Multi-Level Multiple Security for CSP, Tenant & User via Edge System in Cloud

B.Angelm Rubavathy[1], Rebecca Jeyavadhanam Balasundaram[2], S. Albert Antony Raj[3]

[1]Dept, of Comp Applications, CSH, SRMIST, Kattankulathur – 603403, India.
[2]Dept, of Comp Science, York St John University London, UK.
[3]Dept, of Comp Applications, CSH, SRMIST, Kattankulathur – 603403, India.

Received: 10.07.2024 Revised: 17.08.2024 Accepted: 14.09.2024

ABSTRACT

In today's era there are speed developments in Cloud Service Provider (CSP) and data allocation. The major concern in data allocation is inchoosing an accurate storage for the user data and allocating the data along with enhanced security. A proper resource allocation strategy should achieve elasticity and provide a high-levelsecurity to the user information. The system deliberates an 'Optimized Allocation Strategy (OAS)for data allocation. The designed OAS algorithm provides an optimized Analysis over choosingsuitable service provider.Service providers allocate storage that scrutinizes the details. Further, the system leverages a Benefit calculator that performs a mutual analysis over cloud server, Edge System, tenant and client. The system incorporates a multi-level security framework employing various cryptographic algorithms to bolster overall security. Empirical findings substantiate for cloud service providers and diverse user implementation with allocation strategies that the optimization algorithm proposed herein yields advantageous outcomes. Notably, the results indicate a notable improvement of over 50% in scheduling speed and security enhancement compared to extant models. The utilization of a multilevel security approach further amplifies the efficacy of the proposed model in contrast to its predecessors, affirming a heightened level of security.

Keywords: Optimized Scheduling (OS) Model, Tenant, Edge System, CSP, Profit/Loss Calculator and User

1. INTRODUCTION

Advanced developing on technology and popularization, choosing a suitable resource allocation in cloud is being a major challenging task. Edge computing is the main supporting technology used for both CSP and Client. Edge computing represents a decentralized model offering storage solutions for user data. Efficient data storage, coupled with robust security measures, emerges as a critical consideration in cloud computing. Cloud service providers typically fall into categories such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), adopting a pay-as-you-go model for end-users.[1]. In Edge computing the firms' applications are brought closer to local edge servers. This immediacy to data can strengthen business benefits, including improvised response times and higher bandwidth availability. To improvise the yields of cloud service the cost of energy should the reduced towards the point. Hence, for each and every access the main servers can't be used here the edge systems and tenants are used for storage [2]. In edge systems a local server is maintained for accessing all the user information. All the user information is fetched arranged and allocated later to the main server.

The Resource optimized scheduling technology has attracted wide attention of many experts and scholars. Moreover, to produce an effective scheduling method Edge system, tenants and even data centres are utilized. There are many cloud based services like Amazon, Google and MS containing distributed for universal containers and have ability to require for providing workflow for resources and user data [3].Generally, the access to the user is deliberated with pay-per-use bill model for any kind of data storage purpose. It is more important that the arriving data should be placed and maintained in an appropriate container. It attains the required state of the models for properly scheduled theory ensures the cost effective resource utilization [4] [5].

Many resource allocation strategies and algorithms are used previously which are not up to-the-mark. An effective Benefit/loss calculator that analyses the cloud service provider and allocates user data accordingly is projected. The Benefit/loss calculator method can improve the viability and efficiency of the resource allocation by helping the user in choosing an appropriate cost-effective storage in cloud.

The subsequent sections of this manuscript are structured as follows. In Section 2, an in-depth exploration of preceding methodologies is provided, presented in the form of an extensive literature survey. Section 3 outlines the methodologies utilized and offers solutions to challenges encountered during the allocation process. Moving on to Section 4, the paper examines strategies relevant to the Cloud Service Provider (CSP), tenants, edge system, and users. This section elaborates on the core optimization theory and introduces the proposed multi-level multiple security implementations. Section 5 is devoted to validating the experimental analysis, demonstrating the effectiveness of the proposed algorithm through comparisons with existing models. Lastly, Section 6 concludes the paper with a summarization of key findings.

