vector refers to an organized representation of the environment at a given moment. It encodes the current conditions or features such as latency, cost, storage availability, or workload, used by the neural network to make decisions. In some designs, dimensionality reduction is applied to state vectors to streamline processing and reduce computational complexity. This means that instead of sending the entire raw state data, a smaller, more relevant subset of features is selected and passed to the ANN for decision-making. Over time, many cloud service providers have experienced security vulnerabilities, often due to service interruptions or unpredictable loads, which emphasize the importance of building systems that can adapt and recover quickly. While these may not always be direct security threats, they do highlight the need for resilient and adaptive distribution techniques (Kaur and Dilbag, 2021). In response to this need, a novel RL-based approach leveraging ANNs has been introduced, as depicted in Fig. 2.

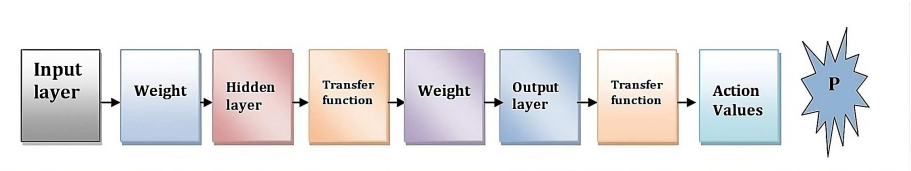

Figure 2. Private cloud services RL of file distribution enhancement

This proposed model is designed to optimize cloud file distribution by evaluating the key parameters, such as writing and reading durations, delay times, total costs, and median latency across multiple providers. Its performance is validated in the experimental section by comparing these metrics with existing methods. Additionally, reliability and security factors are assessed to demonstrate the robustness of the approach. The proposed system offers considerable improvements in optimization, predictive analytics, automation, cost control, and user experience. Key functionalities include:

- **Dynamic resource allocation:** Cloud resources scale up or down based on real-time demand, reducing unnecessary expenditures.
- **Predictive cost optimization**: Forecasting future costs based on usage trends.

- **Multi-cloud load balancing**: Distributing files efficiently to prevent congestion and bottlenecks.
- **Predictive analytics for delay detection**: Anticipating performance drops and proactively adjusting workloads.
- **Resource scheduling**: Prioritizing critical operations to minimize waiting time.

Each cloud region is treated independently. The ANN receives state features for each cloud (such as utilization rate, latency, and cost), provided by the MPSA. The ANN then generates a response (or policy output), which is translated into action parameters specific to file transfer decisions based on the context of each cloud region. The policy decision can be represented as:

$$(a_i)t = P(y_i)t \quad \forall\, p(t) \tag{4}$$

where $(a_i)t$ is the selected action for cloud provider i at time t. $P(y_i)t$ is the probability output (policy) from the ANN for that provider, and $p(t)$ represents the total collection of possible actions or cloud state categories at time t. This formula guides the RL agent in assigning a major portion of each file to the most suitable cloud provider. The rest of the file may be distributed across other clouds based on secondary scores. After execution, the RL system receives feedback (or rewards) from each environment based on achieved performance. These rewards (positive for fast, low-cost transfers and negative for delays or high costs) help in calculating the objective functions and failure probabilities using additional learning formulas (Sutton and Barto, 2018). Two primary factors influence the scoring system, namely cost and latency. To normalize latency, we compute:

$$\text{Normalized Latency Score} = (\text{Current Latency} \,/\, \text{Maximum Latency}) + 1. \tag{5}$$

This adjusted value ensures that higher latency results in lower performance scores. However, latency cannot always be reduced in proportion to cost increases, making trade-offs complex. Additionally, cloud costs fluctuate during billing cycles. This makes it difficult to precisely forecast cost for every operation, especially under heavy data movement. As a starting point, we use initial cost estimation formulas that aim to predict the upper bound of future expenses. By embedding AI into smart cloud storage systems, this architecture enables adaptive decision-making, significantly improving efficiency, minimizing operational cost, and ensuring better quality of service across geographically distributed cloud environments.

Algorithm 1: Algorithm 1: AI-Based File Distribution Process Using ANN

Input:

- Initial connection costs: C_ij and C_jk

- Number of events: Q
- Number of training iterations: N
- File metadata: size, access frequency, contact type

Begin
For event = 1 to Q do
For t = 1 to N do
pc_count ← Count available private cloud storage services
Configure ANN output layer with pc_count nodes
// Step 1: Feature acquisition
Input to ANN: file size, access frequency, contact model
For i = 1 to pc_count do
y_i(t) ← Estimated storage cost of the i-th private cloud node
End For
// Step 2: Normalized action generation
For i = 1 to pc_count do
a_i(t) ← y_i(t) / Σ_j y_j(t)
End For
Generate action vector a(t)
// Step 3: Reward tracking
in_i(t) ← incentive or utility received from i-th cloud storage
Compute prediction error for each private cloud node
Update ANN weights via backpropagation
End For
End For
End

In the proposed model, several key variables and procedures are used to evaluate the efficiency of data distribution across private cloud platforms. The variable O_cost refers to the total cost of cloud storage, including storage fees, data transfer charges, and additional overheads, associated with service usage. Then, O_utilized_i represents the amount of data stored at cloud provider i, where i ranges from 1 to I, the total number of cloud providers considered. The symbol S denotes the overall system cost, which incorporates both data storage and data transfer across cloud environments. AO_cost indicates the aggregate operational cost involved in executing a complete cloud-based workflow. The study aims to evaluate how cost increases in proportion to the data consumption

and delay across various service providers. Typically, the achieved resulting cost ratio (i.e., actual cost divided by benchmark cost) falls below 1, indicating efficiency.

However, since pricing and delay metrics follow different distribution patterns, the same formula cannot be applied to both. Hence, a custom cost-delay evaluation model was used. In this model, values exceeding 1 were treated as indicators of performance inefficiency and were flagged for adjustment through learning-based optimization. To better support decision-making, separate matrices were developed for cost and delay, allowing the architecture to minimize both simultaneously. Furthermore, the dynamic behavior of data files was analyzed using historical access patterns to assign importance ratings. Files predicted to be frequently accessed in the near future were prioritized for low-latency cloud allocation, whereas cost-efficient storage was prioritized for less active files. Algorithm 2, provided further on, defines the reward function based on these dynamics. Cost and latency reductions are further enhanced through the use of intelligent techniques, such as predictive cost modeling, load balancing, dynamic resource allocation, and auto-scaling. These allow the system to anticipate high-demand periods and adjust resources accordingly, minimizing costs without sacrificing performance. Additionally, predictive analytics are employed to forecast waiting times and mitigate potential bottlenecks, while task scheduling mechanisms prioritize critical operations to reduce delay.

Algorithm 2: Reward Function for File Distribution Across Private Cloud Providers

The algorithm serves to calculate the reward score for distributing each file across multiple private cloud storage providers by balancing cost and delay, while also considering file importance based on access patterns.

