International Journal of Data Informatics and Intelligent Computing (IJDIIC)
Vol. 5, No. 1, March 2026, pp. 50~61
DOI: 10.59461/ijdiic.v5i1.262

Integrating Federated Transfer Learning for Secure Multi-Tenant Data Management in Decentralized Cloud Infrastructures

Aditya Gupta[1], Sai Kiran Oruganti[2]
[1]Department of Computer Science and Engineering, Lincoln University College, Petaling Jaya, Selangor Darul Ehsan, 47301, Malaysia.
[2]Faculty of Engineering and Built Science, Lincoln University College, Petaling Jaya, Selangor Darul Ehsan, 47301, Malaysia.

Article Info

Article history:

Received February 02, 2025
Revised March 10, 2026
Accepted March 19, 2026

Keywords:

Federated Transfer Learning
Cloud Infrastructures
Tenant-Aware Security
Privacy-Preserving
Data Management

ABSTRACT

Ensuring secure multi-tenant data management while enabling cross-domain knowledge sharing is challenging in decentralized cloud infrastructures. Traditional centralized learning methods pose risks like data leakage and non-compliance with privacy regulations. To address these concerns, this research integrates Federated Transfer Learning (FTL) for secure and efficient multi-tenant data management. The proposed approach employs Federated Learning (FL) to enable collaborative model training while keeping raw data localized, preserving privacy. Additionally, BERT-based transfer learning improves knowledge sharing by adapting pre-trained models to tenant-specific tasks, enhancing efficiency and reducing computational overhead. A tenant-aware security mechanism dynamically assesses trust levels to ensure secure workload allocation and mitigate risks. Furthermore, a decentralized aggregation strategy enhances data privacy and prevents single-point failures, improving system robustness. The framework was evaluated using real-world datasets, assessing privacy, adaptability, communication overhead, and computational efficiency. Experimental results demonstrate that FTL-driven decentralized architectures achieve low latency (6ms), high throughput (850000 requests/sec at 10000 tenants), better utilization of resources (83%), and good security compliance (9.5/10). Consistency models also come with a 50% overhead for strong consistency, which shows the trade-offs involved in ensuring data consistency. The results confirm the proposed model as a scalable, efficient, and privacy-preserving solution for multi-tenant cloud environments.

Corresponding Author: Aditya Gupta (e-mail: gupta.aditya56@gmail.com)

1. INTRODUCTION

Large-scale data processing in cloud environments has become essential for modern industry and research organizations to fulfill their ever-increasing computing demands. Because of this increase in data volume, businesses are adopting multi-tenant cloud infrastructures, which provide cost-effectiveness, scalability, and elasticity. However, the extensive use of cloud services for big data analytics also raises several security and privacy issues, which are made worse by the complexity of large-scale distributed systems [1]. In settings where several tenants share hardware, network segments, and frequently operating system resources, sensitive data is handled and stored. In this situation, ensuring security necessitates a deep comprehension of the emerging threat vectors that malicious actors may exploit, as well as the underlying computing architecture. Because physical hardware and virtualization platforms function together in the multi-tenant paradigm, adversaries may be able to compromise confidentiality and integrity by taking

advantage of side-channel leakage, hypervisor flaws, or cross-tenant data access [2]. The growing complexity of isolation techniques introduced by cloud service providers still cannot supply the entire range of attack surfaces that arise from varying levels of resource sharing. Sensitive data stored within these facilities presents a higher risk because the locality incorporates valuable items like financial records, intellectual property, and personal identifying information. Any security breach in this situation could lead to catastrophic consequences, which include massive financial damage, reputational damage, and operational interruptions [3]. Framework development for big data analytics security primarily comes from the need to satisfy legal compliance, together with data protection rules. International governments have established strict regulations ordering that businesses maintain comprehensive security protocols for managing and storing their data implementations throughout their systems. Security measures protect all three forms of big data existence: data at rest, data in transit, and data in use [4].

Street assemblies of different software components, including resource management layers, distributed file systems, and parallel data processing engines, support typical big data pipelines during massive dataset processing operations. These various features enable possible utilization. Security precautions, along with cryptographic costs, tend to create delays in system performance when system loads become substantial [5]. For any complete structure, security needs and system performance demands must be solved at once. Security configurations in multi-tenant environments require double functionality because resource conflicts between tenants remain unpredictable; therefore, protection systems need resilience and adaptability. The struggle persists to find the most suitable method of consolidating cryptographic capabilities with big data analytics, which achieves proper security safeguards while maintaining computing effectiveness [6]. The selection of cryptographic protocols for the different stages of the data pipeline is one of the complexities. For example, conventional block-based encryption schemes may be computationally expensive for streaming data analysis but may be suitable for data at rest. Additionally, while partial homomorphic encryption methods offer secure computation on encrypted data, they often accompany additional complexity and issues related to memory usage and key management [7]. Sophisticated scheduling algorithms to potentially manage the interaction between encryption processes and computational workloads are also required to ensure resource availability within dynamic, heterogeneous environments. Scheduling also has to consider resource utilization, completion time of a job, and little security needs like real-time encryption of intermediate data or creation of transient keys [8].