2. LITERATURE REVIEW

Numerous allocation techniques have been explored by researchers, with a focus on optimization methodologies within data centres documented in the existing literature. The following section critically reviews current studies on Resource Allocation and security in the Cloud. A refined process incorporating Semi-Supervised Learning and Ensemble Transformations, integrating prediction consistency and confidential optimization concepts, is detailed in [6]. Subsequently, the paper outlines inherent limitations prompting a need for system enhancements in allocation performance and job scheduling metrics. The antecedent model delves into resource allocation challenges and issues related to optimizing data scheduling. Zhang et al. [7] introduced a resource allocation model that does not take into account arrival and departure times, limiting applications to a specific timeframe. Guan et al. [8] expanded their investigation beyond energy consumption, integrating the expense associated with information exchange between containers.

2.1 Discussion on Previous Optimization Algorithm

There is several optimization techniques used to allocate the client data is right place. Some of the previous scheduling techniques are Multi-objective optimization, Ant-colony optimization, self-adaptive method and PSO based Scheduling approach etc.., B. Tan et al. in [8] to resolve the multi-object optimization problem the author established a multi-objective NSGA-II model for more efficiency. Considering the energy consumption and application availability the cloud edge system requirement are validated and acknowledged to the main server atonce. To address the challenge of resource allocation container (RAC), a Genetic Programming hyper-heuristic model is proposed [10]. In addressing sporadic job scheduling within the Dynamic Voltage and Frequency Scaling (DVS) domain, Mei et al. [11] present an energy-aware scheduling concept. Additionally, Kai Huang et al. [12] introduce a self-adapting job scheduling scheme, imbuing the system with adaptability and employing dynamic priority scheduling for the allocation and management of resources.L. Yin et al. in [14] produced an optimized job Scheduling theory in fog computing. Further the author provides a resource Allocation and elasticity management relating Containers for Manufacturing [15].

Table 1. Comparing Cloud resource Scheduling Algorithm

REFERENCES	Method Implementation	Merits	Demerits
[8]	Micro-service scheduling in ant colony algorithm	Multi-objective optimization	Possess increased time complexity Very Slow Convergence
[10]	Scheduling based on genetic algorithm	Contains global search ability Uses collaborative optimization	High time High cost Slow Convergence
[12]	Resource allocation using Self-adaptive technique	Adaptive scheduling Low network delay	Single goal optimization
[17]	Locality aware scheduling model	Load balance and ability maintenance	Falls easily into local optimum

2.2 Security Challenges and Algorithms Used

Cloud servers (CS) may not be consistently reliable due to the public administration's limited control. Attacks frequently target transaction and execution periods, leaving the uploaded sensitive data vulnerable to unauthorized access or theft. This vulnerability extends to potential threats from malicious users, attackers, or even the cloud server providers (CSP) [14]. Consequently, the imperative to secure every piece of data within cloud server environments has gained increased significance [15].Zhu et al.

[16] tackle certificate management concerns and propose a searchable encryption scheme for proof identification. Chen et al. [17] discuss a cryptographic scheme related to a certificate-less crypto-system, utilizing a random oracle model to enhance resilience against keyword attacks. Some researchers explore the application of lightweight techniques, such as Hadamard transforms [18], to enhance data security. Additionally, endeavors have been made to address issues related to trust, privacy protection, access control, and searchable encryption in the cloud. [19-22].

3. Method Implementation

The most formidable aspect of working in the cloud involves managing and allocating user resources. Allocating resources allows the provider to maximize profits and determine the extent to which resources are assigned. To facilitate this, the system employs an Optimized Allocation Strategy (OAS), where the scheduler's main focus is formulating decisions related to allocation and transactions. This process takes into consideration the current state of the cloud infrastructure. The subsequent diagram outlines a comprehensive procedure for the proposed secured optimization model.Generally, data allocation needs enormous space for placing the user data. The system needs multiple servers and service providers to implement a proper allocation. To compensate the need of servers the system uses edge system as tenants. Data centre acts as a storage unit to store large amount of user data. If any data centre owns the server then that server can be used for elastic storage.

