International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(15s) 2026 pp. 604-622
https://doi.org/10.70917/ijcisim-2026-4440
Article

RFGB-Hybrid Machine Learning Framework for Secure and Efficient Cloud Resource Allocation

Keerthana.R[1*], Selvakumar.S[2]

[1*]Department of Computer Science and Engineering,School of Engineering and Technology, Dhanalakshmi Srinivasan University,Samayapuram,Trichy,Tamilnadu.India. keerthi.it27@gmail.com
[2]Department of Computer Science and Engineering,School of Engineering and Technology, Dhanalakshmi Srinivasan University,Samayapuram,Trichy,Tamilnadu. ssksri@gmail.com
*Corresponding Author: Keerthana.R

Abstract: In advanced digital infrastructure, cloud computing is the foundation, which provides a cost-effective, scalable, and adaptable solution for various applications. There are several challenges in cloud security and efficient resource allocation due to the increased use of cloud services. The dynamic workload management is lacking in traditional methods, which leads to underutilization, inefficiency, and an increased risk of security threats. These challenges are addressed by the proposed ML-based framework that combines an RF and GB model to improve cloud resource allocation and simultaneously ensure cloud security. The proposed framework also used data preprocessing techniques to reduce the noise, produce a balanced dataset, and extract the key features for workload prediction and anomaly detection. The workload demand can be predicted automatically; intrusion is detected and prevents malicious resources through the proposed hybrid model RF-GB (RFGB). The proposed framework combines anomaly detection and security-aware labelling to eliminate risks like distributed DOS attacks and unauthorized access, ensuring a balance between efficiency and resilience. As an experimental result, the proposed hybrid model attained high-performance results compared to individual RF or GB models in terms of accuracy, recall, and throughput. Moreover, the adaptive optimization techniques are also used by the proposed framework for enhancing the load balancing and service-level agreements. This research also highlights the potential of ML-based intelligent resource allocation in a secure and adaptive cloud environment. Additionally, for the future digital environment, this research contributes a scalable and reliable methodology that can be utilized in several domain applications like finance, healthcare, and government services.

Keywords: Cloud Resource Allocation, Security Enhancement, Random Forest, Gradient Boosting Model, Workload Demand Forecasting.

1. Introduction

Cloud computing has revolutionized the world of technology within the last decade, which has become the foundation of modern information and communication technology (ICT) [1]. The cloud has transformed the manner in which organizations, industries, and individuals operate their digital infrastructure by providing scalable, flexible, and low-cost solutions to store, compute, and deliver software. From small business applications utilizing Software-as-a-Service (SaaS) to large enterprises with mission-critical workloads on Platform-as-a-Service (PaaS) platforms or Infrastructure-as-a-Service (IaaS), the cloud has significantly transformed the availability and performance of computing resources [2]. Nevertheless, there are also new challenges that arise with the exponential growth of the cloud, especially in the field of resource allocation. As millions of users and applications require computational resources in real-time, the importance of secure, efficient, and intelligent mechanisms of their allocation has become the most crucial factor. Cloud resource allocation is the allocation of scarce computing resources (CPU cycles, memory, bandwidth, and storage) among two or more competing users and tasks in a shared environment [3]. The performance, fairness, and cost-effectiveness of the process should be ensured, and security and trust should not be undermined. Historical allocation methods are useful in a static environment but fail to perform effectively in a dynamic workload that varies rapidly in size, complexity, and urgency [4]. Moreover, cloud computing also carries its

own inherent risks because unauthorized users, malicious workloads, or mismanaged resources may disrupt the confidentiality, integrity, and availability of services due to the multi-tenant nature of cloud computing [5]. These issues require the creation of highly adaptive, intelligent, and dynamic resource allocation frameworks that are capable of maintaining equilibrium between performance and security on the fly.

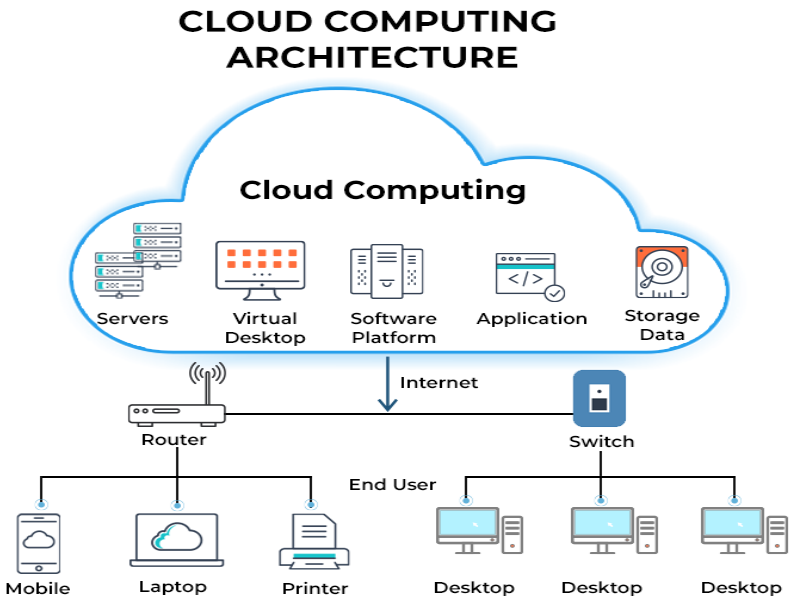

Figure 1. Cloud computing architecture [25].

Here is where the concept of machine learning (ML) can be seen as a groundbreaking facilitator. ML is a branch of artificial intelligence that involves creating models capable of learning patterns based on historical and real-time data and making intelligent decisions with little human oversight [6]. ML has great benefits when used with cloud computing: it is able to forecast the demand of the workload in the future, identify irregularities in user activity, categorize tasks according to their priorities, and dynamically optimize the allocation of resources [7]. Moreover, ML models can be used to incorporate security features into the allocation process itself, recognizing suspicious usage patterns, detecting intrusions, and serving only legitimate requests [8]. Therefore, when paired with ML and cloud resource allocation, not only will the efficiency be increased, but also the system security will be reinforced to establish a secure and efficient cloud resource allocation framework [9].

Efficiency and security are the most important, and must be used to create sound structures. An efficiency-only model will allocate resources in the most efficient way, but does not take into account security, which will leave the system vulnerable to malicious exploitation [10]. On the other hand, an allocation in terms of pure security can excessively focus on defense and therefore lead to inefficiencies and increased costs [11]. To address this gap, an ML-based system that considers both performance optimization and security will be employed to make cloud environments as efficient and robust [12]. One way to apply reinforcement learning in a training system to maximize throughput is by considering user trust scores, ensuring the malicious agent does not receive a disproportionate share of resources. The real promise of ML in resource allocation in the cloud is in such hybrid approaches [13].

In summary, the workloads and threats would be too complex to be handled in the traditional methodology [14]. ML is an efficient solution as it brings efficiency and security to resource allocation [15]. Thus, this research explained the detailed architecture that uses ML models to provide intelligent, secure, and efficient allocation of cloud resources. It also assists in improving the resiliency, cost-efficiency, and reliability of cloud computing, which is headed towards the next generation of intelligent digital ecosystems.

1.1 Contributions of the Paper

- The proposed ML-based framework for cloud resource allocation through combining efficiency and security in performance optimization and threat resilience.

- The high-quality data is provided by the advanced data preprocessing for removing noise, imbalanced, and incomplete data.
- The proposed framework captures both dynamic and anomaly patterns by using a feature work mechanism and utilizes an ML model to enhance the memory, CPU, and storage while reducing the cost and energy usage.
- The load-balancing and service-level agreement (SLA) compliance are improved by using the advanced optimization techniques with ML.
- The benchmark cloud dataset is used to validate the proposed framework; as a result, it attained higher performance compared to the existing techniques.

2. Literature survey

In this section, various earlier research works are discussed in detail with their merits and demerits. The main focus of this section is to address the issues and challenges of the existing model on cloud resource allocation and security enhancement. For instance, Petrovska et al. (2023); Peng et al. (2022) have presented an effective data preprocessing-based cloud security enhancement approach. This method combines an adaptive resource allocation approach capable of protecting the cloud environment against cyber threats. The method ensures timely resource allocation using the NSGA-II algorithm, resulting in a 5% reduction in time and being 1.2 times quicker than other methods for cloud allocation tasks. Saxena et al. (2023); Oloruntoba et al. (2024) proposed a Sustainable and Secure Load Management (SaS-LM) model, which aims to enhance security in the cloud. It evaluates resource utilization and congestion using the DPBHO algorithm. The overall result indicates that this model reduces carbon emissions (46.9%) and energy consumption (43.9%), while increasing the resource utilization level by 16.5%. Silva et al. (2024); Dritsas et al. (2025) have presented an ML-based framework for resource allocation in IoT-Fog-cloud networks. The proposed framework combines the Random Forest Classifier (RFC) with Deep Q-Network (DQN) to predict workload and learn an adaptive offloading policy. The 50,000 entries are generated by the iFogSim simulator, and preprocessing techniques like SMOTE-based class balancing and Min-Max normalization are used. The ε-greedy policy is employed to train the DQN, and the experimental results show that the hybrid RFC-DQN model achieved 91% classification accuracy, reduced task delay by 63.2 ms, and increased offloading accuracy by 91.3%, with approximately 22.5% less energy consumption compared to the baseline model. Gadde et al. (2022) proposed ML algorithms and data analytics to dynamically allocate resources based on workload requirements, ensuring consistently high scalability, availability, and performance. The system improves resource utilization by integrating automated scaling mechanisms, real-time monitoring, and predictive modeling. As a result, it minimizes resource waste by up to 30% and enhances response time by up to 25% compared with existing allocation systems.