Inputs:

- nu.privateCloud: Number of available private cloud storage providers.
- File Parameters: Access frequency (read/write), file lifetime.
- midweight: Maximum observed file importance.
- maxDelayArray[i]: Maximum delay observed for cloud provider i.
- maxCostArray[i]: Maximum cost observed for cloud provider i.
- Weights (w): Scalar weight between 0 and 1 to adjust the importance of cost vs. delay.
- Historical references: Sathyaraj et al. (2022); Rao and Rao (2012).

Process:

Step 1: Compute File Importance Score (pa):

pa = nu.Read + nu.Write + lifeTime

// This score reflects how dynamic or frequently accessed a file is.

Step 2: Update Maximum Weight:

- If pa >maxWeight, then set maxWeight ← pa (indicating this file is currently the most significant).
- Normalize file importance: pa = pa / maxWeight

Step 3: Initialize Reward Arrays:

For each cloud provider i from 1 to nu.privateCloud, do:

Delay Reward: If fileDelay[i] >maxDelayArray[i], then update

maxDelayArray[is] ← fileDelay[i]

Compute delay-based reward:

DelayReward[i] = -1 * (fileDelay[i] / maxDelayArray[i])

Cost Reward:

Compute current cost: cost[i]

If cost[i] >maxCostArray[i], then update maxCostArray[i] ← cost[i]

Compute cost-based reward:

CostReward[i] = -1 * (cost[i] / maxCostArray[i])

Total Reward:

Combine cost and delay rewards using the weight w:

SumReward[i] = (1 − w) * CostReward[i] + w * DelayReward[i]

Step 4: Return:

Final reward score array SumReward[i] for all i, represents the suitability of each cloud provider for storing the file.

In the proposed reward-based file distribution mechanism, the parameter *pa* signifies the importance of a file, which is determined based on its access frequency, reflected through read and write operations and its expected lifespan. Files that are frequently accessed or are time-sensitive are deemed more critical and therefore are processed with lower latency during processing. To ensure fair comparisons across varying system conditions, normalization is applied, scaling all values relative to their maximum observed benchmarks.

The reward values, which are generated by the algorithm, are intentionally negative, as lower values correspond to more desirable outcomes, specifically, reduced cost and delay. Furthermore, a weighting factor w is introduced to balance the trade-off between cost and latency. By adjusting the value of this factor, w, users can fine-tune the system's behavior: for instance, setting $w = 0.7$ emphasizes minimizing delay, while $w = 0.3$ shifts focus toward cost-efficiency. This flexible configuration enables dynamic optimization, tailored to specific

application requirements and performance goals. The algorithm presented allows the system to dynamically evaluate and assign a reward score to each cloud provider, based on how well it balances storage cost and delay. Files with higher importance (e.g., frequent access or shorter lifetime) are prioritized for lower latency, while less active files are allocated to lower-cost options. The use of reward function ensures optimal distribution of data across cloud providers by penalizing high delay or cost and incentivizing performance improvements.

5. Experimentation

To evaluate the proposed algorithms, we employed a Java-based emulator. Specifically, the Cloud Spanner simulator, developed in Java, was used to simulate the cost and efficiency of various cloud storage services. This simulator is capable of computing the total cost associated with memory usage and network bandwidth for each cloud provider. Its flexible and efficient design enables the simultaneous simulation of multiple cloud storage platforms, as highlighted by Kapoor and Panda (2019).

The characteristics of the evaluated cloud vendors were modeled to reflect the strengths of the real-world platforms such as Google Cloud Storage, IBM Cloud, Microsoft Azure Storage, and Amazon Web Services (AWS), as this is illustrated in Fig. 3. Furthermore, Table 1 provides a detailed overview of the diverse cloud storage options offered by the respective providers.

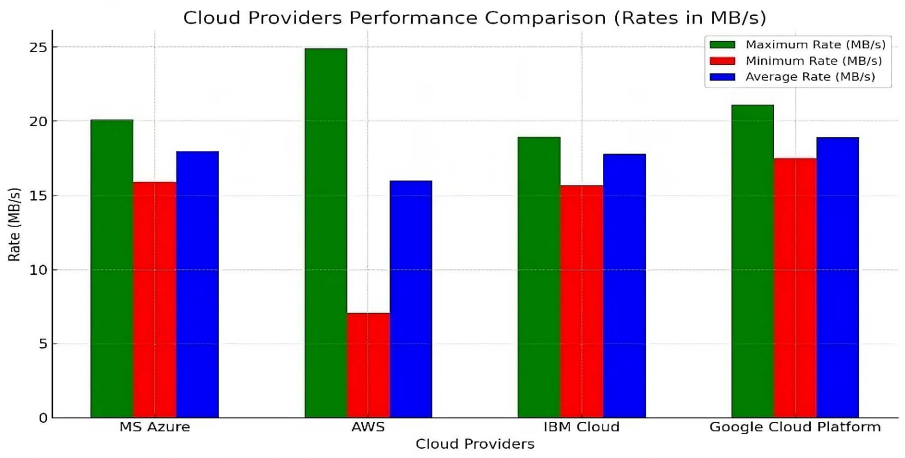

Figure 3. Range of cloud storage services

The MPSA begins by forecasting network traffic for each document as the initial step in the simulation process. Once the traffic behavior is estimated,

Table 1. The cloud storage options considered

Cloud providers	Maximum Rate MB/s	Minimum Rate MB/s	Average Rate MB/s
MS Azure cloud Storage	20.11	15.9	17.96
AWS	24.89	7.06	16.02
IBM Cloud	18.95	15.71	17.78
Google Cloud Platform	21.09	17.51	18.92

we proceed to train the RL algorithm using an ANN. The ANN receives key input features such as file size, file type, access frequency, and lifespan, which are essential for learning optimal data distribution strategies. Before the RL algorithm and ANN could be effectively utilized, it was necessary to configure several critical parameters, such as learning rate, exploration rate, and discount factor, which significantly influence the convergence and accuracy of the learning model. These features (referred to as "traits"), including file characteristics and predicted network load, play a central role in determining how effectively the system can allocate cloud resources. Table 2 summarizes the reinforcement learning parameter settings adapted from Zhou et al. (2015). The neural network includes a bias node in both the input and hidden layers, which maintains a stable activation value of +1, allowing the model to better generalize nonlinear relationships. Hidden layers are only included when data relationships are complex and not linearly separable. The output layer uses randomly initialized filter coefficients to estimate final outcomes, while the hidden layers utilize radial basis functions to map non-linear patterns in the input space. Moreover, understanding and analyzing network traffic is vital in the context of cloud database resource allocation. Traffic analysis allows systems to optimize performance, improve cost efficiency, detect potential security threats, and enhance the overall user experience by dynamically adjusting to workload demands.