3.1 Edge System for Resource Allocation

As the main servers cannot be used for all access and allocation purpose the edge systems are used. Here, the edge systems are used as a local server for assigning and scheduling purpose. In edge system the incoming data are collected and set for optimizer check.

Optimization Model

The optimization is the main concept in this model where the optimizer performs optimizing and arranges the container according to the need of user. The optimizer works in a way when a space is freed up the space is allocated to the user waiting next. Likewise, the optimization executed and resources are allocated.

Benefit /Loss Calculator

The system during allocation must concentrates on the benefit/loss it gains. To calculate the benefit/loss the system all the transaction and storage are well-monitored and recommended to the system. All the CSP, edge system and tenant attain benefit via benefit/loss calculator.

From the below Figure 1 the data entry and exit are validated and noted to analyse and structure a reasonable storage space allocation of edge system to the tenant. The CSP, tenant and client will be under a contract where they should be convinced with the pricing. If not convinced then they can withdraw from the system implementation. If the tenants are satisfied with the cost, then the tenant can prolong the data storage by choosing Elastic Processing.

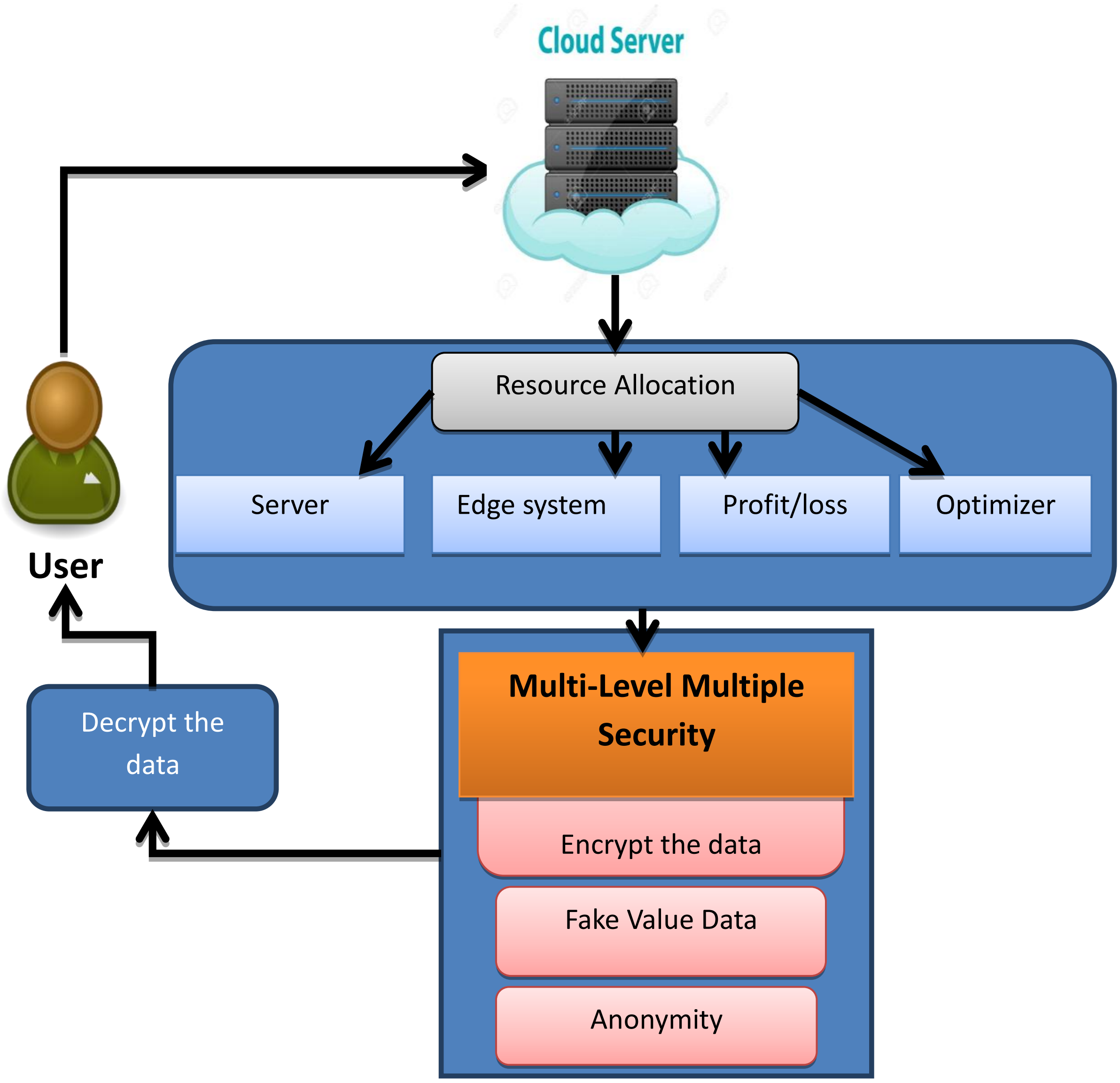

Figure 1. System Architecture for Multi Level

The cloud service provider choosing by tenant and user are performed by checking the ratings listed form high to low. If high rating provided then the edge system is locked for storage. By implementing this model, the proposal provides an improvised optimization model with benefit prediction that coordinates resource allocation in a right path. A classification classifying user and tenant is implemented. All the edge systems are checked and prepared for user data allocation. Then a benefit/loss calculator set to monitor transaction and intimate the CSP about the benefit and loss occurrence. The system uses an optimizer that optimizes the allocation order wise calculating the free and occupied space. To enhance the security a multi-level multiple security indulging various algorithm (ECC and Cuckoo Hashing) for encryption are produced.

3.2 Multi-Level Multiple Security