Attou et al. (2023) have presented an ML-based anomaly detection model for cloud security enhancement. The proposed model utilizes two datasets, such as Bot-IoT and NSL-KDD, for training and validation. The model incorporates the RF classifier to improve the detection accuracy in cloud computing. As an experimental result, the model was evaluated on two datasets and achieved 98.3% accuracy for Bot-IoT and 99.99% for NSL-KDD. These results are compared, and the proposed model surpasses traditional models in terms of accuracy, precision, and recall. Lekkala et al. (2024) and Ratnayake et al. (2024) have presented an AI-based dynamic resource allocation method for a cloud environment. The approach utilizes ML and DL techniques to develop a predictive algorithm for dynamically allocating resources in the cloud. The model also predicts workload and resource usage to allocate resources accordingly. Evaluation shows that the proposed model increased resource utilization by 25% and reduced violations of quality-of-service by 30%. These results demonstrate that the proposed framework improves efficiency and competitiveness in real-world cloud computing systems. Anbarkhan et al. (2025); Kamble et al. (2023) present a framework for optimizing cloud resource allocation using ML models. To analyze historical data and determine real-time metrics, the proposed framework integrates regression models and neural networks. The framework accurately predicts demand variations and enables dynamic resource distribution. Its effectiveness is evaluated through simulation, showing a 30% improvement in resource allocation and 25% in functional cost compared to existing approaches. Khan et al. (2022) and Nuthalapati, A. (2024) have presented a model for predicting workload and energy state emissions using an ML model in cloud data centers. Various ML models, such as LR and others, and DL models such as GRU, are studied for workload prediction. Experimentally, the GRU achieved better performance than the other models. To evaluate the energy state, four clustering methods — TP-teda, TKmeans, TCLA, and TSSAP —were used. The results showed that TSSAP clustering achieved a high accuracy of 87.48% and 53.8% for detecting VM classes in the proposed clustering approach.

Nassif et al. (2021); Abdallah et al. (2024) investigated ML models and techniques for cloud security. This study classifies them into three categories: types of threats in cloud security, ML techniques, and performance results. The most important cloud security concerns are data privacy, accounting for about 16%, and distributed denial-of-service (DDoS) attacks, around 14%. This study utilizes both individual and hybrid SVM models. It employs the KDD CUP'99 and KDD datasets for training and validation, which results in faster training times. There are 13 threat identification metrics, such as precision, ROC, and others, that evaluate the model's performance. About 60% of the studies compare their model's performance with other models. Kasula et al. (2024) present an AI-based framework for cloud environments to prevent data breaches. The proposed framework, SecureCloudAI, is designed to protect sensitive data in the cloud and enhance overall security. In a real-world cloud environment, the threat can be detected, classified, and responded to by using an RF and LSTM model. The SecureCloudAI model has been experimented with and attained 94.78% accuracy in malware detection and 90.92% threat classification accuracy. The SecureCloudAI framework also provides web intrusion detection, network traffic analysis, and threat detection, such as malware and viruses, without compromising the scalability and efficiency of the cloud environment.

Barua et al. (2024) have presented an AI-based resource allocation framework for microservices in Hybrid Cloud Platforms. The proposed framework aims to reduce costs and enhance performance through optimal resource utilization using RL. Based on microservices, the RL system regularly adjusts resources and predicts future consumption trends. The framework was evaluated through simulations, and the results show that it reduces expenditure costs by 30 to 40%, improves resource utilization by 20 to 30%, and decreases latency by 15-20%. These findings conclude that this dynamic framework offers a scalable solution for cloud resource management. Singh et al. (2023) and Fahimullah et al. (2024) proposed an allocation technique using ML for SDN-enabled fog computing environments. The proposed model combines resource allocation techniques and is validated with FogBus and iFogSim, assessing performance in terms of latency, energy, and power consumption. The attained results demonstrate that the technique reduces response time by 18.14%, decreases run time by 21%, lowers network utilization by 9%, reduces energy utilization by 7%, and cuts delay by 25.29% compared to existing methods. Choudhury et al. (2024); Latha et al. (2024) have studied cloud scalability with AI-based resource management. The performance of the five AI models is evaluated by this study for cloud resource management to enhance scalability. The five AI models—RL, LSTM, Autoencoders, Gradient Boosting Mechanics, and Neural Architecture—are evaluated to reduce operational costs and improve resource utilization. As a result, the RL model decreases cost and delay by 20% and 30%, respectively; the LSTM model improves prediction accuracy by 12% and increases resource utilization by 22%; the GBM reduces prediction error and cost by 30% and 20%; the Autoencoders detect anomalies with 97% accuracy; and finally, the neural architecture enhances computational speed and accuracy by 25% and 18%. When all five models are combined, these methodologies improve cloud scalability and resource allocation. Li et al. (2023) have proposed an ML-based technique for resource allocation and task placement in edge computing. The proposed technique, named TapFinger, aims at reducing the execution time of ML tasks in the cloud. TapFinger offers distributed scheduling utilizing multi-agent reinforcement learning (MARL), combined with Bayes' theorem and masking schemes. The model was initially deployed for single-task scheduling, then extended to multi-task scheduling for broader applications. Experimental results show that the proposed technology reduced execution time by 54.9% and enhanced resource efficiency compared to other schedulers. Saini et al. (2024); Safi et al. (2024) have proposed an ML-based framework to improve the security and reliability in a cloud environment. The proposed framework protects the private data of both users and consumers, and also reduces the data re-identification risk by using the K-anonymity technique. Additionally, the T-MBFD algorithm, with key parameters like resources, time constraints, and job size, is used by the framework to assist in resource allocation. The framework was evaluated on 95,000 instances and achieved 96.41% accuracy, an F1-score of approximately 0.97, precision around 0.95, and recall near 0.98, with average process time around 0.22 sec per task. These results suggest that the proposed framework offers privacy preservation, trust enhancement, and resource efficiency.

2.1 Limitations

The proposed DL-based model, used for efficient and secure resource allocation, has demonstrated many improvements, but some limitations still exist. First, using historical workload data to train the model can limit its flexibility in responding to sudden or unexpected demand surges and may lead to temporary underutilization of resources or excessive provisioning. High-quality, labeled datasets are also essential to the framework, but they are not always available in real-world cloud environments where anomalies and threats are not always logged. Additionally, deep machine learning models impose an extra computational burden, which might not be acceptable in applications with strict latency requirements. Scalability is another issue; although the framework works well in small to medium-sized cloud infrastructures, its performance in very large, heterogeneous, and multi-cloud environments

remains unverified. Furthermore, security considerations may not address all emerging cyberattacks or prevent the system from new types of intrusions. Lastly, ML-based allocation decisions can be imprecise, providing administrators with no clear explanation of how resources are allocated, particularly when addressing compliance and regulatory demands in sensitive sectors.

2.2 Motivation

Cloud computing has evolved rapidly within organizations for managing and gathering large amounts of cloud data, but the main challenge in a cloud environment is achieving fair, secure, and effective resource allocation. Traditional static allocation techniques cannot handle the high dynamic workloads and diverse user requirements, leading to decreased performance, resource wastage, and unmet Service-Level Agreements (SLAs). Additionally, threats such as DoS attacks, workload-based intrusions, and unauthorized access threaten system reliability and security. Therefore, the proposed framework primarily focuses on improving efficiency and security in the cloud. It combines machine learning to enhance pattern detection and adapt to complex data. The model also aims to create an energy-efficient, low-cost, and sustainable cloud operation framework. Furthermore, the framework integrates workload prediction, real-time optimization, and anomaly detection to become an intelligent system. The challenges and gaps are addressed to ensure the cloud environment remains secure, reliable, and scalable in real time, thereby increasing trust among users and supporting various domains like government, finance, and healthcare services.

2.3 Problem Statement and Proposed Methodology

The various applications, dynamic workloads, and different resource requirements are used to classify the cloud computing environment. The existing resource allocation methods are mostly static or reactive, resulting in issues like low utilization, over-allocation of resources, or hindered framework performance. To identify and reduce security threats, traditional built-in mechanisms can degrade data integrity and trust in cloud services. As the cloud platform scales up, ensuring secure and efficient resource allocation while maintaining Service-Level Agreements becomes more complex. Traditional ML models perform well but often face overloads when integrating workload optimization with security-aware allocation, leading to disruptions. To address these issues, this research proposes a framework capable of allocating resources by predicting balancing efficiency, workload demand, and incorporating security-aware metrics. The architecture of the proposed work is depicted in Figure-2. In real-time cloud environments, the main challenges are achieving high scalability, accuracy, and trustworthiness while minimizing costs and enhancing security.