Federated Transfer Learning (FTL) is a solution to allow secure multi-tenant data processing in decentralized cloud systems. The system protects data privacy through FL and uses TL to expand information sharing between different tenant systems. A security mechanism, together with decentralized aggregation functions, is used to protect the system against attacks while improving its overall resilience. The system is tested using real-world datasets, where privacy, adaptability, communication overhead, and computation efficiency are optimized.

Key Contribution

- The research recommends FTL to securely process multi-tenant data in decentralized cloud environments while maintaining data privacy.
- The Bidirectional Encoder Representations from Transformers (BERT) transfer learning achieves pre-trained model transfer to tenant-based tasks, improving knowledge sharing, learning efficiency, and lowering computational overhead.
- A tenant-aware security module dynamically evaluates the level of trust, providing secure workload distribution and adversarial resistance.
- The suggested decentralized aggregation solution enhances data privacy and system resilience, as well as avoids single points of failure in multi-tenant cloud services.

2. RELATED WORKS

Advanced cybersecurity methods were presented to safeguard data in cloud and decentralized networks, including confidentiality, integrity, and availability [9]. Homomorphic encryption, quantum-resistant cryptography, and zero-trust models were important techniques that were discussed. It also emphasized secure multi-party computation (SMPC) and distributed ledger technologies (DLTs) to gain privacy and data scalability. However, real-world implementation needs to be further examined. An elliptic curve cryptography-based energy-efficient routing protocol (ECC-EERP) was developed to provide security for patient information and lower energy consumption in healthcare 5.0 networks [10]. Public and private key pairs were applied for encryption and decryption in wireless sensor networks (WSNs). The approach outperformed other solutions in terms of security, energy efficiency, and network performance. Scalability and accommodating the system on different hardware platforms need further improvements.

Resource management in federated cloud infrastructures was intricate, particularly when dealing with heterogeneous monitoring needs across various tenants [11]. The Federated Architecture for Resource Management and Monitoring in Clouds Version 1.0 (FEDARGOS-V1) architecture was implemented to monitor resources and detect constraints effectively. It was executed on a real-time OpenStack-based FEDGEN cloud testbed, demonstrating more efficient and scalable monitoring. Yet, there was a need to improve diverse tenant requirements handling and complete scalability. Security and privacy within multi-tenant cloud systems were critical issues, especially in green computing [12]. Blockchain technology was considered a solution to the issues at hand, and it was employed with Ganache and MetaMask to establish secure cloud tenant accounts. The outcomes showed that blockchain enhanced security and privacy. Improvements were indeed needed in terms of scalability and integration with existing cloud systems. Privacy and security issues in multi-domain, multi-tenancy environments were a major concern in 6G networks [13]. An investigation into whether Federated Analytics (FA) can enhance the performance, security, and confidentiality of data in such domains was performed. It also identified synergy between FA and FL for network orchestration. Further research is necessary on practical deployments and handling intricate multi-domain situations. Table 1 presents a comparison between the proposed method and related approaches.

Table 1. Compares the proposed work with the related work

Studies	Scalability	Security	Throughput
[10]	Yes	No	No
[12]	No	Yes	Yes
[14]	Yes	No	No
[15]	No	Yes	No
[16]	No	Yes	No
Proposed	Yes	Yes	Yes

3. METHODOLOGY

In modern multi-tenant environments, ensuring security, privacy, and efficient resource allocation is crucial. Exploring a Tenant-Aware Security Mechanism that integrates Federated Learning (FL) to enhance data privacy while maintaining model accuracy across tenants is performed in this research. By leveraging BERT-based transfer learning, optimization of tenant-specific tasks without direct data sharing is executed. Additionally, the decentralized aggregation strategy strengthens system resilience by distributing model updates across multiple secure aggregators, minimizing security risks and performance bottlenecks. Figure 1 illustrates the overall structure of the proposed model.

3.1. Security Challenges

Cloud networks need encryption and both permission rules and reliability protocols to properly secure multi-tenant data. During FL operations, raw data leakage is prevented, but vulnerabilities can occur while updating models and when tenants interact with each other. The use of different levels of data isolation causes security issues, which also affect collaborative work. The protection of model updates through encryption must remain uncomplicated. Security for tenants remains secure through this mechanism, although decentralized aggregation might permit gradient leakage vulnerabilities to occur. A secure solution requires three features for encrypted communication, detailed access permissions, and trust assessment features to protect privacy while enabling secure multi-tenant cooperation, as described in [14].

3.2. Federated Learning (FL) for Multi-Tenancy

A methodological approach integrates FL within the cloud framework that includes upgraded data privacy functions and AI model instruction. FL is considered a privatized machine learning (ML)-based training framework that operates without central control. FL provides distinct functionality from standard ML by permitting data to remain on local devices, where only model changes get transmitted to a central server for combination purposes. However, the system does not require the full transfer of data to a central server for aggregation. This concept attracted a lot of attention since it addressed privacy issues in delicate sectors like healthcare and banking. Early research also concentrated on confirming that it was possible to train models on non-sensitive data and implementing FL for model training in small-scale distributed networks.