As the user data undergoes several systems there isno guarantee that the data is secured.In cloud security has always been a challenging task.To address security concerns and augment overall security measures, the system introduces an innovative concept termed the Multi-level Multiple Security Theory. This framework incorporates multiple key generation algorithms and employs various learning processes. Within this scheme, the cloud incorporates essential security algorithms specifically tailored for live streaming data transactions.The system uses anelliptic curve cryptographic (ECC) and Cuckoo hashing in shuffle basis for encryption and decryption. This implementation makes the process tedious for the attackers to hack the data.

Elliptic Curve Cryptographic (ECC)

Ensuring security poses a formidable challenge in stream processing and various social network operations. Achieving secure communication among tenants, Cloud Service Providers (CSP), and users necessitates the implementation of a sophisticated security algorithm [10-12]. The system prioritizes the utilization of a high-level public-key cryptographic algorithm known as 'Elliptic Curve Cryptography (ECC)' for the encryption and decryption of data. ECC, based on algebraic representations of elliptical curves, employs finite field cryptography, enabling the management of shorter bit keys while maintaining an equivalent level of security.

Cuckoo Hashing Technique

The Cuckoo hashing technique is an improved hashing method that incorporates two or more hashing attributes to address collisions inherent in hashing. This technique introduces a load-bearing mechanism and facilitates efficient queries with constant time complexity even in challenging scenarios. In the proposed approach, the Cuckoo hash technique is employed for data occupancy, suggesting available positions for client data.

3.2.1 Fake Value Data Implementation

This implementation is carried when edge systems are used. After using the edge system even if the informationis flushed it can be retrieved easily. To avoid such circumstances, the system uses fake value data.The data in the edge system are allocated in the main server. After loading process, the data are deleted and instead a Fake Value data is created and again deleted.In the event of a data breach recovery attempt, the individual would only obtain a fabricated dataset rather than the original, ensuring the maintenance of a robust high-level security protocol.

3.2.2 Anonymity Maintenance

At this stage, the system stores comprehensive user and anchor personal information. Stringent data management practices are implemented to ensure the secure handling of all records, transactions, and data sharing.

The algorithm for the proposed model is outlined in Algorithm 1.