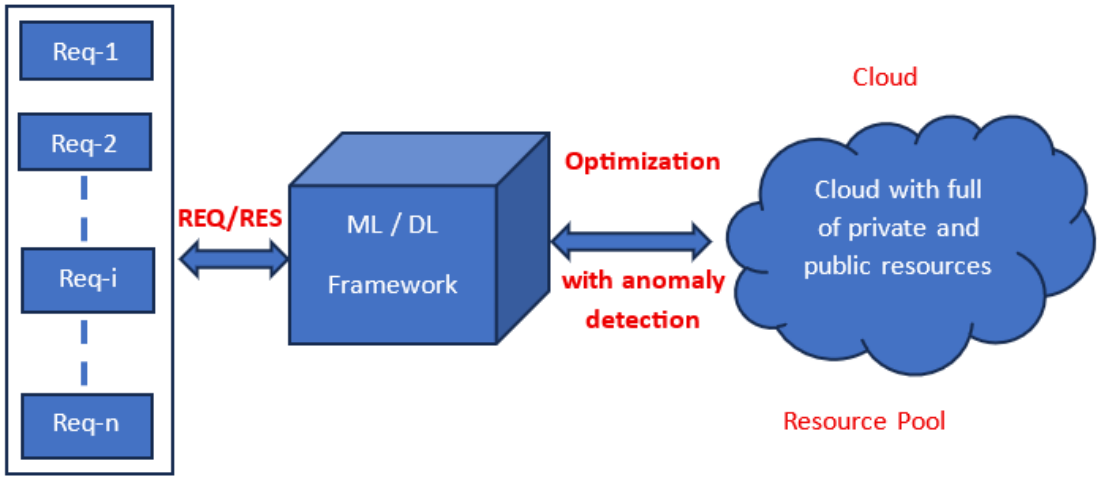

Figure-2. Proposed Resource Allocation Architecture

In this paper, a hybrid machine learning framework is developed to secure and manage data in the cloud. The proposed RFGB model involves several steps in cloud resource allocation, such as data pre-processing, feature extraction, anomaly detection, workload forecasting, and resource allocation. These steps are briefly discussed in this section. The overall Workflow of the proposed model is shown in Figure-3.

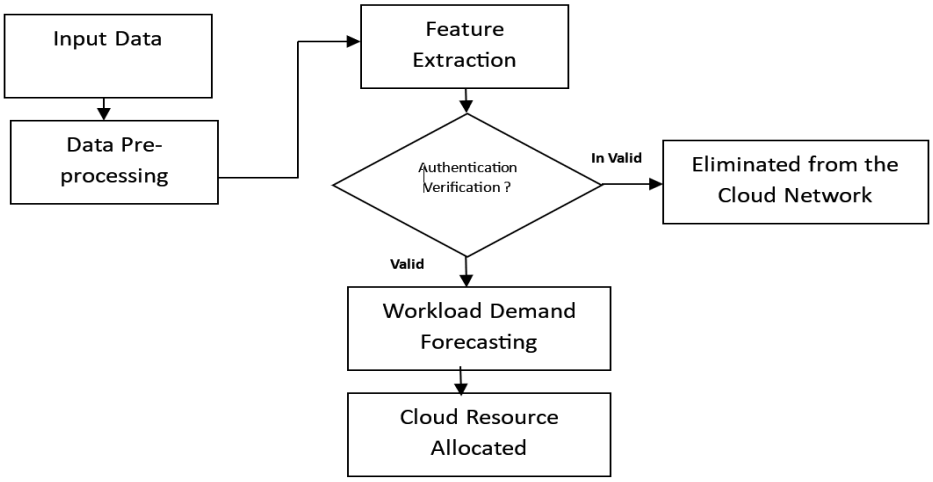

Figure-3: Proposed Model Workflow

2.4 Data Pre-processing

Pre-processing of data is one of the key steps in the creation of a machine learning framework to allocate resources in the cloud as securely and efficiently as possible. As cloud environments create heterogeneous, large-scale, and dynamic datasets, appropriate pre-processing is crucial to ensure that raw data is converted to structured, consistent, and high-quality data ready to be used in the machine learning algorithms. This step eliminates redundancy, process noises, fixes missing values, normalizes scores, and identifies meaningful features that capture patterns of resource usage and possible anomalies. The most important steps of pre-processing and the mathematical formulations are listed below.

2.5 Data Collection And Integration

The obtained raw dataset in the cloud may contain various logs such as security logs, resource usage metrics, energy consumption data, and task information. Where security logs contain authentication failures, access logs and intrusion alerts, resource usage metrics contain memory usage, CPU load, bandwidth, and disk I/O, energy consumption data consists of cooling overhead and server usage, and task information contains execution time, arrival time, and completion status. The raw datasets in the cloud are determined as;

$$D = \{(x_1, x_2, \ldots\ldots, x_n), y\}$$

In the above equation, y represents the target variables such as anomaly label, allocated resource or workload category, x_i represents the CPU memory or bandwidth of the i^{th} feature. Various datasets from heterogeneous sources are combined using the integration function that ensures the alignment of attributes regularly. If two datasets such as D_1 *and* D_2 shares the same key, like VM ID or timestamp, then the integration of the two datasets would be determined using the following equation.

$$D = D_1 \bowtie D_2$$

In the above equation, $\bowtie$ represents the joining operation.

2.6 Handling Missing Values

Most of the cloud data consists of incomplete or missing records due to network delays or system failures. Missing values are identified using the imputation approaches such as mean imputation and KNN-based imputation.

2.7 Mean Imputation

$$x_{i,j}^{*} = \frac{1}{N}\sum_{k=1}^{N} x_{k,j}$$

2.8 KNN-based Imputation

In the feature of x_iThe missing values are identified and estimated using the weighted average of their K nearest neighbours.

$$x_{i,j}^{*} = \frac{\sum_{j=1}^{k} w_j x_{ij}}{\sum_{j=1}^{k} w_j}, \qquad w_j = \frac{1}{d(x, x_j)}$$

In the above equation, $d(x, x_j)$ represents the distance metrics like Euclidean.

2.9 Outlier Detection and Removal

Outliers typically result from abnormal activities, such as sudden workload surges, Distributed Denial of Services (DDoS), or sensor failures. The outliers are identified using the z-score normalization approach.

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$$

In the above equation, σ_j represents the standard deviation of the j^{th} feature and μ_j represents the mean value of the j^{th} feature.

2.10 Normalization

The cloud data contains various heterogeneous features, including memory in GB, CPU usage in %, and bandwidth in Mbps. Thus, scaling needs to bring the given values into a common range.

2.11 Min-Max Normalization

$$x'_{i,j} = \frac{x_{i,j} - min(x_j)}{max(x_j) - min(x_j)}$$

This equation helps to produce unit variance features and a zero mean, which makes the data more suitable for ML models.

2.12 Feature Extraction

For secure and efficient cloud resource management, feature extraction is also one of the key factors that helps to convert the raw obtained telemetry, such as memory, CPU, I/O, auth logs, network flows, and some configuration changes, into important and compact vectors, which are very useful for anomaly detection and ML/DRL.

$$\mu_t^{CPU} = \frac{1}{W} \sum_{i=0}^{W-1} CPU_{t-i}$$

And the variance $\sigma_t^{2,CPU} = \frac{1}{w} \sum_{i=0}^{W-1} (CPU_{t-i} - \mu_t^{CPU})^2$. Security-relevant features such as unusual port access, failed login counts (f_t) and destination-IP entropy.

$$H_t = -\sum_{i} p_{i,t} \log p_{i,t}$$

In the above equation, $p_{i,t}$ represents the empirical share of connection to the IP i in the window. In this, where features are normalized before modeling: $\tilde{X} = (X - \mu)/\sigma$. Dimensionality reduction, like the PCA approach, to a low-dimensional latent space $z = U^t(\tilde{X} - \mu)$, which helps to preserve the variance in control. An anomaly score is more effective for security scoring, which is evaluated using the following equation;

$$D_M(\tilde{X}) = \sqrt{(\tilde{X} - \mu)^T \sum^{-1} (\tilde{X} - \mu)}$$

Flagging example with the large D_M. Finally, the feature selection crops the redundant signals, which helps to improve interpretability and also reduce latency. Advanced extraction techniques, including statistical, temporal aggregation, informative-theoretic features, selection, and normalization, enable the cloud manager to receive robust and compact inputs for both threat detection and resource optimization.

2.13 Data Splitting

The pre-processing step is completed, then the dataset is divided into training, validation, and testing subsets:

$$D = D_{train} \cup D_{val} \cup D_{test}, D_{train} \cap D_{val} \cap D_{test} \neq \emptyset$$

In Dataset D, 70% of the data is allocated for training, 15% for validation, and the remaining 15% for testing, ensuring the ML model generalizes accurately without overfitting.

2.14 Anomaly Detection

Thus, in this paper, a hybrid ML (RFGB) model is utilized to verify the authentication of the input data and to allocate resources in the cloud based on the workload demand.

2.15 Random Forest

The Radom Forest (RF) is an ML algorithm, which is used for both classification and regression tasks, and its prediction is based on using an ensemble of decision trees (DT). The main motive of this RF model is to generate a large number of DT, which contain various subtrees and where each subtree is trained on a distinct subset of the data, as shown in Figure-4. Each individual subtree's predictions are integrated to make the final prediction. The RF algorithm is applicable for both classification and regression tasks. In a classification task, the final prediction is attained by combining the model prediction of individual trees, whereas in a regression task, the final prediction is attained by the mean prediction of all individual trees.