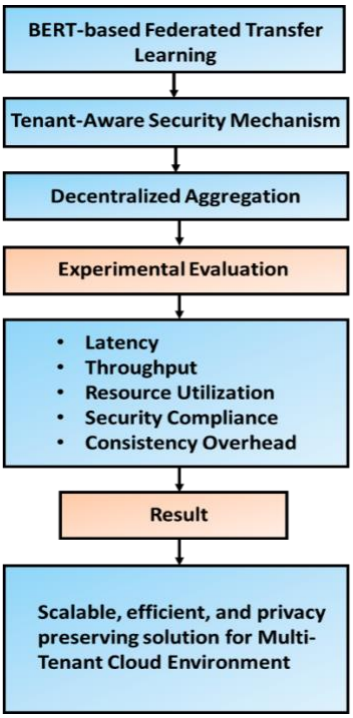

Figure 1. Overall framework of the proposed model

3.3. Architectural Framework

A conceptual architectural framework was created to examine the interactions between FL and cloud systems. Lastly, the framework illustrates the functions of key elements, including cloud servers, edge devices, and FL communication protocols. This design is based on existing literature, and it is then utilized to investigate implementation options and performance measures. Figure 2 denotes the FL architecture designed for multi-tenancy.

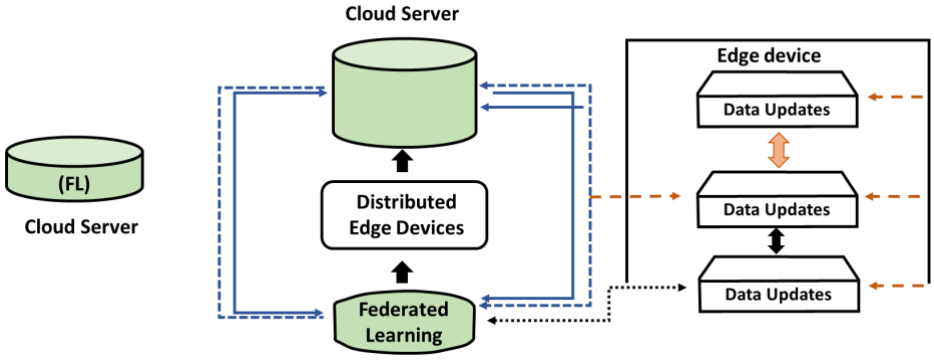

Figure 2. Federated learning (FL) architecture for multi-tenant data updates

FL's incorporation into cloud systems has been quite useful, enabling resource optimization and scalability. FL, in conjunction with cloud computing, enables the training of large-scale AI models on distributed systems. While model aggregation and orchestration on cloud platforms are completed effectively, the computational demands of FL are met. Recent research has taken advantage of cloud-based FL in sectors like healthcare, where it's critical to train medical AI models in a private, safe environment. For example, it has been shown that FL may be used to train prediction models over patient records that contain sensitive data without disclosing this data. This is despite the benefits of FL, which naturally has certain problematic difficulties that prevent it from being widely used, despite not having all the problems that plague the larger optimization field. One of the most difficult problems, sometimes referred to as non-IID (non-independent and identically distributed data), is the heterogeneity of data among devices. Furthermore, communication cost becomes a barrier due to the rapid interchange of model updates between devices and servers [15].

3.4. BERT-based Transfer Learning

To enable efficient knowledge sharing while preserving privacy in multi-tenant cloud environments, BERT-based transfer learning is employed. Pretrained BERT models serve as the foundation, allowing adaptation to tenant-specific tasks with minimal computational overhead. The approach mainly involves two distinct steps:

Pretraining: It undergoes training of the BERT model on a capacious unlabeled dataset, automatically absorbing typical linguistic patterns and structural representations.

Fine-tuning: Such pre-trained parameters have been further fine-tuned with tenant datasets, facilitating the model's adaptation towards differing domains while still exploiting knowledge accumulated from large-scale datasets. This transfer-learning boasts improved generalization capacity in resource-poor tenants. Figure 3 illustrates the architecture of the BERT model.

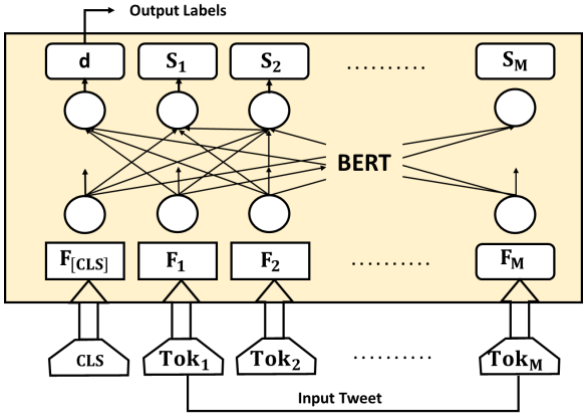

Figure 3. BERT model architecture

3.5. Pretrained BERT Variants for Domain-Specific Adaptation

To improve the adaptability of the different tenant domains, the following pre-trained BERT models are utilized according to the nature of the dataset:

- Google AI's BERT: Trained on BooksCorpus and English Wikipedia, it is suitable for general-purpose NLP tasks.
- SciBERT: Specialized in scientific text processing, pre-trained on Semantic Scholar papers.
- BioBERT: Optimized for biomedical text mining, pre-trained on PubMed and PMC corpora.

As mentioned above, the pre-trained models provide domain-specific information for transferring knowledge from one interested source to another, thus lessening the load of labeling resources while improving accuracy and efficiency in multi-tenant conditions.