Algorithm 1. Pseudocode for the CSP

```
Initialize a Content Storage Platform (CSP) to store data.
Function checkForFreeServer():
  if there are free servers:
    return available_server
  else:
    return NULL
Function allocateServerToUser(user):
  server = checkForFreeServer()
  if server is not NULL:
    allocate server to user
  else:
useEdgeSystemsForStorage(user)
Function useEdgeSystemsForStorage(user):
  use blockchain approach to store data on edge systems
Function encryptDataWithECC(data):
encrypted_data = ECC_encrypt(data)
  return encrypted_data
Function storeData(user, encrypted_data):
server_or_edge_system = allocateServerToUser(user)
  if server_or_edge_system is not NULL:
    store encrypted_data in server_or_edge_system
Function cuckooHash(data):
hash_value = perform_cuckoo_hash(data)
  return hash_value
Function verifyDataIntegrity(data, hash_value):
  if cuckooHash(data) equals hash_value:
    return true
  else:
    return false
Function retrieveAndDecryptData(user):
  data = retrieveData(user)
  if verifyDataIntegrity(data, stored_hash_value):
decrypted_data = ECC_decrypt(data)
    return decrypted_data
  else:
    return "Data integrity check failed."
Function allocateEdgeSystem(user):
  allocate edge system dynamically based on user ratings
Function createAndDeleteFakeData():
  create fake data
  delete fake data
Function maintainAnonymity():
  anonymize user and data throughout the process
Main Process:
  user = requestUserData()
allocate_edge_system(user)
createAndDeleteFakeData()
storeData(user, encryptDataWithECC(user_data))
notifyUser("Data stored successfully.")
retrieveAndDecryptData(user)
maintainAnonymity()
notifyUser("Data retrieved securely.")
```

4. RESULT ANALYSIS

In the analysis of results, a comprehensive comparison is conducted between the proposed model and existing algorithms. The comparison includes established algorithms such as Artificial Neural Network (ANN) and Naïve Bayes classifier, juxtaposed with the proposed model titled 'Advanced Multi-Level Multiple Security.' The proposed models exhibit heightened sustainability and efficacy in edge system selection and user data security. This evaluation is based on specific parameters.

Figure 2 illustrates the comparative analysis of scheduling speed between existing models and the proposed model. Figure 3 presents a comparative overview of security enhancement between existing models and the proposed model. Furthermore, Figure 4 undertakes a comparison of load balancing, while Figure 5 scrutinizes the benefit calculator, providing a comprehensive assessment of the proposed model against existing models.

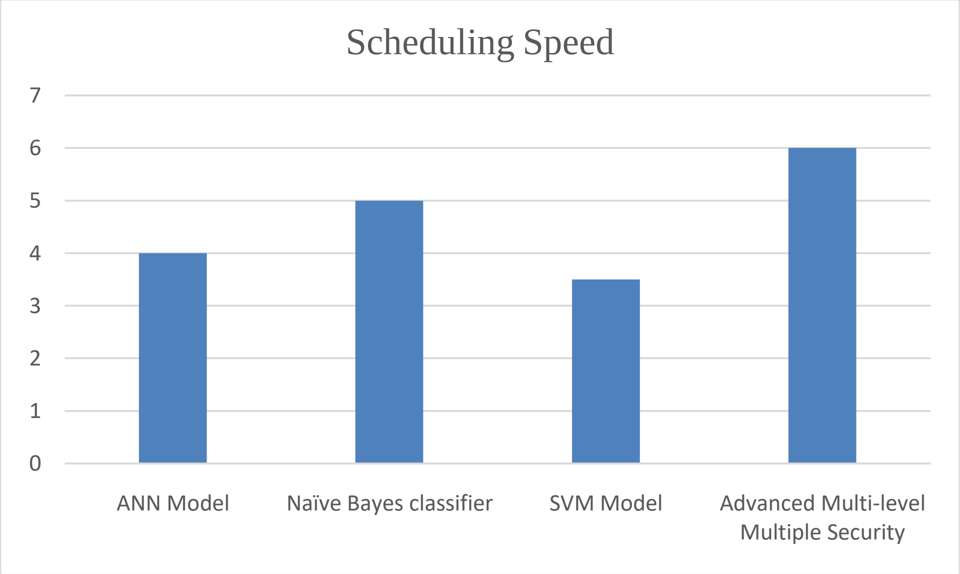

Figure 2. Comparison of existing models with proposed model with respect to scheduling speed