Table 2. RL parameter configuration

Restraints	**values**
Number of preparation periods (events)	150 000
Knowledge speed	$\alpha = 0.001$
Succeeding situation decay parameter	$\gamma = 0.65$
Sequential reduction features	$\lambda = 0.6$

The values in Table 2 correspond to key parameters in the RL model. And so, α (learning rate) controls the degree, to which new information overrides

old information. For instance, a low value like 0.001 ensures slow but steady learning. Then, γ (discount factor) determines how much future rewards are valued compared to immediate ones. A value of this parameter equal 0.65 balances immediate and future rewards. Finally, λ (decay rate) governs the rate at which features' influence diminishes over time. A value of 0.6 for this parameter suggests a moderate reduction in influence.

As previously specified, the accessibility of private cloud beneficiaries, being determined by their operational parameters, such as bandwidth and storage usage, at each iteration step determines the number of production nodes. The iteration step refers to the process, in which the model updates its decisions or performs calculations based on new data inputs. These updates occur at each iteration, guiding the model toward optimal performance. Latency and cost are critical metrics in evaluating cloud providers for a given task. Latency refers to the time delay in data transmission, while cost refers to the charges associated with using the cloud services. These metrics play a significant role in assessing the efficiency and suitability of different cloud providers for specific tasks. A biased node refers to a node in the neural network that includes a bias term, which is added to the input of the node before passing it through an activation function. This bias term is typically set to a constant value, often +1, and is used to adjust the output of the node regardless of the input values. In both the input and hidden layers of the network, the biased node helps the model make consistent predictions by allowing the network to fit the data more flexibly. The output node uses stochastic filter coefficients, and Radial Basis Functions (RBF) are applied to all hidden layer connections. This configuration enables the network to model complex non-linear relationships, improving its ability to generalize and perform well across various scenarios.

After conducting a test by transmitting full files to each cloud service, the latency times and overall costs of each cloud provider are assessed. These metrics help in determining the most efficient cloud provider for the task at hand.

For reference, Vengerov (2008) provides foundational information regarding cloud computing performance metrics, such as latency and cost. Specifically, Vengerov's work outlines methodologies for comparing cloud providers based on their performance in distributed systems. The RAID method (Redundant Array of Independent Disks) is employed to ensure redundancy and fault tolerance in the data distribution process across cloud service providers. In this context, RAID is used for optimizing data storage and retrieval across multiple nodes, improving reliability and performance. The database software manages the distribution of data, ensuring that it is balanced and available across the cloud infrastructure.

Four datasets, totaling over 9,254 documents and 874 GB of data, were used to build the framework (see Klimt and Yang, 2004). The datasets are used to

simulate real-world cloud storage scenarios, testing each provider's ability to handle large-scale data transfer. For the evaluation, four cloud services were considered: Microsoft Azure Storage, Google Cloud Storage, IBM Cloud, and AWS. The choice of four services ensures a comprehensive comparison of latency, cost, and scalability. Notably, RAID-6 is employed, which provides three storage sites for data redundancy. The mention of the "fourth" alternative refers to a potential additional configuration or service provider used in the evaluation for comparison. The evaluation strategy involves simulating data distribution and assessing the performance of these cloud providers in terms of latency, cost, and scalability. The strategy is validated by comparing results across the four services in a controlled environment, simulating real-world cloud usage scenarios.

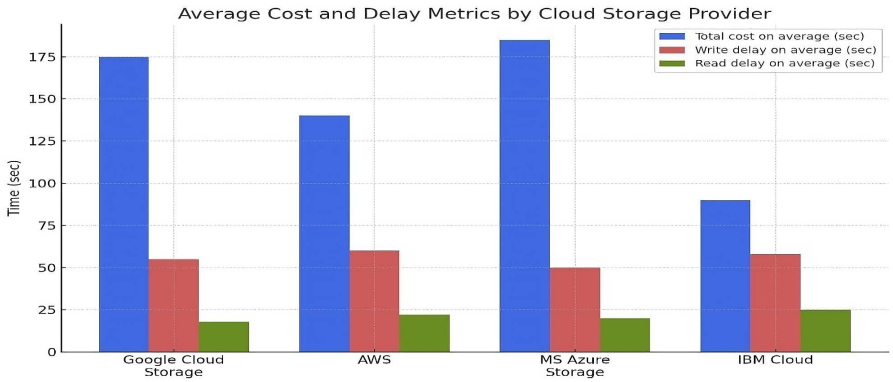

Figure 4. Average cost and delay metrics by cloud service provider

Figure 4 presents a comparative analysis of four major cloud service providers considered, i.e. Google Cloud Storage, AWS, Microsoft Azure Storage, and IBM Cloud in terms of their average total cost, write delay, and read delay, measured in seconds. The chart is derived from an experimental setup, within which entire datasets were transferred to each cloud platform without any segmentation or optimization (e.g., no intelligent file splitting or load balancing). This baseline approach enables the evaluation of inherent performance differences among providers. The blue bars represent the average total cost (in seconds, combining infrastructure cost and time-to-service), while the red and green bars represent write and read delays, respectively. The data reveal the following key insights: Google Cloud Storage and Microsoft Azure incurred the highest total cost, both exceeding 160 seconds on average, followed by AWS. IBM Cloud exhibited the lowest total cost, indicating a more economical data handling capability under the given load. Write delays (red bars) were consistently higher than read delays (green bars) across all providers, which is expected, given the overhead involved

in commit operations and consistency mechanisms. AWS showed a notably high write delay, while IBM Cloud maintained comparatively lower write and read delays. This baseline performance evaluation sets the context for testing the proposed intelligent framework. By transferring entire datasets directly (without intelligent division), the system exposes provider-specific bottlenecks and variations in service efficiency. These results emphasize the potential benefits of advanced resource allocation and data management strategies, such as the one proposed in this work.

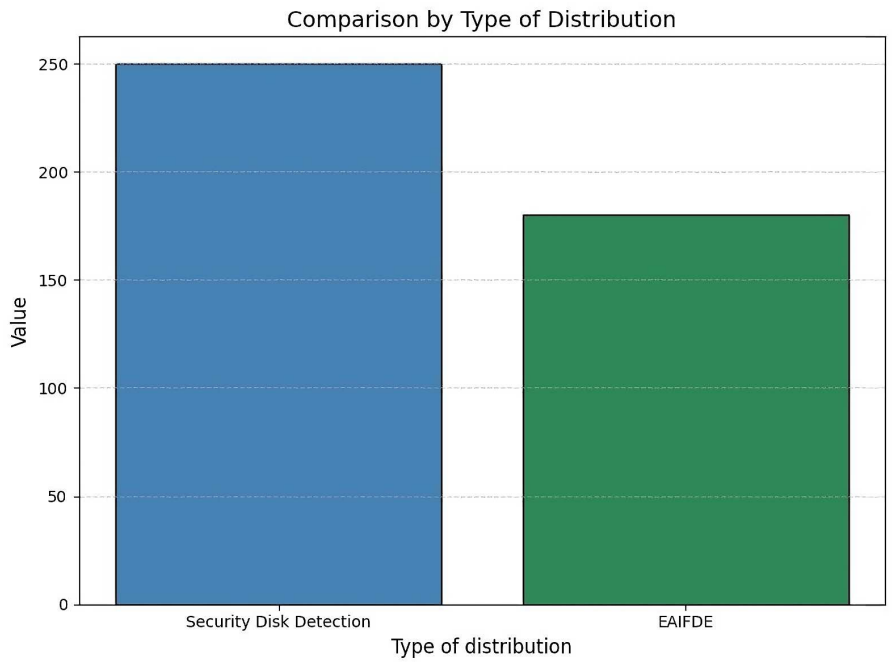

Figure 5. Comparison of expected costs for data distribution using EAIFDE and Security Disk Detection

Each data task or process in a cloud environment must be allocated the appropriate type and number of computational resources to ensure successful execution. Cloud systems typically consist of virtualized networks, operating systems, and dynamically provisioned resources, all of which can be adjusted or migrated across cloud infrastructures in real time.