3.6. Security and Scalability in Federated Learning

The implementation of FL with BERT-based transfer learning assures secure and scalable learning via multiple tenants. Key advantages include:

Privacy-Preserving Model Updates: Instead of sharing raw data, only model gradients are exchanged, ensuring compliance with data protection regulations while enabling cross-tenant knowledge sharing.

Computational Efficiency and Scalability: Using pre-trained BERT models reduces the need for extensive computational resources, making multi-tenant learning efficient and scalable.

Handling Data Heterogeneity: The fine-tuned models can effectively process heterogeneous data across different tenants, improving model adaptability and accuracy.

The integration of FL and transfer learning functions as an approach that establishes privacy-protecting, adaptable, resource-efficient knowledge sharing in decentralized cloud systems while handling issues with secure multi-tenant data management [16].

3.7. Tenant-Aware Security Mechanism

The tenant-aware security mechanism enables secure operation in multi-tenant systems that combine shared infrastructure with strict tenant isolation. The security framework exists in four main sections, which form its structure.

3.8. Data Isolation

Every tenant possesses their own Virtual Routing and Forwarding (VRF) table that provides total network separation. Virtual Extensible LAN (VXLAN) and Generic Network Virtualization Encapsulation (GENEVE) inspect protocols to establish virtual network partitions to achieve virtual networking. A virtualization solution that relies on hardware assistance provides separate encryption boundaries to each tenant while they are sharing resources. Every tenant that makes use of this infrastructure will have their own separately administered interfaces, configuration databases, and logging resources for enabling automation capabilities.

3.9. Hierarchical Access Model of Access Control Management

3.9.1. Managed Service Provider (MSP) Model

Global MSP administrators hold complete system access to all tenants for central management throughout the whole environment with this model. Configuration and security compliance maintenance are tasks assigned to the tenant manager in specific customer areas. Support engineers are granted limited temporary access to diagnose and solve issues quickly, but security comes first for them. Audit managers at the company are given permission to read only for purposes of compliance checking and carrying out security audits.

3.9.2. Tenant-Specific Administration

Each tenant has a tenant administrator to set their own security policies and configurations unconstrained by other tenants. Network & security administrators protect the infrastructure and monitor operational stability. Help desk staff have limited access to systems that allow them to monitor IT operations while preventing unauthorized changes.

3.10. Infrastructure Security

3.10.1. Shared Infrastructure Risks Mitigated By

There are many security measures to be put in place to reduce risks deriving from shared infrastructure. The user accesses DDoS protection features that utilize security layers that include traffic scrubbing capabilities, rate controls, and automatic blocking of malicious sources for securing operations. Micro-segmentation operates as a strategy for risk reduction through the placement of policy enforcement points across the infrastructure to stop unauthorized access at different security layers.

3.10.2. Encryption Protocols

A set of strong encryption measures exists to protect valuable data assets. An independent key management system operated by tenant-specific key management systems uses automated key rotation to deliver security integrity through separate encryption key management protocols for each tenant. Hardware security modules serve as secure platforms for key storage to protect cryptographic operations against possible security risks.

3.11. Threat Detection & Response

3.11.1. Security Monitoring

The security monitoring system makes use of tenant-specific correlation rules along with anomaly detection algorithms to recognize security incidents that apply to individual tenants. Precise threat detection, minimal false alerts, and unnecessary system warnings are achieved through this approach.

3.11.2. Security Automation

Security automation enables the maintenance of uniform security standards, which extend to every tenant. The automated processes facilitate fast deployment of security updates, which distribute them across the entire multi-tenant system to speed up vulnerability patching. System integrity benefits from security consistency, which effectively reduces potential vulnerabilities that cross between tenants [17].

3.12. Decentralized Aggregation Strategy

The traditional FL system implementation with central aggregation introduces the following two main vulnerabilities:

- The system becomes completely disrupted whenever the central server experiences failure or an attack because it functions as a single point of failure.
- Due to its complete access model updates, the central aggregator exposes the system to data leakage dangers.
- The performance capabilities of a single server are limited when it must handle many FL clients.

The Decentralized Trusted Aggregation (DeTA) system overcomes these issues through the secure distribution of aggregation duties between separate trusted aggregation nodes.

3.13. Secure Multi-Aggregator Design

Instead of a central server, DeTA uses multiple functional aggregators, each deployed within a Secure Encrypted Virtualization (SEV) Confidential Virtual Machine (CVM). These aggregators operate independently, ensuring that no single entity has full access to model updates. Key features include:

Randomized Model Partitioning: Each aggregator only handles a subset of model updates, preventing any single entity from reconstructing full updates.

Synchronization among Aggregators: The aggregators exchange encrypted updates periodically to ensure consistency of models while securing their contents.

End-to-End Secure Communication: Transport Layer Security (TLS) keeps encrypted data transmissions between FL parties and aggregators as updates travel from one device to another.

Trust Establishment via Two-Phase Authentication

Given that there is no central authority, DeTA works to instill trust across several aggregators using a two-phase authentication protocol:

Phase I: Trustworthy Aggregator Deployment

Each aggregator is containerized in SEV CVMs with Kata Containers for isolation.

During startup, an AMD Secure Processor generates an attestation report, verifying that the aggregator runs an untampered, integrity-checked software environment.

This report is sent to an Authentication Proxy (AP), controlled by FL participants, not by the aggregators themselves.