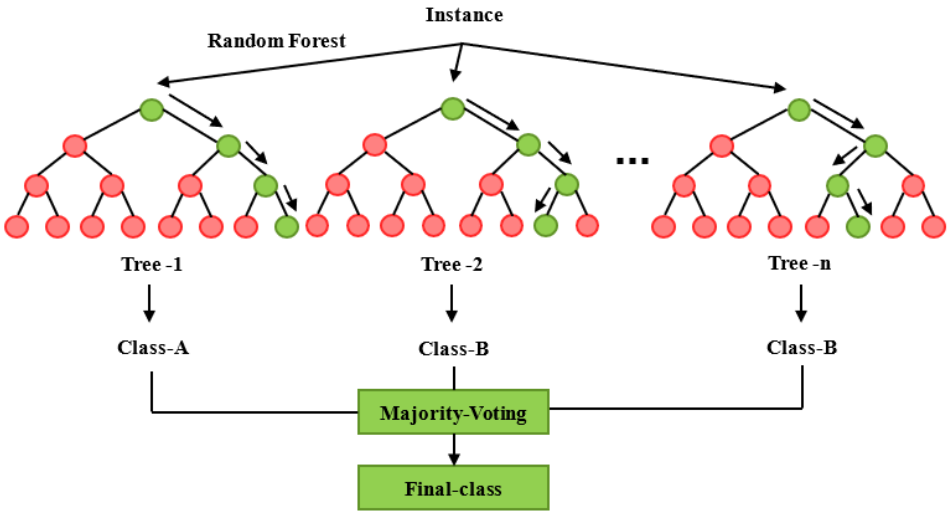

Figure-4: Structure of the Random Forest Algorithm

Where the input dataset is denoted as ,D

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

The feature vector for the ith instance is denoted as x_i and label for regression or classification is denoted as $y_i \in \{1, 2, 3, \ldots, K\}$.

$$\hat{P}_k(x) = \frac{1}{B}\sum_{b=1}^{B} 1[h_b(x) = K]$$

Where the frost size and prediction of the b-th tree decision is denoted as B and $h_b(x)$. The final output of the classification task is:

$$\hat{y}(x) = arg \max_{k \in \{1,\ldots,K\}} \hat{P}_k(x)$$

For the binary class $\{0, 1\}$, the RF probability of class 1 and class 0 is denoted by $\hat{P}_1(x)$ and $\hat{P}_0(x)$.The output of each tree's real value $\in \mathbb{R}$. The forest prediction is the average:

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x)$$

2.16 Gradient Boosting

One of the machine learning models, Gradient Boosting, is used here to check the resource availability, management, and allocation in the cloud based on the user requests. It uses both classification and regression methods. The GB model predicts the correct resources according to the demands and the availability. It learns from past errors and weak decision trees, predicting the required resources and improving the efficacy of dynamic resource allocation in the cloud. It is a framework that can be used for classification or regression processes. GB, XGB, and LightGB models help to create an effective classifier or regressor for forecasting cloud resources. In particular, GB is integrated with several base methods, like KNN, NN, and NB, to filter the optimal values to obtain higher performance. Among these, GB is an advanced and efficient system, known as a gradient boosting decision tree, which exhibits competitive performance in processing regression tasks. The GB can obtain the optimum gradient values using the base methods. It uses the regression process to process the dataset with n entities under different states, and rewards are obtained from simulation to solve the second-order cone problem, using

$$D = (x_i, p_i)(|D| = n, x_i \in D \cup y, p_i \in \mathbb{S})$$

The state representation $(D \cup y) = [y_1^i, y_2^i, \dots, y_m^i, d_1^i, d_2^i, \dots, d_n^i]$ is denoted as x_i and the solution is represented as p_i. The loss value is minimized to optimize the regression process over the dataset D, which is expressed as:

$$\mathbb{E}_D\left[L(\hat{P}, P)\right] = l\left(\hat{P}, P\right) + \Omega(\phi) = l(f(X), P) + \Omega(\phi)$$

The model ϕ and $f(X)$ maps the requested demand input values with the available resource values P to determine the correct cloud resources. The X is the system model, and P is a second-order solution problem.

In this paper, the GB method used a dynamic resource allocation process for real-time cloud applications by evaluating the defined states, actions, and rewards. It computes the station-action-reward values in a table to compare each time of resource requests. In GBDT weak model predicts the mean value P simply, and this is enhanced by aggregating K additively to fix the size of the decision tree as base estimators to predict the pseudo-residual in the previous results. The final prediction is done through a linear combination of all the outputs of K regression trees. This output could be expressed as:

$$f(x) = \sum_{k=0}^{K} f_k(x) = f_0(x) + \sum_{k=0}^{K} \theta_k \phi_k(x)$$

Where f_0 is the initial guess, $\phi_k(x)$ is the base estimator in iteration k and θ_k is the weight for the K_{th} estimator in fixed learning rate. The product $\theta_k \phi_k(x)$ denotes the step at iteration k. The structure of the GB model is shown in Figure-5.

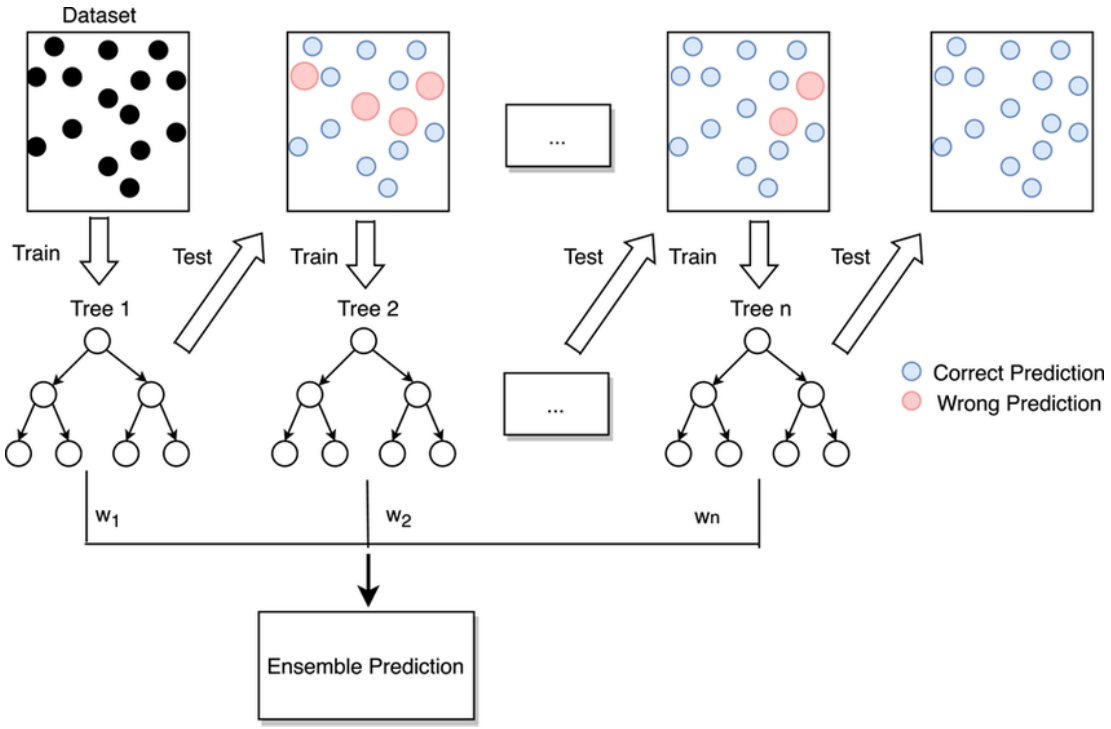

Figure-5. Structure of the Gradient Boosting Algorithm [26]

2.17 Workload Anomaly Detection

To increase the robustness and trust in resource allocation, the hybrid RFGB work must be forecasted, which is merged with anomaly determination. This allows verifications and trust in the scoring modules that enhance predictions and allocation decisions, ensuring they are not manipulated by adversarial inputs or unauthorized access. The hybrid forecast $\hat{y}_t^{hybrid}$ is generated as a weighted ensemble of RF and GB predictions.

$$\hat{y}_t^{hybrid}=\alpha\hat{y}_t^{RF}+(1-\alpha)\hat{y}_t^{GB}, \qquad 0\le a\le 1,$$

Where $\hat{y}_t^{RF}$ and $\hat{y}_t^{GB}$ Workload forecasts are used, and balances are made in terms of accuracy. To determine tampering or abnormal spikes such as potential DoS, resource hijacking, or adversarial poisoning, this shows the residual-based anomaly score is computed:

$$A_t=\left|y_t-\hat{y}_t^{hybrid}\right|,$$

This denotes an anomaly flag is raised if $A_t>\theta, where\ \theta$ is a changing threshold, for example, mean $+3\alpha$ of residuals. Security-aware allocation acts as a modification of the resource allocation decisions, like $\hat{R}_t$ by combining forecasted demands with a trust score of $T_t\in[0,1]$ provides systems merging and authentication status, and anomaly likelihood:

$$\hat{R}_t^{secure}=\hat{y}_t^{hybrid}\times\left(1+\lambda(1-T_t)\right),$$

Where λ handles the security penalty. This denotes that the trust is high ($T_t\approx 1$) , allocation obeys the forecast closely, and if anomalies or unauthorized signals reduce T_t, then the system upgrades reserved resources conservatively or switches to a decreased level of policy, to prevent service-level agreement (SLA) authentication checks, anomaly probability, and past behaviour:

$$T_t=w_1S_t^{auth}+w_2(1-P_t^{anom})+w_3S_t^{history}, \qquad \sum w_i=1,$$

Where S_t^{auth} is the success a measure of access authentication for P_t^{anom} is the probability of an anomaly from residual analysis and $S_t^{history}$ denotes capture consistency of past interactions, such as integration that enhances the workload forecaster (RF+GB), which produces accurate predictions or analysis of the allotted safeguard against

malicious manipulations, unauthorized access, or fatal poisoning, hence enhancing both efficiency and trust of cloud operations. Once the input data is verified, malicious data is eliminated, and the normal data demands are forecasted and securely allocated the resource in the cloud using the proposed hybrid ML-model.