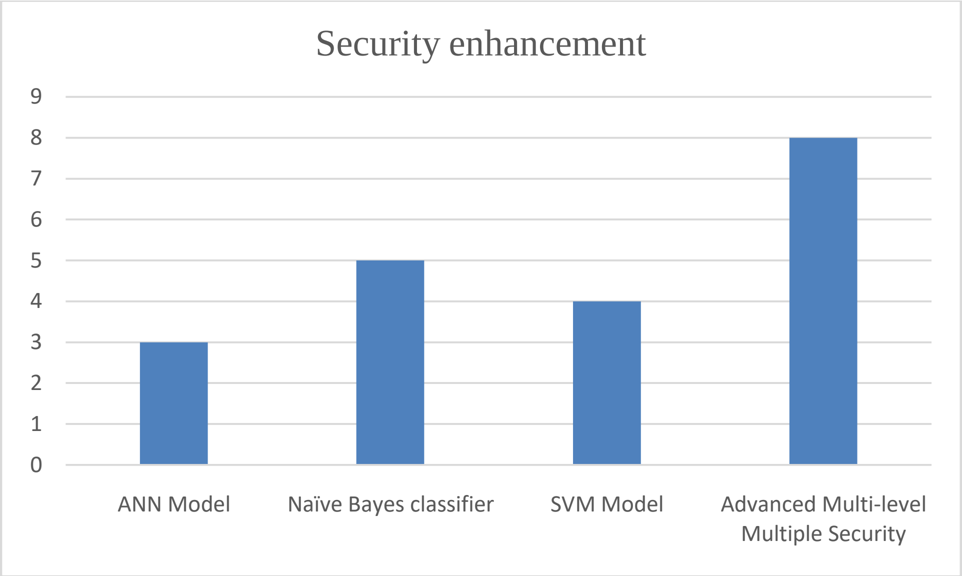

Figure 3. Comparison Of Existing Models Vs Proposed Model With Respect To Security Enhancement

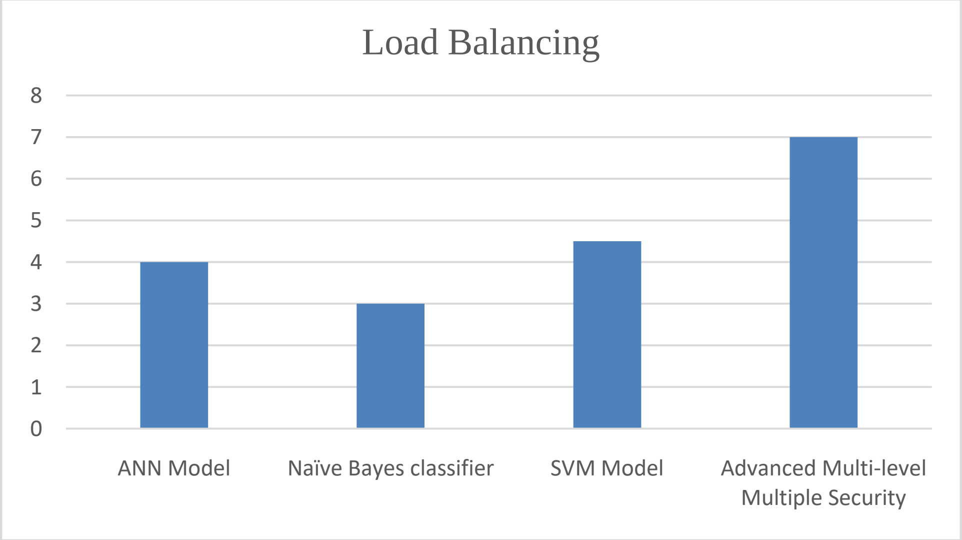

Figure 4. Comparison of existing models with proposed model with respect to load balancing