The AP verifies the aggregator’s certificate chain and firmware integrity before assigning a unique cryptographic authentication token.

Phase II: Multi-Aggregator Authentication

The FL client's challenge-response mechanism authenticates model aggregators before model updates are provided. In detail:

The FL client sends a random nonce to an aggregator.

The aggregator signs the nonce with its ECDSA authentication token.

Before receiving model updates, the client checks that the signature is valid to verify the authenticity of the aggregator [18].

4. RESULT AND DISCUSSION

A secure and multi-tenant data management framework provides capabilities for cross-domain knowledge sharing in decentralized cloud infrastructures. In the performance assessment, the BERT-Based Multi-Tenant Data Management System (MTDS) was investigated in contrast to existing models regarding several significant metrics. The research team assessed latency together with throughput efficiency and resource consumption to develop scalable and effective systems. It is the performance capacity of their entire proposed system in the multi-tenant cloud environment on which their entire research relies. In-depth evaluations by running their experiments across 20 virtualized cloud nodes on 8-core, 32-GB RAM vCPUs, and each host with 500 GB SSD storage.

4.1. Evaluation Metrics

The four main indicators of cloud computing performance measurement are effectiveness, resource utilization, compliance score, and latency, all of which evaluate system efficiency, security, and responsiveness. These metrics help in enhancing the system's performance by guaranteeing maximum resource allocation and compliance.

Effectiveness (%): It assesses how far a system achieves its goals with accuracy and reliability. Better performance prevails if the percentage approaches the maximum. The effectiveness measurement will follow the representation in Equation (1).

$$Effectiveness = \frac{SuccessfullOperation}{TotalOperation} \times 100 \quad (1)$$

Resource Utilization (%): The efficiency of a system in the way it uses resources is measured in terms of CPU, memory, and storage. The use of resources at their optimal levels improves system stability and decreases costs. Equation (2) represents the resource utilization calculation.

$$\text{Resource Utilization} = \frac{UsedResource}{TotalAvailableResource} \times 100 \quad (2)$$

Compliance Score (1-10 scale): This standard measures how well the system follows security regulations, including the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). The scoring system directly correlates with the level of patient system adherence. The system evaluates compliance through the calculation method presented in Equation (3).

$$\text{Compliance Score} = \frac{AchievedCompliancePoints}{MaximumPossiblePoints} \times 100 \quad (3)$$

Latency (milliseconds): The system requires this parameter to determine the duration it spends before reacting to user requests. Lower latency indicates better performance. Latency receives its calculation from Equation (4).

$$\text{Latency} = \frac{TotalResponseTime}{NumberofRequests} \times 100 \quad (4)$$

4.2. Performance Analysis

The effect of various consistency models on the average latency of the proposed BERT-Based MTDS was analyzed in this section. It contrasts latency under two scenarios: without consistency enforcement and with consistency enforcement, and measures the consistency overhead percentage, which is the additional latency caused by ensuring data consistency. The performance is evaluated concerning two popular consistency models, eventual consistency and strong consistency, to study their trade-offs concerning system performance. Table 2 represents the impact of consistency models on latency and overhead. Figure 4 illustrates the comparison of latency across different consistency models.

Table 2. Impact of consistency models on latency and overhead

Consistency Model	Avg-Latency without consistency (ms)	Avg-Latency with consistency (ms)	Consistency Overhead (%)
Eventual	12	14	16.6
Strong	12	18	50

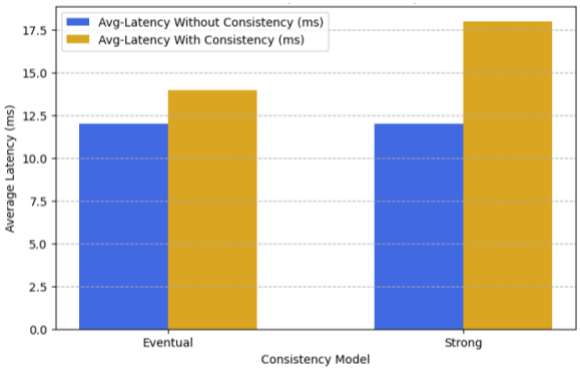

Figure 4. Consistency model vs latency comparison

The implementation of eventual consistency methods enables a system to achieve 16.6% latency while maintaining performance through delays from 12ms to 14ms that users can handle. The system performance suffers a major decline due to strong consistency since it produces a significant delay between 12 and 18 milliseconds at a 50% cost. Treatment systems with eventual consistency allow better performance by reducing data latency effects, while preserving strong consistency requires long response times for

upholding data accuracy. Cloud-based models are available for choice when determining multi-tenant cloud environments based on specific operational requirements that compare system performance to data protection needs.

4.3. Comparative Analysis

Examine the mean throughput and latency performance between SAMTS [19] and MTDS as proposed by this research at various cloud tenant scales. System efficiency can be determined through the measures of latency for how quickly the system responds, along with throughput, representing the number of requests the system completes per second. Table 3 illustrates the evaluation of latency and throughput for SAMTS and BERT-based MTDS. Figure 5 showcases the performance analysis of BERT-Based MTDS and SAMTS, highlighting (a) Latency vs. the Number of Tenants and (b) Throughput vs. Number of Tenants.