To respond effectively to fluctuating user demands, dynamic resource allocation techniques are employed. These include on-the-fly scaling of computing power, memory, and storage, allowing the system to increase or decrease

resource availability, based on workload requirements. Additionally, network traffic analysis plays a crucial role in managing cloud performance. Through continuous monitoring of the network activity and accessibility, it becomes possible to identify bottlenecks, detect anomalies, and respond to potential threats, thereby improving system performance and ensuring security (Whiteson, 2012).

Following the baseline comparison, the same dataset was redistributed using the EAIFDE architecture, operating through the same private cloud storage provider. This experiment aimed at the evaluation of the efficiency improvements introduced by EAIFDE in contrast to traditional approaches. According to the results, presented in Figs. 5 and 6, the mean total cost of storage and access was significantly lower under the EAIFDE model when compared to the Security Disc Detection (SDD) method. Specifically, the proposed EAIFDE model achieved the 38.9% reduction in total cost over SDD and the improvement by 20.87% compared to previous EAIFDE configurations. In terms of performance, read and write delays were also substantially reduced. Compared to the SDD method, the EAIFDE framework lowered the read delay by over 71% and the write delay by 52%. Under the EAIFDE model, the observed mean read latency was 3.12 seconds, while the write delay was measured to be 0.82 seconds. These results confirm that the EAIFDE architecture not only improves cost efficiency, but also significantly enhances data access performance compared to conventional storage optimization strategies.

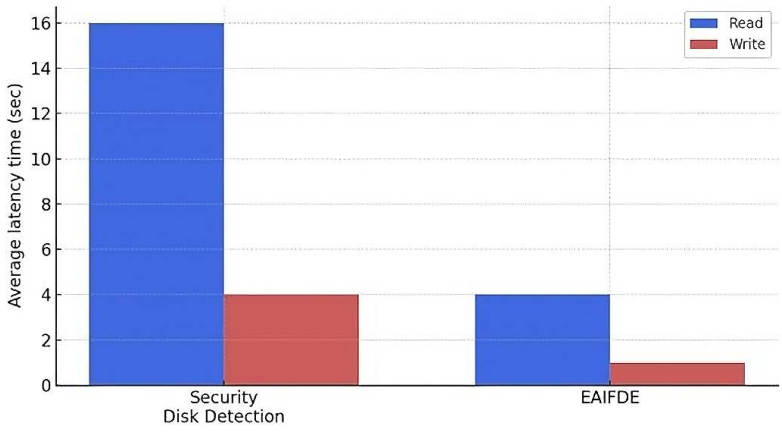

Figure 6. Comparison of average read and write latency times during document distribution using Security Disk Detection and EAIFDE frameworks

Figures 7 and 8 present a comparative analysis of the average total cost for three data distribution approaches: the Heuristic Approach (Singh and Sharma, 2019), OFDAMCSS (Tanimoto et al., 2013), as well as the proposed EAIFDE framework. These figures highlight the substantial efficiency improvements offered by EAIFDE. As shown in Fig. 7, the EAIFDE framework achieved the lowest average cost when distributing data across cloud providers. Specifically, it reduced the total cost by approximately 42.7% compared to the Heuristic Approach, and by 36.8% compared to the OFDAMCSS method. These reductions emphasize the economic advantage of using an intelligent, learning-based distribution model over the traditional or rule-based alternatives.

Figure 8 focuses on system latency, both for reading and writing operations. The EAIFDE design recorded the reading delay of 2.10 seconds and the writing delay of 0.77 minutes, outperforming the other two methods. Relative reductions in latency were approximately 68% for EAIFDE, 55% for OFDAMCSS, and 42% for the heuristic baseline. These values confirm EAIFDE's ability to enhance data access speed while maintaining cost efficiency. The term "latency effort and expenditure" in this context refers to the average time delays and operational costs associated with distributing all datasets across cloud storage services, without major disruptions or reallocation. These metrics differ from total cost and total latency in that they reflect ongoing system responsiveness and efficiency rather than cumulative resource usage.

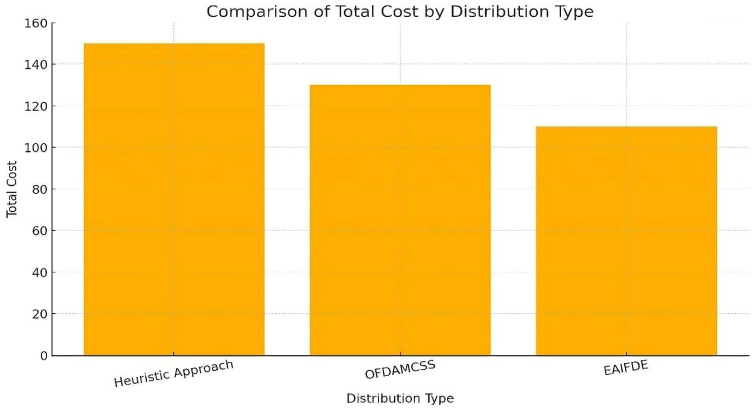

Figure 7. Comparison of average total cost for three different distribution approaches

Figure 9 compares three key performance metrics, highest, mean, and lowest disk-throughput speeds for four major cloud providers under Security Disk Detection. MS Azure Storage leads overall, peaking at 50 MB/s and maintaining

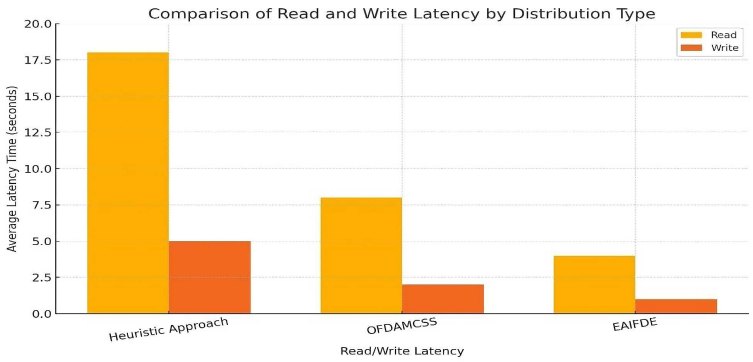

Figure 8. Comparison of read and write latency by distribution alternatives

Figure 9. Security Disk Detection speeds by various cloud providers (total cost and median latency time)

the highest average (44.5 MB/s) with its slowest read/write still at the solid 38 MB/s. AWS comes in second, with the top speed of 48 MB/s, the average of 41.5 MB/s, and the minimum of 35 MB/s. Google Cloud Storage follows, featuring the maximum of 45 MB/s, the average of 38.5 MB/s, and the floor of 30 MB/s. IBM Cloud trails slightly behind the other three, with its top speed at 42 MB/s, average at 37.2 MB/s, and the minimum of 33 MB/s. Overall, all providers exhibit relatively tight ranges between their highest and lowest speeds, but Azure consistently outperforms the others across every metric.