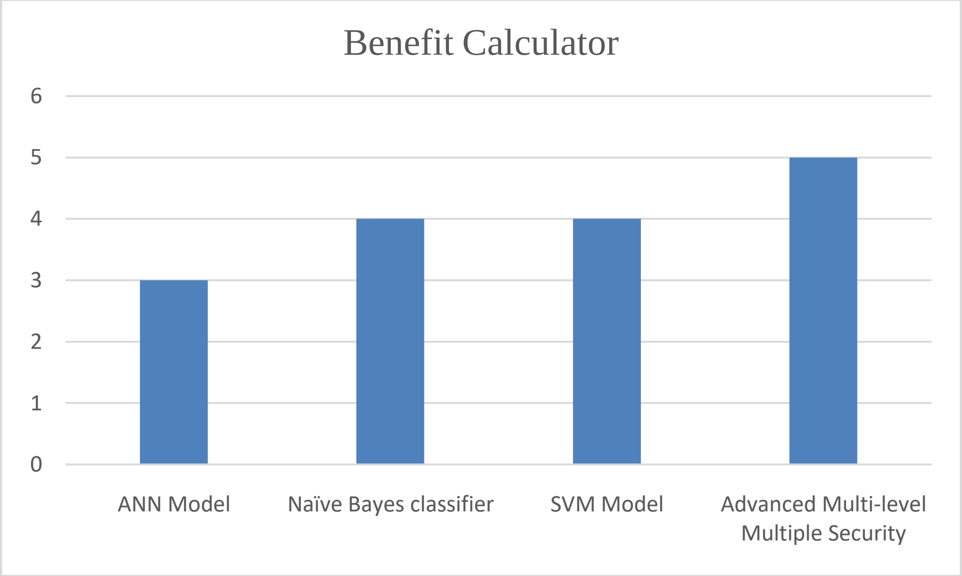

Figure 5. Comparison of existing models with proposed model with respect to benefit calculator

5. CONCLUSION

As a conclusion proposed system focuses on an optimized resource allocation for user data and implements anOptimized Allocation strategy (OAS) for efficient allocation. A benefit/loss analyzerproduced to analyzethe profit or loss graph occur while transaction. To enhance the security while allocation a Multi-level Multiple security mechanism that uses ECC and Cuckoo Hashing Algorithm for encrypting is implemented. The proposed implementation model executed successfully to confuse the attackers. Hence, from the proposal implementation the system allocation and security level is made more effective. In future the block allocation may be used for easy and enhanced retrieval of data. The security level implementation in real-time application is focussed.

REFERENCES

[1] D. Chemodanov, P. Calyam, S. Valluripally, H. Trinh, J. Patman and K. Palaniappan, "On QoE-Oriented Cloud Service Orchestration for Application Providers," in IEEE Transactions on Services Computing, vol. 14, no. 4, pp. 1194-1208, 1 July-Aug. 2021, doi: 10.1109/TSC.2018.2866851.

[2] X. Fu, Q. Shen, W. Wang, H. Hou and X. Gao, "Slice Merging/Spliting Operations and Tenant Profit Optimization Across 5G Base Stations," in IEEE Access, vol. 9, pp. 9706-9718, 2021, doi: 10.1109/ACCESS.2020.3048062.

[3] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," IEEE Transactions on Services Computing, 2015, doi: 10.1109/TSC.2015.2466545.

[4] R. Tang, ``Research on resources scheduling strategy of container cloud platform based on Kubernetes,'' M.S. thesis, Dept. Elect. Eng., UESTC Univ., Si Chuan, China, 2017.

[5] A. C. Zhou, B. He, X. Cheng, and C. T. Lau, "A declarative optimization engine for resource provisioning of scientific workflows in geo-distributed clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 3, pp. 647–661, 2016.

[6] X. Wang, D. Kihara, J. Luo and G. -J. Qi, "EnAET: A Self-Trained Framework for Semi-Supervised and Supervised Learning With Ensemble Transformations," in IEEE Transactions on Image Processing, vol. 30, pp. 1639-1647, 2021, doi: 10.1109/TIP.2020.3044220.

