2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) | 979-8-3503-1920-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSES60034.2023.10465563

Enhancing Cloud Computing Resource Allocation with LSTM-Based Predictive Modeling

M Jananee[1], K. Nimala[2]* and R. Nareshkumar[3]

Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai.

E-mail :jm7168@srmist.edu.in, nimalak@srmist.edu.in andnr7061@srmist.edu.in

Abstract-Cloud resource demands have become more complicated as big data and AI have progressed, with characteristics such as being abrupt, arriving in bunches, and diversified. This has led to resource allocation lagging far behind the requests and an uneven resource utilization that wastes resources. Predicting when resources will be needed is done so that apps can get their hands on them right when they're needed, satisfying their dynamic resource requirements. Maintaining uniformity in service provision is challenging because of fluctuating consumer demand. This research presents a proactive approach to cloud computing resource allocation using LSTM to predict resource demands. In particular, this approach suggests an LSTM prediction method that enhances the accuracy of resource request predictions and then constructs a Round Robin resource allocation algorithm that reduces allocation latency and evenly distributes the various types of resources available on a physical machine.

Keywords—LSTM, Deep learning, VM, Energy, Load balancing.

I.INTRODUCTION

The method of organising, controlling, and accumulating jobs in a production process is referred to as job scheduling. Scheduling can become entangled within incidental variables such as causal processing durations and desultory due dates; this type of scheduling is sometimes referred to as stochastic scheduling. In order to predict the future request for the next day, week, month, or year, we need to look at an example of sequential data, such as climate or stock market data. In contrast to conventional neural networks, LSTM features a feedback connection. Weather, stock market, audio, and video are only some examples of the types of sequential data it can anticipate. There is no time restriction on when or when you can access the features for scheduling jobs, and you can do it at any time. Everyone who is participating in the work will be able to observe the procedure. The scheduling system processes the job, with the remaining to be done during the day-to-day process, resulting in a reduction in the amount of time as well as increases in both productivity and efficiency. Increasing both productivity and efficiency in the production process while using fewer people will be possible if work and workloads are optimised. First-come, first-served scheduling has been implemented for this project. The first task has the opportunity to finish the work first, while the subsequent duties are required to wait until the earlier tasks are finished. Starvation is the primary risk associated with FCFS. It will take less time to finish the job if the time it takes to arrive is cut down.

The fact that the individual who arrives first is always assigned to the same environment is the most significant drawback of the FCFS system. Using the Shortest-Job-First (SJF) scheduling method will select the procedure that calls for the fewest amount of resources. One of the drawbacks of utilising SJF is that the longest resource requirement is postponed until all of the shorter requirements have been finished before it is started. This is because only the shortest resource requirement is run first when using SJF. It is difficult to have an accurate understanding of the requirements of the process in advance. When using priority scheduling, tasks with the lowest resource requirements must wait while those with the greatest priority requirements have access to those resources more frequently. It is possible that the job with the lesser priority will be disregarded. A scheduling mechanism known as a Round Robin ensures that each activity receives a same amount of time to utilise the available resources.

As a consequence of this, all jobs are finished effectively within an average time limit, and the efficiency with which resources are utilised is improved. As a consequence of this, the suggested system makes use of the method of resource allocation known as Round Robin.

II.RELATED WORK

Prediction-based Energy Conserving Resource Allocation (ECRASP) was developed by Wang et al. [1] and implemented on the cloud. Specifically, the ECRASP is made up of two parts: the prediction mechanism and the job allocation mechanism. The prediction method helps determine whether incoming jobs will be sparse or dense, allowing for more efficient resource management. Incoming jobs are distributed among available PMs through the allocation system. However, if the system is already heavily utilised, the allocation mechanism will start the new PM to process the incoming work rather than shutting down idle PMs (i.e. weakly loaded).