Table 3. Latency and throughput comparison of SAMTS and BERT-based MTDS

Number of Tenants	Average latency (ms)		Throughput (Requests/Sec)	
	SAMTS [19]	BERT-Based MTDS [Proposed]	SAMTS [19]	BERT-Based MTDS [Proposed]
100	12	10	10000	12000
500	15	13	48000	55000
1000	20	18	90000	100000
5000	28	25	430000	450000
10000	35	30	800000	850000

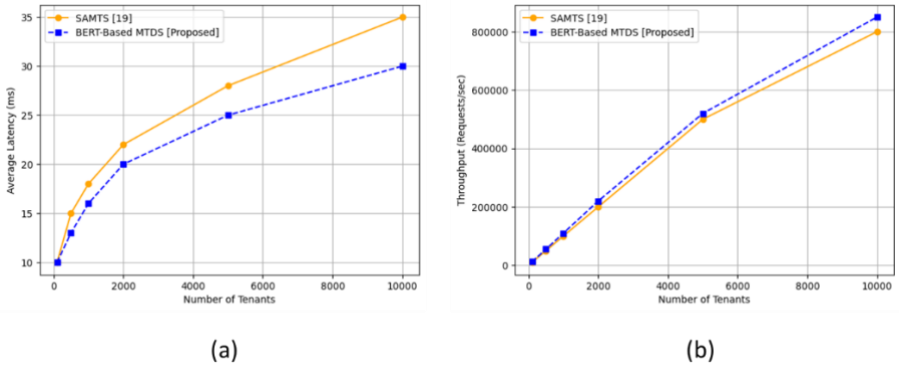

Figure 5. Performance comparison of BERT-based MTDS and SAMTS (a) latency vs. number of tenants (b) throughput vs. number of tenants

The research establishes that Bert-based MTDS delivers superior performance at all testing tenant levels through lower latency and higher throughput figures than SAMTS. The suggested system decreases average latency performance by 5ms more effectively than SAMTS, which leads to enhanced processing speed for multi-tenant workloads. The BERT-Based MTDS achieves substantially higher throughput (requests per second) because it reaches up to 50,000 requests per second when processing 10,000 tenants simultaneously. The proposed system outperforms traditional models by better handling growing demands because it demonstrates superior performance as tenant numbers increase. The BERT-Based MTDS demonstrates its value as an enhanced, optimized approach for multi-tenant cloud platforms by delivering improved responsiveness together with processing capability advancement.

A performance comparison exists between the Multi-Tenant Cloud Security (MTCS) [20] model and the proposed BERT-Based MTDS solution based on main evaluation criteria such as effectiveness, resource consumption, compliance score, and latency. Calculations reveal that the system being proposed is better than the older system based on various parameters like accuracy, as well as efficiency concerning rules and regulations, and reduced response time, making it appropriate for secure and scalable multi-tenant cloud deployments. The assessment of performance for MTCS with BERT-Based MTDS appears in Table 4. The performance comparison between MTCS and BERT-Based MTDS is demonstrated in Figure 6 through evaluation of (a) Effectiveness & Resource Utilization, (b) Compliance Score, and (c) Latency.

Table 4. Performance comparison of MTCS and BERT-based MTDS

Parameter	MTCS [20]	BERT-Based MTDS [Proposed]
Effectiveness (%)	95	97
Resource Utilization (%)	78	83
Compliance Score (1-10 scale)	9.2	9.5
Latency (millisecond)	8ms	6ms

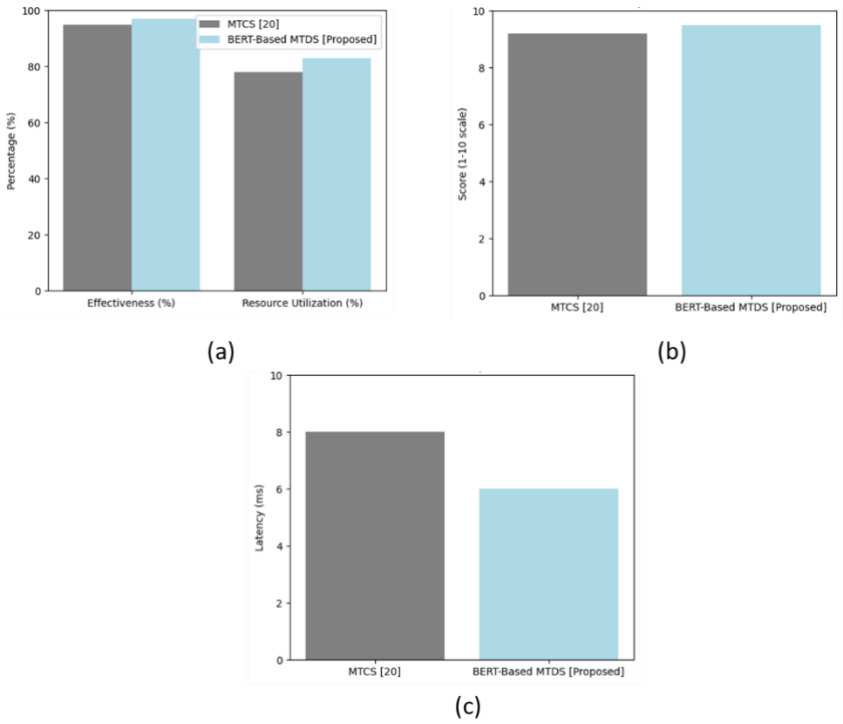

Figure 6. Performance comparison of BERT-based MTDS and MTCS (a) effectiveness and resource utilization (b) compliance score (c) latency