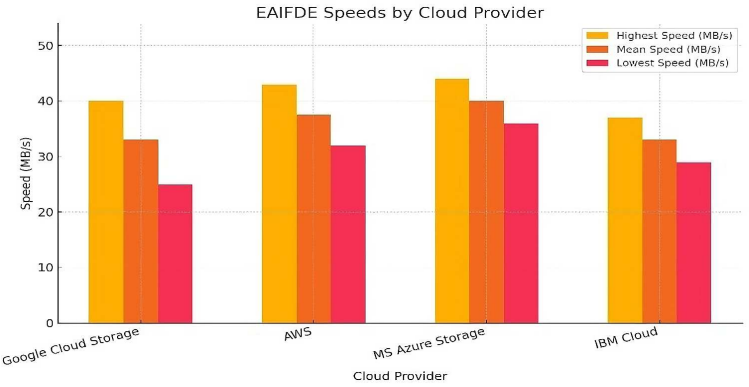

Figure 10. EAIFDE speeds for various cloud providers (total cost and median latency time)

Figure 10 compares three key performance metrics, highest, mean, and lowest disk-throughput speeds for four major cloud providers under EAIFDE. Thus, under EAIFDE, Google Cloud Storage delivers the peak throughput of 40 MB/s, the average of 33 MB/s, and the minimum of 25 MB/s. AWS slightly outperforms Google, topping out at 43 MB/s, averaging 37.5 MB/s, and dipping to 32 MB/s at its slowest. MS Azure Storage registers the highest overall speeds, with the maximum of 44 MB/s, the mean of 40 MB/s, and the floor of 36 MB/s. IBM Cloud rounds out the group with the top rate of 37 MB/s, the average of 33 MB/s, and the lowest speed of 29 MB/s. Hence, Fig. 10 clearly shows Azure's lead in both its peak and mean performance, while IBM Cloud exhibits the most modest throughput under EAIFDE. By incorporating these factors, the evaluation better reflects real-world usage patterns and provides a more accurate analysis of the system's design.

Figure 11 compares the latency times (i.e., the time required to read from or write to cloud storage) across different cloud storage providers, based on the OFDAMCSS approach. If this is a method to optimize cloud storage, the figure

should display how each provider performs under the OFDAMCSS approach, showing the difference in latency times.

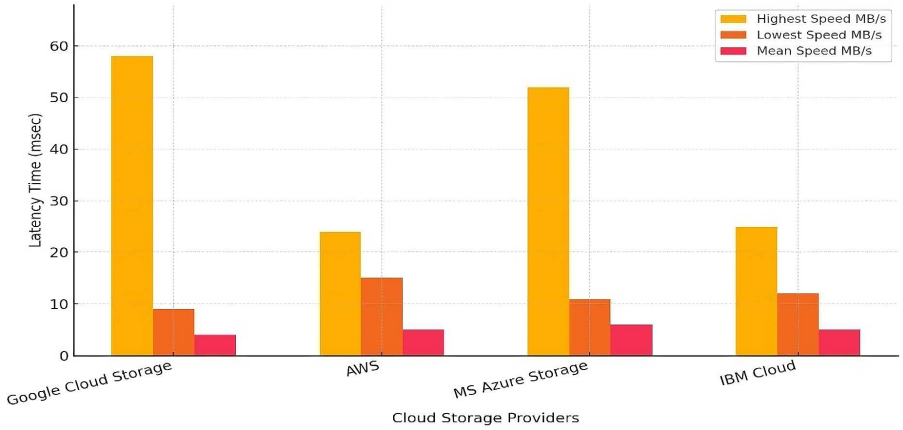

Figure 11. Latency time comparison of cloud storage providers based on OFDAMCSS

Now, Fig. 12, which follows, illustrates the latency performance (measured in seconds) of the four major cloud storage providers for the EAIFDE framework. It compares the highest, lowest, and mean latency times, represented in blue, red, and green bars, respectively. Among the providers, Google Cloud Storage exhibits the highest latency peak, indicating potential delays under maximum load, whereas IBM Cloud shows comparatively lower maximum latency. AWS and Azure demonstrate more balanced latency profiles, although Azure has a relatively higher peak than AWS.

The mean latency values across all providers remain fairly close, indicating consistent average performance. This comparison provides insight into the suitability of each provider for the EAIFDE framework, where minimizing latency is crucial for ensuring efficient and responsive data operations.

The effectiveness of the EAIFDE framework in minimizing storage costs and reducing delay time varies depending on the cloud platform and the data handling characteristics. To better assess this, the cost of data handling throughout its lifecycle was calculated by incorporating knowledge-based read and write operations, increasing the realism and accuracy of cost estimations. Fig. 13 presents storage costs across cloud platforms for various write operations (Write=1 to Write=6) using the EAIFDE framework. Similar trends are observed, with Google Cloud Storage and IBM Cloud continuing to offer lower costs, while AWS and Microsoft Azure are more expensive. These patterns high-

Table 3. Total cost and median latency time of each cloud provider considered using Security Disk Detection and EAIFDE method

Cloud Provider	Total cost and median latency time of each cloud provider assessed using Security Disk Detection					Total cost and median delay period of all cloud storage providers assessed using the EAIFDE method				
	Total Cost	**Median Latency (ms)**	**Highest Speed (MB/s)**	**Lowest Speed (MB/s)**	**Mean Speed (MB/s)**	**Total Cost**	**Median Delay (ms)**	**Highest Speed (MB/s)**	**Lowest Speed (MB/s)**	**Mean Speed (MB/s)**
Google Cloud Storage	78	12	45	30	38.5	74	9	40	25	33
AWS	30	18	48	35	41.5	28	16	43	32	37.5
MS Azure Storage	75	15	50	38	44.5	65	10	44	36	40
IBM Cloud	38	16	42	33	37.2	30	11	37	29	33

light the EAIFDE model's ability to optimize costs more effectively on specific cloud platforms, potentially influenced by various characteristics and optimizations inherent to each platform. Regarding delay times, the data transmission times to IBM Cloud and Google Cloud Storage were significantly reduced, while they increased slightly for AWS and Microsoft Azure. These differences may be attributed to the underlying infrastructure and service optimization techniques unique to each provider.