2.18 Security-Aware Labelling

The secure allocation is assured by the proposed framework through data labeling, for each sample, either labelled as benign or malicious. The labelling function is represented as:

$$y_i = \begin{cases} 1 & if\ the\ request\ is\ secure \\ 0 & if\ the\ request\ is\ malicious \end{cases}$$

Where the y_i able to offer optimized allocation and security for the ML classifiers.

2.19 Workload Demand Forecasting

The workload- forecasting module makes a task a time- series determined of observed resource metrics and concept features $X_t = [u_{t-w+1}, u_t, h_t, d_t, c_t, \dots]$ which makes with a sliding window module of length w (past utilizations u), time features (hour/day h_t, d_t), categorical/ concepts features c_t and choice exogenous signals; that supervised tasks are the following step (or multi- step) workload$y_{t+\tau}$. For the Random Forest (RF) model, this makes to understandable and an ensemble of Mregression trees $\{T_m(\cdot)\}_{m=1}^{M}$ denote that an individual tree is fit on the bootstrap sample and random feature subsets. The Rf point prediction is the average of the tree's outputs:

$$\hat{y}^{RF}(x) = \frac{1}{M}\sum_{m=1}^{M} T_m(X)$$

While training each tree T_m decreases the squared error on its bootstrap sample; the ensemble decreases variance by averaging so the RF expected squared error decomposes as $bias^2 + variance - covariance$ terms overall trees, explaining its strength to noisy workload traces. A crucial feature is computed from mean decrease in impurity (MDI) or permutation importance to identify which lagged inputs or concept signals drive forecasts. For Gradient Boosting (GB/GBDT), this provides the model with the estimates as an additive ensemble of K weak learners, which generally use shallow regression trees that fit in sequence to the residuals:

$$\hat{y}^{GB}(X) = \sum_{k=1}^{K} \gamma_k h_k(X),$$

Where each base learner h_k is taken to reduce the loss, commonl0079 Mean Squared Error $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$.Training is provided by gradient descent in function space: at each iteration, it computes k pseudo residuals $r_i^{(k)} = -\frac{\delta L(y_i \hat{y}_i)}{\delta \hat{y}_i}|_{\hat{y}=\hat{y}^{(k-1)}(X_i)} = y_i - \hat{y}^{(K-1)}(X_i)$ for squared loss, Fit $h_k(\cdot) or\ r^{(k)}$,The set $\hat{y}^{(k)}(X) = \hat{y}^{(k-1)}(X) + \eta\, \gamma_k\, h_k(X)\ where\ \eta \in (0,1)$ is the learning rate and γ_k a line-search scalar (or per-leaf values). Regularization techniques, which involve the learning rate η, maximum tree depth L_2, shrinkage in leaf weights due which subsampling fractions s that workloads. For this multi-step forecasting, control overlifting to burst or anomalous events within the horizon H This could use direct modelling (train separates the models for each horizon or recursive forecasting, which feeds $\hat{y}_{t+1}$back as input; the GB formulations support either.

2.20 Dynamic Resource Allocation

The hybrid proposed model integrates the RF and GB models for the dynamic Resource allocation and scheduling module. The hybrid model proactively predicts the result based on given computing resources like (VM, CPU cores, or Containers) and decides how tasks are scheduled in real time. The requirement of the prediction is $\hat{y}_{t+1}^{hyb}$ is produced by each workload prediction model at each decision epoch t. The hybrid model prediction is based on the calculation of a weighted aggregation of RF and FB predictions:

$$\hat{y}_{t+1}^{hyb} = \alpha \cdot \hat{y}_{t+1}^{RF} + (1-\alpha) \cdot \hat{y}_{t+1}^{hyb}, \qquad \alpha \in [0,1]$$

The provisioning function is used to map the predicted demand to the required resource capacity, like quantifying the VM or cores:

$$R_{t+1} = \left\lceil \frac{\hat{y}_{t+1}^{hyb}}{C} \right\rceil$$

Where the per-resource processing capacity is denoted as C. The required resource capacity is calculated. After the R_{t+1} is determined then the scheduling problem is formulated for allocate task like $T = \{T_1, T_2, T_3, \ldots, T_n\}$ with workload w_i for the obtainable resources $R = \{r_1, r_2, r_3, \ldots, r_{R_{t+1}}\}$. To minimize the response time or delay is the main objective of the scheduling task:

$$\min_{\pi} \sum_{i=1}^{n} (Wait(T_i, \pi) + Exec(T_i, \pi))$$

Where the task-to-resource assignment policy is denoted as π, where the queueing delay and execution time are denoted as $Wait(\cdot)$ and $Exex(\cdot)$. Based on the feedback, the allocation is automatically updated. When the observed requirements y_{t+1} deviate from prediction $\hat{y}_{t+1}^{hyb}$ an error signal is evaluated and stored in the RF and GB learners for online retraining and minimizing the drift.

$$e_{t+1} = y_{t+1} - \hat{y}_{t+1}^{hyb}$$

To assure the safety and stable buffer δ is combined for the uncertainty:

$$R_{t+1}^{*} = R_{t+1} + \delta, \;\; \delta = \beta \cdot \sigma_{\hat{y}_{t+1}^{hyb}}$$

Where the prediction of the variance $\sigma_{\hat{y}_{t+1}^{hyb}}$ is evaluated based on the ensemble dispersion, and the confidence scaling factor is denoted as β. The proposed hybrid model integrates the RF and GB for their ability to handle the noisy input feature and can capture the difficult non-linear workload patterns. The proposed hybrid approach ensures the accurate approximation of requirements and proactive scheduling. The approach leads to both minimization of SLA violation and over-provisioning at the same time, where compared to single or static model allocation methods, the proposed approach improved the cost and energy efficiency.

3. Experiment Results and Discussion

In this section, various simulation results of the proposed model are explained in detail with a graphical representation. The input data samples are collected from the publicly available Kaggle dataset [27]. The input data includes both normal and abnormal data. Initially, the proposed model detects and classifies the two different classes of input data. Once the data are classified, the malicious data are removed from the network, and the normal data are analysed for resource allocation. The efficiency of the proposed model in analysing the data class and resource allocation is evaluated using different metrics through the simulation software installed in the system with a 1 TB hard disk, 64 GB RAM, Intel i7 12th Gen processor, and Windows OS. Figure-5 shows the class distribution of the dataset. It contains about 13449 normal samples and approximately 11737 anomaly samples. Although there are more normal samples, the difference is 1706, indicating the dataset is nearly balanced. This balance benefits the ML model by providing reliable classification and reducing bias toward any one class.

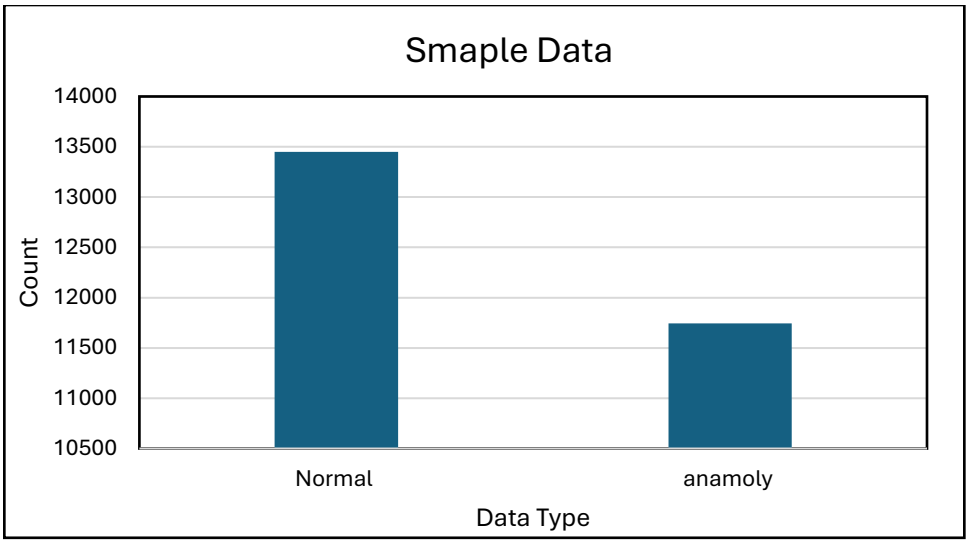

Figure-5: Input Sample data

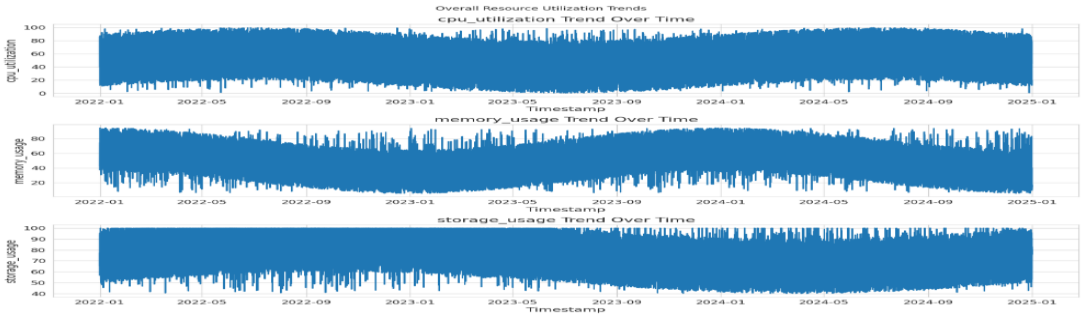

Figure-6 CPU utilization

Figure-6 shows the CPU utilization over time. The CPU usage varies dynamically from 0% to 100%, indicating changing computational demands from low to high. The system mostly uses the CPU between 80% and 100%, suggesting it is under heavy load most of the time and may approach saturation at high levels. This high usage indicates that the CPU has a reserve capacity for handling complex tasks. Based on the visual analysis of the dense area of the graph, the average CPU utilization is around 70% to 80%. The periodic low points under 20% reflect periods of low activity during maintenance, Windows updates, or off-peak hours. The spikes are managed effectively through CPU scaling policies, such as dynamic cloud VM allocation.