A comparative analysis of major performance parameters between the suggested BERT-Based MTDS and the MTCS model [20]. The BERT-Based MTDS produces enhanced results across all metrics, which indicates greater performance quality together with better security and efficiency. The efficiency rating improves from 95% to 97%, which indicates that the multi-tenant data management system operates with higher precision. The suggested model optimizes resource management by improving its usage level from 78% to 83% within the system. The compliance score improved 9.5 points from an initial 9.2 points, indicating better compliance with the provisions of the GDPR and HIPAA. The proposed system operates at a latency of 6ms, which is 2ms better than the currently adopted MTCS, thus offering better responsiveness and operational efficiency. Through the results presented, the BERT-Based MTDS shows clear indications that it provides a highly secure, scalable, and efficient approach to multi-tenant cloud environments.

4.4. Discussion

An efficient and secure data management system for multi-tenancy, which uses FTL to operate within a decentralized cloud infrastructure, was introduced. Many challenges reduce the effectiveness of existing approaches targeting multi-tenant cloud infrastructure. The performance inconsistencies and scalability issues that are compounded by the operational overhead in SAMTS would result in such differences being aggravated towards poor performance in data dissemination and load-balancing activities. The inability to effectively guarantee the separation of tenants is what turns into security issues within the systems. The high cost of managing the security measures in MTCS models at the same time outlaws performance deterioration and regulatory compliance, even if security is better guaranteed through measures of encryption and isolation. The proposed BERT-Based MTDS gives solutions to these inherited problems. As the system utilizes reduced latency, which creates better responses of the system when the load is higher,

it enables the development of much superior speed and flexibility. The advantages of resource management will help in improving the computational functioning rates. The BERT-Based MTDS security upgrade improves standards for tenant protection by increasing compliance scores in MTCS while keeping latency levels low to maintain performance security balance. The solution proposed here, KBERT-Based MTDS, can realize a scalable, efficient, secure multi-tenant cloud architecture according to its key implementation constraints.

5. CONCLUSION

The system designs a multi-tenant data management system that retains properties of security and privacy alongside efficiency in a decentralized cloud infrastructure. The proposed BERT-Based Research System could incorporate FTL in supporting collaborative learning while maintaining tenant privacy per the design specifications. The application of BERT-based transfer learning served to advance knowledge exchange by performing model fine-tuning for particular tenant applications, which improved both performance efficiency and cut down computational requirements. A tenant-aware security system dynamically evaluated trust levels to guarantee secure workload assignment, while a decentralized aggregation strategy improved data privacy and system resilience through the avoidance of single-point failures. Experimental testing validated low latency (6ms), high throughput (850,000 requests/sec, 10,000 tenants), efficient use of resources (83%), and strict security compliance (9.5/10). Consistency models also imposed a 50% overhead for strict consistency, documenting the performance data integrity trade-offs. While the proposed system effectively improved scalability, security, and efficiency, ensuring strong consistency led to some performance overhead. However, it was a trade-off that required data integrity in multi-tenancy. For future research, AI-based security features can be explored to further discover threats and react to them, and adaptive models of consistency can be set up to dynamically balance performance and data integrity. Expanding the system to support heterogeneous cloud infrastructures would further increase its usefulness. These findings confirmed that the BERT-Based MTDS provided an efficient, scalable, and privacy-preserving solution for multi-tenant cloud systems, paving the way for further innovations in secure and decentralized cloud computing.

DATA AVAILABILITY STATEMENT

The data presented in this study are available on request from the corresponding author.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest to this work.

REFERENCES

[1] S. Deochake and V. Channapattan, “Identity and Access Management Framework for Multi-tenant Resources in Hybrid Cloud Computing,” in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, Aug. 2022, pp. 1–8, https://doi.org/10.1145/3538969.3544896.

[2] A. Bhargava, D. Bhargava, P. N. Kumar, G. S. Sajja, and S. Ray, “Industrial IoT and AI implementation in vehicular logistics and supply chain management for vehicle mediated transportation systems,” *International Journal of System Assurance Engineering and Management*, vol. 13, no. S1, pp. 673–680, Mar. 2022, https://doi.org/10.1007/s13198-021-01581-2.

[3] S. Saxena, D. Yagyasen, C. N. Saranya, R. S. K. Boddu, A. K. Sharma, and S. K. Gupta, “Hybrid Cloud Computing for Data Security System,” in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Oct. 2021, pp. 1–8, https://doi.org/10.1109/ICAECA52838.2021.9675493.

[4] A. M. Mostafa *et al.*, “Decentralized Identity Management in Cloud Computing: A Blockchain-Based Solution With Automatic Provisioning Techniques,” *International Journal of Intelligent Systems*, vol. 2025, no. 1, Jan. 2025, https://doi.org/10.1155/int/2969737.

[5] Y. Zhang, X. Chang, and X. Liu, “Inference of gene regulatory networks using pseudo-time series data,” *Bioinformatics*, vol. 37, no. 16, pp. 2423–2431, Aug. 2021, https://doi.org/10.1093/bioinformatics/btab099.