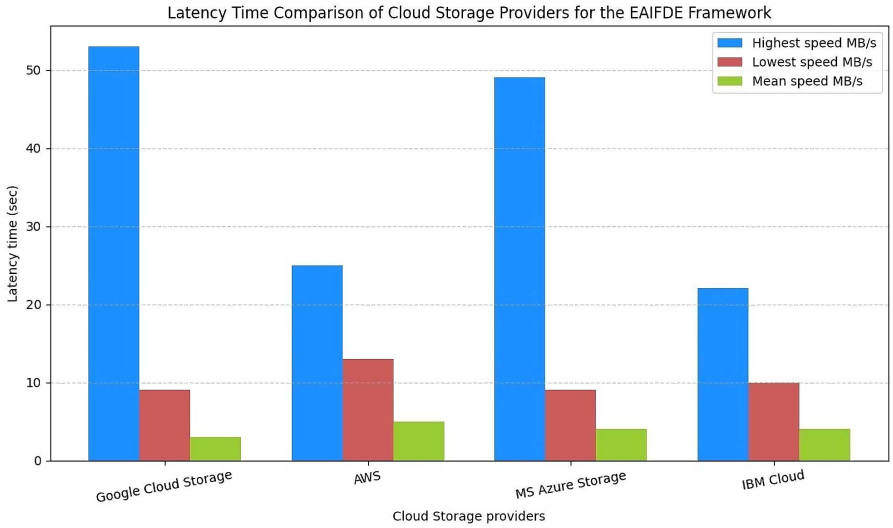

Figure 12. Latency time comparison of cloud storage providers for the EAIFDE framework

The proposed EAIFDE technique not only minimizes average costs and delay times but also enhances the functionality of cloud storage systems. It supports robust data backup strategies across multiple private cloud data centers, significantly reducing the risk of data loss. Even in the cases of system failures, data remains securely replicated in multiple geographic locations, ensuring continuity and resilience. To ensure effective management of such systems, it is essential to evaluate current storage performance metrics, assess capacity planning, and implement proactive monitoring strategies. These practices promote data integrity, scalability, and consistent performance, improving decision-making processes within private cloud environments. The randomness in parameter selection stems from the limited availability of commercial private cloud storage providers for benchmarking. Although this presents a challenge, the generalized results based on data from the four private clouds demonstrate the model's robustness. The success of the framework in optimizing cross-cloud file migration

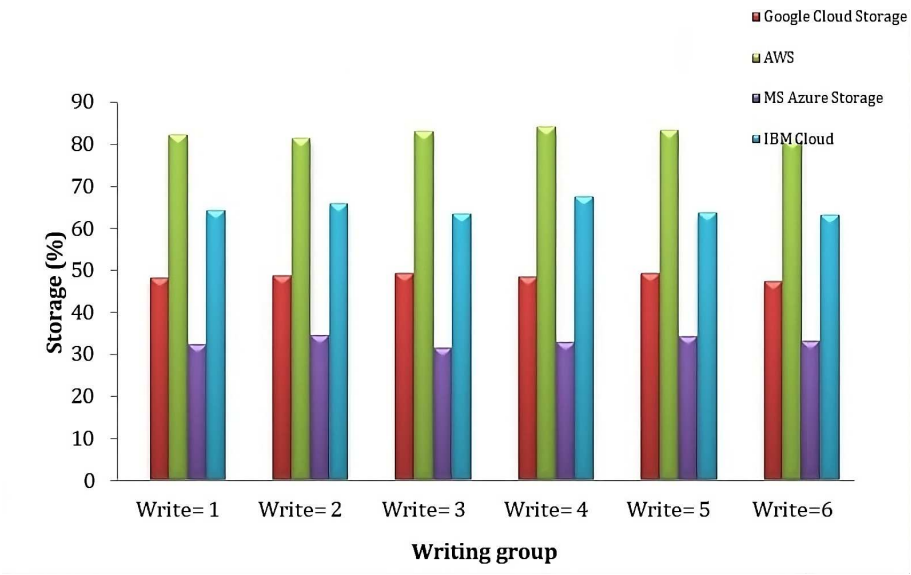

Figure 13. Comparison of storage cost (%) across different cloud providers for write operations

emphasizes its potential for broader adoption in secure, AI-driven private cloud ecosystems (Kanungo, 2024).

6. Discussion

This work demonstrates that multiple private cloud providers can significantly reduce costs and latency through adoption of the proposed Enhanced AI Framework for Data Efficiency (EAIFDE). The study's results show that, for specific providers, the cost of transferring files across multiple cloud storage platforms was reduced by approximately 38.9% compared to traditional data distribution methods. Additionally, the EAIFDE model outperformed heuristic techniques by approximately 27% in overall performance. It also achieved significant latency reductions: about 79% compared to the SDD approach and approximately 67% relative to empirical methods (Kushwaha et al., 2022). The SDD method, although designed to protect storage and ensure data confidentiality in AI-powered private clouds, was found to be less efficient. In contrast, the RL based EAIFDE framework effectively segmented data according to optimized storage strategies, resulting in an additional 61% cost reduction compared to the OFDAMCSS approach. Initial skepticism from some cloud providers regarding EAIFDE's capacity to manage multi-cloud data transfer strategies was ad-

dressed through experimental validation. The results confirmed that EAIFDE is capable of enhancing performance across a range of providers, including Google Cloud, Microsoft Azure, AWS, IBM Cloud, and simulated environments. Thus, the framework is validated as a scalable and cost-efficient model for optimizing multi-cloud operations (Van Hasselt, Guez and Silver, 2016).

To strengthen data management in private cloud infrastructures, the study recommends the adoption of techniques such as data stewardship, proactive monitoring, and regular data assessments. These practices enhance data confidence, availability, scalability, and reliability, while improving productivity and decision-making within cloud systems. While the concept of "data quality improvement" is relevant, its specific role within this study needs to be better defined in future work—for example, by exploring how the EAIFDE model indirectly contributes to quality via performance gains and intelligent allocation.

Due to the limited number of commercially available cloud platforms for experimentation, randomized parameters were used to generalize performance across varying conditions. The framework was tested with a minimum of three private clouds and demonstrated scalability to support up to ten cloud environments, depending on infrastructure capabilities (Gill et al., 2022). Furthermore, this research highlights the ongoing importance of SDD in private cloud systems to uphold data integrity and prevent unauthorized access (Van Hasselt and Wiering, 2007). In this context, the study integrated AI tools for anomaly detection and threat prediction, enhancing security monitoring. Cloud centers also implemented disc scanning tools to detect and classify devices, ensuring that only authorized storage media are connected.