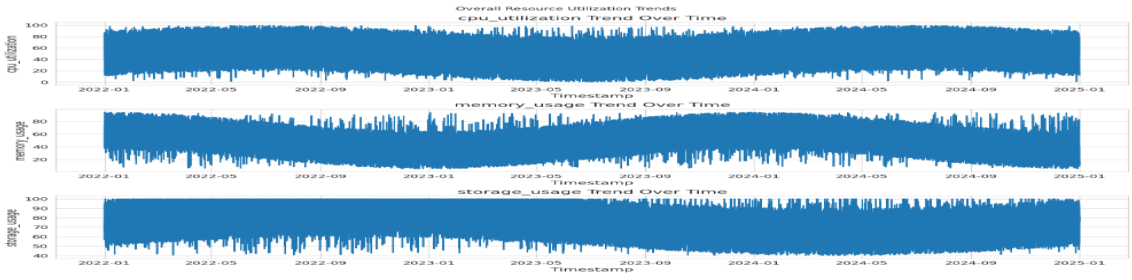

Figure-7 Memory Usage

Figure-7 represents the trend and patterns of the memory usage. It shows that the memory usage was constantly high and fluctuated between 10% and 95% at times. There are periods where memory usage suddenly drops, indicating the process is terminated. It also stated that memory usage exhibits long-term cyclic patterns, including periodic batch processing jobs and integration with seasonal workloads. The baseline of the memory usage is higher than the range of 50% to 60%, indicating that memory-intensive applications are regularly running with more memory usage. Sometimes the memory usage reaches the peak level around 95%, which indicates that at a time of potential memory

pressure or swapping, it will lead to a performance impact on the system. The stability of the system is maintained through regular monitoring and potential memory scaling, such as adding more ROM or optimizing application memory usage.

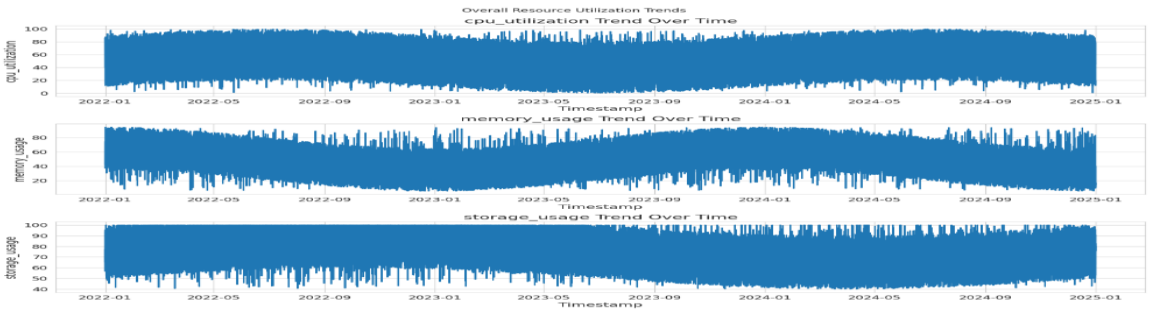

Figure-8 Storage Utilization

Figure-8 represents the storage usage trends. Storage has a consistent high utilization of around 40 to 100, with a gradual increasing trend. This denotes unremedied data dumping, most probably logs, databases, or user-generated files. Periodic decreases to less than 50 percent can be related to cleaning up or archiving. The majority of the storage use ranges between 50% and 100%, indicating that the storage is used more frequently. The cloud stops its ability when it reaches 100% of the storage, indicating the need for proactive storage management.

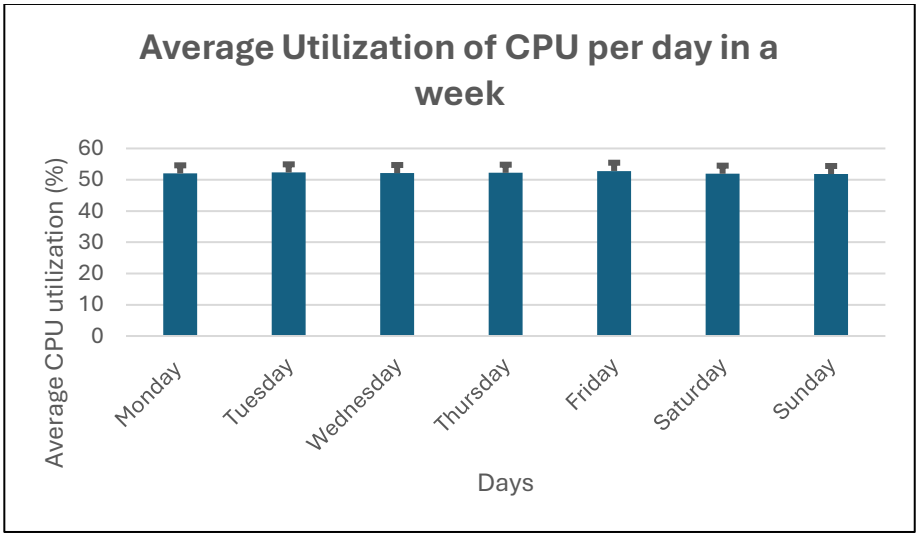

Figure-9: Average Utilization of CPU per day in a week

The average CPU utilization (%) for every day of the week is denoted in Figure-9. The data indicates that CPU utilization remains relatively consistent throughout the week, ranging between 51% and 52%. The average CPU utilization on Monday, Tuesday, Wednesday, Thursday, and Sunday is about 52%. Consequently, on Friday, the utilization increases slightly to about 52.5%, indicating a minor workload peak. On Saturday, utilization is lower at around 51.8%, indicating a little decrease in processing demand over the weekend. The low daily changes in CPU utilization give a steady workload in the system, which determines peaks or troughs, showing steady computing needs all week long. The little error bars also indicate that there are no significant variations in the daily CPU consumption.

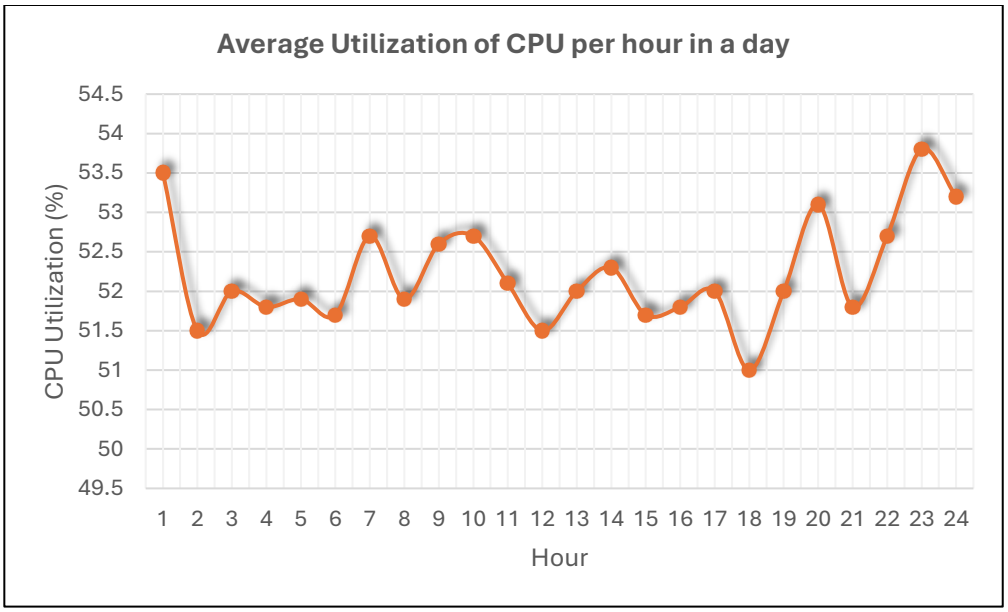

Figure-10: Average Utilization of CPU per hour in a Day

Figure-10 represents the average CPU usage by each hour of the day, evaluated using percentages (%). It shows how system memory usage varies over a 24-hour cycle. The graph indicates that CPU usage remains relatively stable around the 50 to 55% range with some minor fluctuations. The lowest CPU usage occurred at the 17th hour, nearly 51%, indicating reduced activity during that time. The highest CPU usage was at the 23rd hour, about 54%, suggesting increased computational demand during that period. The shaded region on the graph represents variability, showing fluctuations of -2 to 2% from the average. This result indicates that CPU usage was normal and consistent throughout the day. Slight dips occur during the early evening and afternoon, while higher CPU usage is observed during the early morning and late-night hours.