[6] U. S. Basha, “Fortifying Healthcare Data Security in the Cloud: A Comprehensive Examination of the EPM-KEA Encryption Protocol,” *Computers, Materials & Continua*, vol. 79, no. 2, pp. 3397–3416, 2024, https://doi.org/10.32604/cmc.2024.046265.

[7] N. Manikandan, P. Thejasree, K. E. K. Vimal, K. Sivakumar, and J. Kiruthika, “Applications of Artificial Intelligence Tools in Advanced Manufacturing,” 2024, pp. 29–42.

[8] M. K. Horton, S. Dwaraknath, and K. A. Persson, “Promises and perils of computational materials databases,” *Nature Computational Science*, vol. 1, no. 1, pp. 3–5, Jan. 2021, https://doi.org/10.1038/s43588-020-00016-5.

[9] S. K. Gupta, A. Alemran, C. P. Ranjith, and M. S. K. Mohideen, “Biometric Authentication for Healthcare Data Security in Cloud Computing—A Machine Learning Approach,” in *Advancements in Science and Technology for Healthcare, Agriculture, and Environmental Sustainability*, London: CRC Press, 2024, pp. 318–324.

[10] R. Natarajan, G. H. Lokesh, F. Flammini, A. Premkumar, V. K. Venkatesan, and S. K. Gupta, “A Novel

Framework on Security and Energy Enhancement Based on Internet of Medical Things for Healthcare 5.0," *Infrastructures*, vol. 8, no. 2, p. 22, Feb. 2023, https://doi.org/10.3390/infrastructures8020022.

[11] V. P. Nzanzu *et al.*, "FEDARGOS-V1: A Monitoring Architecture for Federated Cloud Computing Infrastructures," *IEEE Access*, vol. 10, pp. 133557–133573, 2022, https://doi.org/10.1109/ACCESS.2022.3231622.

[12] E. A. Adeniyi, R. O. Ogundokun, S. Misra, J. B. Awotunde, and K. M. Abiodun, "Enhanced Security and Privacy Issue in Multi-Tenant Environment of Green Computing Using Blockchain Technology," 2022, pp. 65–83.

[13] J. M. Parra-Ullauri *et al.*, "Federated Analytics for 6G Networks: Applications, Challenges, and Opportunities," *IEEE Network*, vol. 38, no. 2, pp. 9–17, Mar. 2024, https://doi.org/10.1109/MNET.2024.3355218.

[14] B. Han *et al.*, "PBFL: A Privacy-Preserving Blockchain-Based Federated Learning Framework With Homomorphic Encryption and Single Masking," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 14229–14243, May 2025, https://doi.org/10.1109/JIOT.2024.3524632.

[15] Naveen Kodakandla, "Federated learning in cloud environments: Enhancing data privacy and AI model training across distributed systems," *International Journal of Science and Research Archive*, vol. 5, no. 2, pp. 347–356, Apr. 2022, https://doi.org/10.30574/ijsra.2022.5.2.0059.

[16] A. Agrawal, S. Tripathi, M. Vardhan, V. Sihag, G. Choudhary, and N. Dragoni, "BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling," *Applied Sciences*, vol. 12, no. 3, p. 976, Jan. 2022, https://doi.org/10.3390/app12030976.

[17] R. Ibrahim, I. Khider, S. Edam, and T. Mukhtar, "Comprehensive Strategies for Enhancing SD-WAN: Integrating Security, Dynamic Routing and Quality of Service Management," *IET Networks*, vol. 14, no. 1, Jan. 2025, https://doi.org/10.1049/ntw2.70007.

[18] P.-C. Cheng *et al.*, "DeTA: Minimizing Data Leaks in Federated Learning via Decentralized and Trustworthy Aggregation," in *Proceedings of the Nineteenth European Conference on Computer Systems*, Apr. 2024, pp. 219–235, https://doi.org/10.1145/3627703.3650082.

[19] K. Reddy, "Scalable Data Management in Distributed Systems: A Sharding-Based Approach for Multi-Tenant Architectures," *International Journal of Emerging Research in Engineering and Technology*, vol. 5, pp. 1–12, 2024, https://doi.org/10.63282/3050-922X.IJERET-V5I3P101.

[20] W. Hashim and N. A.-H. K. Hussein, "Securing Cloud Computing Environments: An Analysis of Multi-Tenancy Vulnerabilities and Countermeasures," *SHIFRA*, vol. 2024, pp. 8–16, Feb. 2024, https://doi.org/10.70470/SHIFRA/2024/002.

BIOGRAPHIES OF AUTHORS

Aditya Gupta is working as a Research associate in computer science and engineering at Lincoln University College, Malaysia, and as a Senior Software Developer in a reputed organization in Lucknow, UP, India. He can be contacted at email: gupta.aditya56@gmail.com

Sai Kiran Oruganti is a professor specializing in wireless power transfer and IoT security, currently affiliated with Jiangxi University of Science and Technology in China and Lincoln University College in Malaysia. His career includes research at the Ulsan National Institute of Science and Technology, developing wireless systems for Hyundai and Samsung, a faculty position at the Indian Institute of Technology, and a recognized history of innovation, including pioneering work on Zenneck Wave WPT and over 16 granted patents. He can be contacted at email: saisharma@lincoln.edu.my