In summary, the EAIFDE framework delivers a robust solution for intelligent, cost-efficient, and secure multi-cloud file distribution. It opens new possibilities for real-time cloud optimization by integrating reinforcement learning and advanced data segmentation techniques laying the foundation for smarter and more scalable cloud ecosystems.

7. Conclusions

In this paper, we proposed the EAIFDE as an innovative solution to improve file sharing in cloud storage systems. By leveraging reinforcement learning and data allocation techniques, the EAIFDE framework effectively reduces long-term storage costs and minimizes latency times, optimizing resource allocation across multiple cloud providers. Our experiments, conducted using a cloud storage system simulator, demonstrate that the EAIFDE model can substantially outperform traditional approaches by significantly reducing costs and improving service efficiency. Through the comparison of major cloud storage platforms, Google Cloud, Microsoft Azure Storage, AWS, IBM Cloud, and a cloud

emulator, we observed that EAIFDE dynamically adapts its data distribution strategy based on real-time cost and latency metrics. This approach allows stakeholders to make informed decisions by selecting providers that offer the best trade-off between cost, performance, and scalability. The findings suggest that EAIFDE not only enhances the economic efficiency of cloud storage systems but also supports scalability and adaptability in real-world cloud environments. Furthermore, the reinforcement learning-based approach opens up new avenues for improving cloud storage management in highly dynamic and competitive markets, where data transfer costs and latency are critical concerns. In future work, we aim to refine the reinforcement learning algorithm further to handle even more complex cloud architectures and to integrate additional performance metrics, such as security and availability, into the decision-making process. This would further strengthen the ability of the EAIFDE framework to provide optimal cloud resource management solutions.

References

ALGARNI, A. and KUDENKO, D. (2017) Distribution of Data Across Multiple Cloud Storage using Reinforcement Learning Method. In: *Proceedings of the 9th International Conference on Agents and Artificial Intelligence, ICAART 2017*, 431–438.

ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A. D., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A., STOICA, I. and ZAHARIA, M. (2010) A view of cloud computing. *Commun. ACM*, **53**, 4, 50–58. https:// doi.org /10.1145/1721654.1721672.

BALA, A., RASHID, R. Z. J. A., ISMAIL, I. et al. (2024) Artificial intelligence and edge computing for machine maintenance-review. *Artif. Intell. Rev.* **57**(1), 119. https://doi.org/10.1007/s10462-024-10748-9

BELGAUM, M. R., MUSA, S., ALAM, M. and MAZLIHAM, M. S. (2019) Integration Challenges of Artificial Intelligence in Cloud Computing, Internet of Things, and Software-Defined Networking. In: *Proceedings of the 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. IEEE, 1-5. https://doi.org/10.1109/MACS48846. 2019.9024828

BESSANI, A., CORREIA, M., QUARESMA, B., ANDR, F. and SOUSA, P. (2011) Depsky: Dependable and secure storage in a cloud-of-clouds. In: *Proceedings of the Sixth Conference on Computer Systems, EuroSys '11*, New York, NY, USA. ACM, 31–46.

BHARDWAJ, H., TOMAR, P., SAKALLE, A. and BHARDWAJ, A. (2021) Classification of extraversion and introversion personality trait using electroencephalogram signals. In: *International Conference on Artificial Intelligence and Sustainable Computing*. Springer, Cham, 31-39.

BOWERS, K. D., JUELS, A. and OPREA, A. (2009) Hail: A high-availability and integrity layer for cloud storage. In: *Proceedings of the 16th ACM Conference on Computer and Communications Security.* New York, NY, USA. ACM, 187-198.

BUYYA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J. and BRANDIC, I. (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, **25**, 6, 599–616 . doi: 10.1016/j.future.2008.12.001.

CAGLAR, O., TASKIN, F., BAGLUM, C., ASIK, S. and YAYAN, U. (2023) Development of Cloud and Artificial Intelligence based Software Testing Platform (ChArIoT). In: *Proceedings of the 2023 Innovations in Intelligent Systems and Applications Conference (ASYU).* IEEE, 1-6. https:// doi.org /10.1109/ASYU58738.2023.10296551

DHAYA, R. and KANTHAVEL, R. (2022a) Applying Deep Convolution Neural Network (DCNN) Algorithm in the Cloud Autonomous Vehicles Traffic Model. *The International Arab Journal of Information Technology*, **19**, 2, 186-194.

DHAYA, R. and KANTHAVEL, R. (2022b) Dynamic automated infrastructure for efficient cloud data centre. *Computers Materials & Continua* **71**(1), 1625-1639. https://doi.org/10.32604/cmc.2022.022213

DHAYA, R., DEVI, M., KANTHAVEL, R., ALGARNI, F. and DIXIKHA, P. (2019) Exploration of Maximizing the Significance of Big Data in Cloud Computing. In: *International Conference on Emerging Current Trends in Computing and Expert Technology.* Springer, Cham, 695-702.

DHAYA, R., KANTHAVEL, R. and MAHALAKSHMI, M. (2022) Enriched recognition and monitoring algorithm for private cloud data centre. *Soft Computing*, 26. 12871–12881 https://doi.org/ 10.1007/ s00500-021-05967-z

DHIMAN, G. and ALGHAMDI, N. S. (2024) SMoSE: Artificial Intelligence-Based Smart City Framework Using Multi-Objective and IoT Approach for Consumer Electronics Application. *IEEE Transactions on Consumer Electronics*, 70, 1, 3848-3855. doi: 10.1109/TCE.2024.3363720

GHAVAMIPOOR, H., MOUSAVI, S. A. K., FARAGARDI, H. R. and RASOULI, N. (2020) A Reliability Aware Algorithm for Workflow Scheduling on Cloud Spot Instances Using Artificial Neural Network. In: *Proceedings of the 2020 10th International Symposium on Telecommunications (IST).* IEEE, 67-71. https:// doi.org/ 10.1109/IST50524.2020.9345896

GILL, S. S., XU, M., OTTAVIANI, C., PATROS, P., BAHSOON, R., SHAGHAGHI, A., GOLEC, M., STANKOVSKI, V., WU, H., ABRAHAM, A., SINGH, M., MEHTA, H., GHOSH, S. K., BAKER, T., PARLIKAD, A. K., LUTFIYYA, H., KANHERE, S. S., SAKELLARIOU, R., DUSTDAR, S., RANA, O., BRANDIC, I. and UHLIG, S. (2022) AI for next generation

computing: Emerging trends and future directions. *Internet of Things*, 19, 100514. https://doi.org/10.1016/j.iot.2022.100514

GRIESCH, L., RITTELMEYER, J. and SANDKUHL, K. (2024) Towards AI as a Service for Small and Medium-Sized Enterprises (SME). In: J. P. A. Almeida, M. Kaczmarek-Heß, A. Koschmider and H. A. Proper, eds., *The Practice of Enterprise Modeling. PoEM 2023. Lecture Notes in Business Information Processing* **497**. Springer, Cham. https://doi.org/10.1007/978-3-031-48583-1_3

HAGEMANN, T. and KATSAROU, K. (2020) A Systematic Review on Anomaly Detection for Cloud Computing Environments. In: *2020 3rd Artificial Intelligence and Cloud Computing Conference (AICCC 2020)*. Kyoto. Japan. ACM, New York, NY, USA, 14, 18–20.