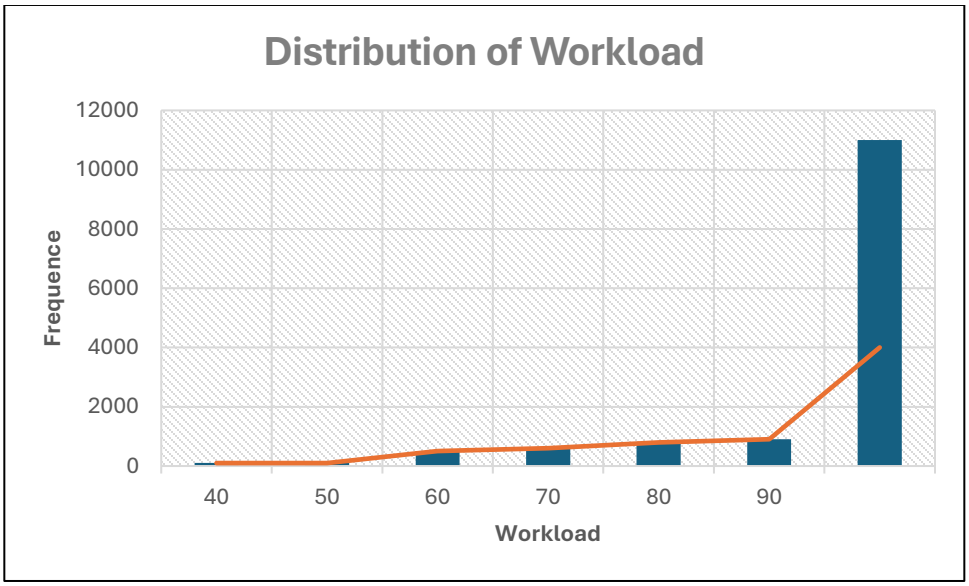

Figure-11: Distribution of Load

The workload distribution shown in figure-11 generally falls between 38 and 100, with notable increases toward the higher end. The maximum workload is the most relied-upon condition, as evidenced by the majority of values clustering around 100 and the frequency rapidly peaking at about 12,000 occurrences. Values below 50 are extremely rare, with near-zero frequencies, while the frequency gradually rises between 60 and 90, staying relatively low (under

1,000). The distribution shows a strong focus of workloads at the higher end overall, indicating that the system often runs close to its maximum capacity.

Table-1: Performance Evaluation Of the Proposed Model On Authentication Verification

Model	Accuracy	Precision	F-score	Recall
RF	**0.996**	**0.998**	**0.996**	**0.995**
GB	**0.996**	**0.997**	**0.996**	**0.996**
Hybrid Model (RFGB)	**0.998**	**0.996**	**0.996**	**0.997**

Table 1 presents a comparison of the performance of three models: Random Forest (RF), Gradient Boosting (GB), and the Hybrid Model. The RF model achieved accuracy, precision, F1-score, and recall of approximately 0.996, 0.998, 0.993, and 0.995, respectively. These results show that the RF model is highly accurate, but its recall is slightly lower. The GB model also performs well, with accuracy, precision, F1-score, and recall scores of about 0.996, 0.997, 0.996, and 0.996, respectively. This indicates that the GB model balances precision and recall effectively. The Hybrid Model surpasses both RF and GB in terms of high accuracy and recall, with scores around 0.998 and 0.997; however, its precision is slightly lower at approximately 0.996. Ultimately, the comparison suggests that while RF and GB models perform well independently, the Hybrid Model provides improved recall and accuracy simultaneously, reflecting its balanced and superior performance.

Table-2 Performance comparison

Models	Throughput (Event/sec)	Energy Consumption (kWh)	Processing Time (s)	Resource utilization (%)	Latency (s)
DQNN [22]	500/s	0.80	0.15	95%	0.0150
Om-FNN [23]	800/s	0.90	0.12	88%	0.0430
HUNTER [24]	600/s	0.70	0.20	74%	0.0350
Proposed Model	1100 /s	1.00	0.10	99 %	0.0120

Table 2 shows the overall effectiveness of the proposed model in terms of processing time (s), latency (s), throughput (events/sec), resource usage (%), and energy consumption (kWh). The proposed model was compared with several existing models such as Om-FNN, DQNN, and HUNTER. It demonstrates that the throughput of Om-FNN reaches 600/sec, DQNN reaches 500/sec, HUNTER reaches 600/sec, while the proposed model achieves 1100/sec. Energy consumption for DQNN is 0.80, HUNTER is 0.70, Om-FNN is 0.90, and the proposed model consumes 1.00. DQNN takes 0.15 seconds for processing, Om-FNN takes 0.12 seconds, HUNTER takes 0.20 seconds, while the proposed model only takes 0.10 seconds. Resource usage for DQNN is 95%, Om-FNN is 88%, HUNTER is 74%, and the proposed model uses 99%. Latency for DQNN is 0.0150 seconds, Om-FNN is 0.0430, HUNTER is 0.0350, and the proposed model has a latency of 0.0120. Overall, these results show that the proposed model outperforms the other models in nearly all aspects.

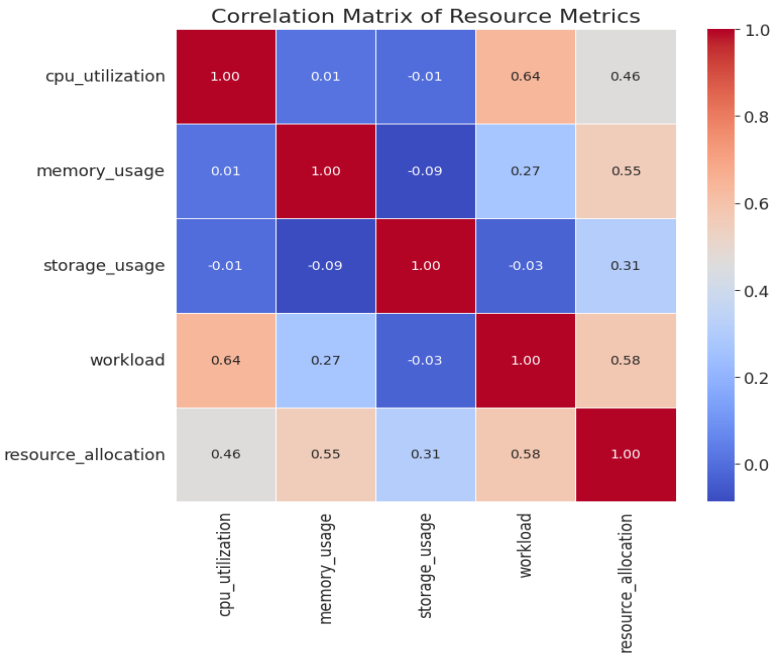

Figure-12: Correlation Matrix for Cloud Resource Allocation

In figure-12, the values range from -1 (negative) to +1 (positive). The correlation matrix shows the links between system resource measures. Workload (0.64) and resource allocation (0.46) are strongly positively correlated with CPU utilization, while memory (0.01) and storage (-0.01) are only weakly correlated. Storage usage has very weak relationships, with resource allocation showing the strongest connection (0.31). Memory consumption has poor correlations with workload (0.27) but shows significant correlations with resource allocation (0.55). Higher workloads lead to increased CPU usage and resource allocation, as indicated by the somewhat positive correlation between workload and resource allocation (0.58). The most important finding is that workload and CPU usage (0.64) form the strongest pair, indicating that workload has a substantial effect on CPU demand.

4. Conclusion

This research presents an ML-based framework for efficient and secure cloud resource allocation by addressing the challenges of traditional static methods in heterogeneous and dynamic environments. The proposed framework integrates RF and GB into a hybrid model, which can predict workload demands, allocate resources, and detect anomalies while maintaining a high level of security. Experimental results indicate that the performance of the proposed model surpasses existing models in terms of accuracy, throughput, and resource utilization, while also reducing energy consumption, costs, and latency. Furthermore, the model enhances cloud trustworthiness by incorporating anomaly detection and intrusion-aware metrics, ensuring strong resilience against malicious workloads without compromising efficiency. Its adaptability highlights its potential for real-time deployment, especially in sensitive domains such as healthcare, finance, and government services, where security and reliability are critical. Overall, this research develops an intelligent, security-aware, and scalable resource management system that revolutionizes the next-generation cloud ecosystem.

Future work will focus on extending edge-environment and multi-cloud frameworks to improve resilience against cyber-attacks and enhance the interpretability of ML-driven decisions.

Reference

1. Matthew, U. O., Kazaure, J. S., & Okafor, N. U. (2021). Contemporary development in E-Learning education, cloud computing technology & Internet of Things. *EAI Endorsed Trans. Cloud Syst.*, *7*(20), e3.
2. Ademilua, D. A. (2023). Advances and Emerging Trends in Cloud Computing: A Comprehensive Review of Technologies, Architectures, and Applications. *Communication In Physical Sciences*, *10*(3), 262-273.
3. Senthilkumar, G., Tamilarasi, K., Velmurugan, N., & Periasamy, J. K. (2023). Resource allocation in cloud computing. *Journal of Advances in Information Technology*, *14*(5), 1063-1072.