[7] X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, and R. Buyya, "Energy-aware virtual machine allocation for cloud with resource reservation," Journal of Systems and Software, vol. 147, pp. 147–161, 2019.

[8] X. Guan, X. Wan, B. Y. Choi, S. Song, and J. Zhu, "Application Oriented Dynamic Resource Allocation for Data Centers Using Docker Containers," IEEE Communications Letters, vol. 21, no. 3, pp. 504–507, 2017.

[9] M. Niu, B. Cheng, Y. Feng and J. Chen, "GMTA: A Geo-Aware Multi-Agent Task Allocation Approach for Scientific Workflows in Container-Based Cloud," in IEEE Transactions on Network and Service Management, vol. 17, no. 3, pp. 1568-1581, Sept. 2020, doi: 10.1109/TNSM.2020.2996304.

[10] B. Tan, H. Ma, Y. Mei and M. Zhang, "A Cooperative Coevolution Genetic Programming Hyper-Heuristic Approach for On-line Resource Allocation in Container-based Clouds," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2020.3026338.

[11] J. Mei, K. Li, J. Hu, S. Yin, and E. H.-M. Sha, "Energy-aware preemptive scheduling algorithm for sporadic tasks on {DVS} platform," Microprocessors and Microsystems, vol. 37, no. 1, pp. 99 – 112, 2013.

[12] L. Zhu, K. Huang, Y. Hu and X. Tai, "A Self-Adapting Task Scheduling Algorithm for Container Cloud Using Learning Automata," in IEEE Access, vol. 9, pp. 81236-81252, 2021, doi: 10.1109/ACCESS.2021.3078773.

[13] R. Zhou, Z. Li and C. Wu, "Scheduling Frameworks for Cloud Container Services," in IEEE/ACM Transactions on Networking, vol. 26, no. 1, pp. 436-450, Feb. 2018, doi: 10.1109/TNET.2017.2781200.

[14] D. Alsmadi and V. Prybutok, ``Sharing and storage behavior via cloud computing: Security and privacy in research and practice,'' Comput. Hum. Behav., vol. 85, pp. 218_226, Aug. 2018.

[15] Y. Li and C. G. Ma, ``Review of research progress on searchable encryption,'' J. Netw. Inf. Secur., vol. 4, no. 7, pp. 13_21, 2018. elastic leaks,'' J. Huaibei Normal Univ., vol. 40, no. 1, pp. 19_25, 2019.

[16] M. Zhu, Y. Chen, and Y. Hu, ``Identity-based searchable encryption scheme supporting proxy re-encryption,'' Comput. Eng., vol. 45, no. 1, pp. 129_135, 2019.

[17] T. Y.Wu, C. M. Chen, K. H.Wang, C. Meng, and E. K.Wang, ``A provably secure certificateless public key encryption with keyword search,'' J. Chin. Inst. Eng., vol. 42, no. 1, pp. 20_28, 2019.

[18] Srisakthi, S., & Shanthi, A. P. (2018). Design of a Secure Encryption Model (SEM) for Cloud Data Storage Using Hadamard Transforms. Wireless Personal Communications: An International Journal, 100(4), 1727-1741.

[19] Sodhro, A. H., Pirbhulal, S., Muzammal, M., &Zongwei, L. (2020). Towards blockchain-enabled security technique for industrial internet of things based decentralized applications. Journal of Grid Computing, 18, 615-628.

[20] P. Jayavel, V. Ramasamy, N. Amaladoss, V. Renganathan, and V.I. Shupeniuk, Chemical Physics Impact **8**, 100476 (2024).

[21] Pitchai P., Nepolraj A., Sathiyaseelan M., Gengen R.M., 4-Dihydroxy-3-(Indol-2-)-Yl-Quinoline via Substantial Methodology-Fisher Indole Synthesis, Heterocyclic Letters, 16 (1): 11 (2016).

[22] Nepolraj A., Pitchai P., Vijayarathinam M., A Facile Synthesis of 2-Hydroxy-3-Phenylquinoline Derivatives, Malaysian Journal of Chemistry, 21(3): 66 (2019).