KANUNGO, S. (2024) AI-driven resource management strategies for cloud computing systems, services, and applications. *World Journal of Advanced Engineering Technology and Sciences*, **11** (02), 559–566.

KAPOOR, S. and PANDA, S. N. (2019) Energy-Aware Parallel Computing for Cloud Architecture using ANN. In: *2019 Global Conference for Advancement in Technology (GCAT)*. IEEE, 1-4.

KAUR, M. and DILBAG, S. (2021) Multi objective evolutionary optimization techniques based hyperchaotic map and their applications in image encryption. *Multidimensional Systems and Signal Processing*, **32**, 1, 281–301.

KAUR, M., SINGH, D., KUMAR, V., GUPTA, B. B. and ABD EL-LATIF, A. A. (2021) Secure and Energy Efficient-Based E-Health Care Framework for Green Internet of Things. *IEEE Transactions on Green Communications and Networking*, **5**, 3, 1223–1231.

KLIMT, B. and YANG, Y. (2004) The Enron Corpus: A New Dataset for Email Classification Research. Carnegie Mellon University https://www.cs.cmu.edu/~enron/

KUSHWAHA, S. S., JOSHI, S., SINGH, D., KAUR, M. and LEE, H.-N. (2022) Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. *IEEE Access*, 10, 6605-6621.

LIN, C., SUN, H., HWANG, J., VUKOVIC, M. and ROFRANO, J. (2019) Cloud Readiness Planning Tool (CRPT): An AI-Based Framework to Automate Migration Planning. In: *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 58–62.

MOHAMED, N., SRIDHARA RAO, L., SHARMA, M., SURESH BABU, R., ALFURHOOD, B. S. and SHUKLA, S. K. (2023) In-depth Review of Integration of AI in Cloud Computing. In: *Proceedings of the 2023 3rd International Conference on Advanced Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 1431–1434. https://doi.org/10.1109/ICACITE57410. 2023.10182738

PAPAIOANNOU, T., BONVIN, N. and ABERER, K. (2012) Scalia: An adaptive scheme for efficient multi-cloud storage. In: *High Performance Computing. Networking, Storage and Analysis (SC). 2012 International Conference*. ACM, 1-10.

PARASHAR, V., KASHYAP, R., RIZWAN, A., KARRAS, D. A., ALTAMIRANO, G. C., DIXIT, E. and AHMADI, F. (2022) Aggregation-Based Dynamic Channel Bonding to Maximize the Performance of Wireless Local Area Networks (WLAN). *Wireless Communications and Mobile Computing*, 2022, Article ID 4464447, 1-11. https://doi.org/10.1155/2022/4464447

PRASAD, N., PANDIAN, P. K. G., NUGURI, S., SAOJI, R. and DEVAGUPTAPU, B. (2024) Integration of Cloud Computing, Artificial Intelligence, and Machine Learning for Enhanced Data Analytics. *International Journal of Intelligent Systems and Applications in Engineering*, **12**(22s), 11–20.

RAO, R. and RAO, S. (2012) Application of Artificial Neural Networks in Capacity Planning of Cloud Based IT Infrastructure. In: *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 1-4.

ROBERTSON, J., FOSSACECA, J. M. and BENNETT, K. W. (2021) A Cloud-Based Computing Framework for Artificial Intelligence Innovation in Support of Multidomain Operations. *IEEE Transactions on Engineering Management*, **69**(6), 3913–3922. https://doi.org/10.1109/TEM.2021.3088382

SATHYARAJ, R., KANTHAVEL, R., CAVALIERE, L. P. L., VYAS, S. and MAHESWARI, S. (2022) Analysis on Prediction of COVID-19 with Machine Learning Algorithms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 30 (Supplement 1), 67–82.

SINGH, P. and SHARMA, A. (2019) Heuristic Approaches for Efficient Cloud Resource Management and Load Balancing. *Future Generation Computer Systems*, 92, 66-80.

SUTTON, R. S. and BARTO, A. G. (2018) *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

TANIMOTO, S., SAKURADA, Y., SEKI, Y., IWASHITA, M., MATSUI, S., SATO, H. and KANAI, A. (2013) A study of data management in hybrid cloud configuration. In: *Proceedings of the 2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel / Distributed Computing*. IEEE, 381-386.

VAN HASSELT, H. and WIERING, M. A. (2007) Reinforcement learning in continuous action spaces. In: *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, IEEE, 272–279.

VAN HASSELT, H., GUEZ, A. and SILVER, D. (2016) Deep reinforcement learning with double Q- Learning. *AAAI,* 2094–2100.

VENGEROV, D. (2008) A reinforcement learning framework for online data

migration in hierarchical storage systems. *The Journal of Supercomputing*, **43**, 1, 1-19.

VOAS, J. and ZHANG, J. (2009) Cloud computing: New wine or just a new bottle? *IT Professional*, 11, 15–17.

WAN, J., YANG, J., WANG, Z. and HUA, Q. (2018) Artificial Intelligence for Cloud-Assisted Smart Factory. *IEEE Access*, 6, 55419 - 55430. doi: 10.1109/ACCESS.2018.2871724.

WHITESON, S. (2012) *Evolutionary Computation for Reinforcement Learning*. Springer, Heidelberg-New York-Dordrecht-London, 325–355.

XU, C.-Z., RAO, J. and BU, X. (2012) A unified reinforcement learning approach for autonomic cloud management. *Journal of Parallel and Distributed Computing*, **72**, 2, 95–105.

ZHANG, Y., WANG, Y. and LIU, J. (2021) AI-Driven Resource Management for Cloud-Based Data Services. *IEEE Transactions on Cloud Computing*, **9**(4), 1230–1242. https://doi.org/10.1109/TCC.2020.2994822

ZHENG, Y. and WEN, X. (2021) The Application of Artificial Intelligence Technology in Cloud Computing Environment Resources. *Journal of Web Engineering*, **20**(6), 1853–1866. https://doi.org/10.13052/jwe1540-9589.2067.

ZHOU, A. C., HE, B., CHENG, X. and LAU, C. T. (2015) A declarative optimization engine for resource provisioning of scientific workflows in iaas clouds. In: *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing, HPDC '15*. New York, NY, USA. ACM, 223–234.

ZHOU, Z. and LIU, X. (2018) GPU-Accelerated Security Algorithms for Cloud-based Authentication Systems. *Journal of Cloud Computing: Advances, Systems, and Applications*, **7**(1), 25-42.

ZULUAGA, M., KRAUSE, A. and PUSCHEL, M. (2016) E-pal: An active learning approach to the multi-objective optimization problem. *Journal of Machine Learning Research*, 17, 1–32.