4. Singh, V., & Yadav, N. (2023). Optimizing resource allocation in containerized environments with ai-driven performance engineering. *International Journal of Research Radicals in Multidisciplinary Fields, ISSN*, 58-69.
5. Alquwayzani, A., Aldossri, R., & Frikha, M. (2024). Prominent Security Vulnerabilities in Cloud Computing. *International Journal of Advanced Computer Science & Applications*, *15*(2).
6. Hanna, M. G., Pantanowitz, L., Dash, R., Harrison, J. H., Deebajah, M., Pantanowitz, J., & Rashidi, H. H. (2025). Future of artificial intelligence (AI)-machine learning (ML) trends in pathology and medicine. *Modern Pathology*, 100705.
7. Khan, A. R. (2024). Dynamic load balancing in cloud computing: optimized RL-based clustering with multi-objective optimized task scheduling. *Processes*, *12*(3), 519.
8. Alzaabi, F. R., & Mehmood, A. (2024). A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. *IEEE Access*, *12*, 30907-30927.
9. Khan, M. M. I., & Nencioni, G. (2023). Resource allocation in networking and computing systems: A security and dependability perspective. *IEEE Access*, *11*, 89433-89454.
10. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, *12*(6), 1333.
11. Dey, S., Sarma, W., & Tiwari, S. (2023). Deep learning applications for real-time cybersecurity threat analysis in distributed cloud systems. *World Journal of Advanced Research and Reviews*, *17*(3), 1044-1058.
12. Arora, S., & Khare, P. (2024). AI/ML-enabled optimization of edge infrastructure: Enhancing performance and security. *sensors*, *4*(2).
13. Khan, T., Singh, K., Shariq, M., Ahmad, K., Savita, K. S., Ahmadian, A., ... & Conti, M. (2023). An efficient trust-based decision-making approach for WSNs: Machine learning oriented approach. *Computer Communications*, *209*, 217-229.
14. Ang'udi, J. J. (2023). Security challenges in cloud computing: A comprehensive analysis. *World Journal of Advanced Engineering Technology and Sciences*, *10*(2), 155-181.
15. Zakariyya, I., Kalutarage, H., & Al-Kadri, M. O. (2023). Towards a robust, effective and resource efficient machine learning technique for IoT security monitoring. *Computers & Security*, *133*, 103388.
16. Petrovska, I., & Kuchuk, H. (2023). Adaptive resource allocation method for data processing and security in cloud environment. *Advanced Information Systems*, *7*(3), 67-73.
17. Peng, K., Huang, H., Zhao, B., Jolfaei, A., Xu, X., & Bilal, M. (2022). Intelligent computation offloading and resource allocation in IIoT with end-edge-cloud computing using NSGA-III. *IEEE Transactions on Network Science and Engineering*, *10*(5), 3032-3046.
18. Saxena, D., Singh, A. K., Lee, C. N., & Buyya, R. (2023). A sustainable and secure load management model for green cloud data centres. *Scientific Reports*, *13*(1), 491.
19. Oloruntoba, O. (2024). Green cloud computing: AI for sustainable database management. World Journal of Advanced Research and Reviews, 23(03), 3242-3257.
20. Silva, A., & Müller, P. (2024). Machine Learning-Driven Resource Allocation in IoT-Fog-Cloud Networks. *Synthesis: A Multidisciplinary Research Journal*, *2*(3), 11-21.
21. Dritsas, E., & Trigka, M. (2025). A survey on the applications of cloud computing in the industrial internet of things. *Big data and cognitive computing*, *9*(2), 44.
22. Gadde, H. (2022). AI-Enhanced Adaptive Resource Allocation in Cloud-Native Databases. *Revista de Inteligencia Artificial en Medicina*, *13*(1), 443-470.
23. Attou, H., Guezzaz, A., Benkirane, S., Azrour, M., & Farhaoui, Y. (2023). Cloud-based intrusion detection approach using machine learning techniques. *Big Data Mining and Analytics*, *6*(3), 311-320.
24. Lekkala, C. (2024). Ai-driven dynamic resource allocation in cloud computing: Predictive models and real-time optimization. *J Artif Intell Mach Learn & Data Sci*, *2*.
25. Ratnayake, S. (2024). A COMPREHENSIVE REVIEW OF AI-DRIVEN OPTIMIZATION, RESOURCE MANAGEMENT, AND SECURITY IN CLOUD COMPUTING ENVIRONMENTS.
26. Anbarkhan, S. H. (2025). Optimizing Cloud Resource Allocation with Machine Learning: Strategies for Efficient Computing. *Ingénierie des Systèmes d'Information*, *30*(1), 1.
27. Kamble, T., Deokar, S., Wadne, V. S., Gadekar, D. P., Vanjari, H. B., & Mange, P. (2023). Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*, *19*(2).
28. Khan, T., Tian, W., Ilager, S., & Buyya, R. (2022). Workload forecasting and energy state estimation in cloud data centres: ML-centric approach. *Future Generation Computer Systems*, *128*, 320-332.
29. Nuthalapati, A. (2024). Cloud data center performance optimization through machine learning-based workload forecasting and energy efficiency. *International Journal of Science and Research Archive*, *13*(02), 2353-2361.
30. Nassif, A. B., Talib, M. A., Nasir, Q., Albadani, H., & Dakalbab, F. M. (2021). Machine learning for cloud security: a systematic review. *IEEE Access*, *9*, 20717-20735.
31. Abdallah, A. M., Alkaabi, A. S. R. O., Alameri, G. B. N. D., Rafique, S. H., Musa, N. S., & Murugan, T. (2024). Cloud network anomaly detection using machine and deep learning techniques—recent research advancements. *IEEE Access*, *12*, 56749-56773.
32. Kasula, V. K., Yadulla, A. R., Konda, B., & Yenugula, M. (2024). Fortifying cloud environments against data breaches: A novel AI-driven security framework. *World J. Adv. Res. Rev*, *24*, 1613-1626.
33. Barua, B., & Kaiser, M. S. (2024). AI-Driven Resource Allocation Framework for Microservices in Hybrid Cloud Platforms. *arXiv preprint arXiv:2412.02610*.

34. Singh, J., Singh, P., Hedabou, M., & Kumar, N. (2023). An efficient machine learning-based resource allocation scheme for SDN-enabled fog computing environment. *IEEE Transactions on Vehicular Technology*, *72*(6), 8004-8017.
35. Fahimullah, M., Ahvar, S., Agarwal, M., & Trocan, M. (2024). Machine learning-based solutions for resource management in fog computing. *Multimedia Tools and Applications*, *83*(8), 23019-23045.
36. Choudhury, A., & Madheswaran, Y. (2024). Enhancing cloud scalability with AI-driven resource management. *International Journal of Innovative Research in Engineering and Management*, *11*(5).
37. Latha, Y. M., Choudhuri, S., Jannepally, A. R., Bhise, Y. D., Swathi, T., & rao Katta, S. (2024, September). Optimizing Cloud Resource Management Through AI Enhancing Efficiency and Scalability in Modern Computing Environments. In *2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit)* (pp. 1044-1049). IEEE.
38. Li, Y., Zhang, X., Zeng, T., Duan, J., Wu, C., Wu, D., & Chen, X. (2023). Task placement and resource allocation for edge machine learning: A gnn-based multi-agent reinforcement learning paradigm. *IEEE Transactions on Parallel and Distributed Systems*, *34*(12), 3073-3089.
39. Saini, H., Singh, G., Kaur, A., Saini, S., Wani, N. A., Chopra, V., ... & Bhat, S. A. (2024). Hybrid Optimization Machine Learning Framework for Enhancing Trust and Security in Cloud Network. *IEEE Access*.
40. Safi, W., Ghwanmeh, S., Mahfuri, M., & Al-Sit, W. T. (2024, February). Enhancing Cloud Security: A Comprehensive Review of Machine Learning Approaches. In *2024 2nd International Conference on Cyber Resilience (ICCR)* (pp. 1-10). IEEE.
41. Malipatil, A. R., Paramasivam, M. E., Gulyamova, D., Saravanan, A., Ramesh, J. V. N., Muniyandy, E., & Ghodhbani, R. (2025). Energy-Efficient Cloud Computing Through Reinforcement Learning-Based Workload Scheduling. *International Journal of Advanced Computer Science & Applications*, *16*(4).
42. Saxena, D., & Singh, A. K. (2021). A proactive autoscaling and energy-efficient VM allocation framework using online multi-resource neural network for cloud data center. *Neurocomputing*, *426*, 248-264.
43. Tuli, S., Gill, S. S., Xu, M., Garraghan, P., Bahsoon, R., Dustdar, S., ... & Jennings, N. R. (2022). HUNTER: AI based holistic resource management for sustainable cloud computing. *Journal of Systems and Software*, *184*, 111124.
44. https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/
45. Zhang, T., Lin, W., Vogelmann, A. M., Zhang, M., Xie, S., Qin, Y., & Golaz, J. C. (2021). Improving convection trigger functions in deep convective parameterization schemes using machine learning. *Journal of Advances in Modeling Earth Systems*, *13*(5), e2020MS002365.
46. https://www.kaggle.com/datasets/mrjkanderson/resourse-utilization