To improve resource utilisation without introducing additional virtual machines, Saraswathi et al. [2] proposed a dynamic model of incomes specification within cloud computing. This is a novel approach to VM specification based on features (i.e. job priority). The VM obtainability is evaluated constantly as new tasks are added. Additional jobs are scheduled to run based on the availability of virtual machines (VMs), but will not run if there are none. The algorithm then selects the least critical task (taking into account the task's occupancy type) and stops running for a period, saving money by doing so (a process known as "incomes fore stalling"). Therefore, the low-priority activity can be completed and resources saved for the high-priority task. If the task's occupancy type is suspend able, work can be paused and resumed once all waiting tasks have been completed by virtual machines. However, if the occupation type of the task cannot be paused, the work can be resumed when the least important task execution has been completed using the task revenues. Any new jobs are handled in the same fashion. The suggested architecture provides obvious simplicity by carrying out all necessary steps to create new virtual machines.

ACO-based cloud computing load balancing was proposed by Padmavathi and Basha et al. [3] Bio-inspired algorithms that mimic biological ants' behaviours were suggested. Ants use trail fragrance to describe their food-seeking trek. Ants migrate by finding the shortest pheromone route. Cloud data centres load balance VMs using Dynamic and Elastic Ant Colony Optimisation Load Balancing (DEACOLB). The experiments show that the average Make Span (finish time) is lower than ACO and FCFS. Increased employment lowers the average Make Span.

To optimally and quickly serve Sudden and Urgent (SSU) resource needs, Chen et al. [4] suggested a cloud resource allocation technique. The proposed method involves allocating resources in accordance with a priority system that takes into account both client and resource urgency. The virtual machines are then assigned to real hosts in an efficient manner by employing a multi-objective optimisation method. In a controlled experiment, three different approaches to allocating available resources—the proposed SSU technique, Round-Robin, and Best-Fit (BF)—were put to the test.The results of the tests showed that the presented technique assigns VMs to PM based on matching resource distance, while BF allocates them to the PM with the lowest CPU consumption, and Round-Robin allocates them in a circular ordered and equal number.

Rengasamy and Chidambaram et al. [5] provided a novel predictive methodology to properly allocate resources while dealing with the challenges of VM migration and placement in cloud computing. The proposed method uses the Random Algorithm, which randomly pairs jobs (cloudlets) with servers to process a large workload evenly. Each user receives a list of accessible servers via the Random Algorithm, eliminating the need for a centralised broker. The evaluation results explored remarkable performance in the cloud and shown that the proposed technique can balance the load in seconds rather than minutes.

In order to distribute and release cloud data centre resources on demand, Bhardwaj et al. [6] established an autonomous resource allocation model. The suggested solution redistributes virtual machines on the server side to lessen the amount of time users must wait for cloud services, with the goal of improving response time and VM utilisation. Horizontal scaling (scale-out) was utilised to accommodate increased VM allocations in response to demand. If the virtual machines are underutilised or overutilized, the proposed method will increase the number of virtual machines (VMs) until the average response time reaches a predetermined goal. In addition to efficiently allocating VMs in response to demand, the results show that the suggested method also increases the number of VMs dynamically throughout operation, which decreases the length of time a request spends in the queue.

Two energy-efficient resource allocation algorithms for use in cloud data centres were presented by Than and Thein et al. [7]. DSJF is the first suggested method, and it combines the Shortest Job First (SJF) resource allocation algorithm with the Dynamic Voltage and Frequency Scaling (DVFS) power management approach and the Cupic power model. DFCFS is a resource allocation method that combines the DVFS and FCFS approaches with the Cupic power model. Using the CloudSim simulator, the power consumption of the suggested algorithms was analysed and rated. Research indicates that DSJF is preferable than DFCFS due to its ability to cut energy consumption by up to 55%.

To combat the challenge of resource scheduling in the cloud, Khodar et al. [8] introduced a genetic algorithm-based load-balancing scheduling technique. One of the main selling points of the genetic algorithm is that it can test out a wide variety of approaches before arriving on the optimal one.The suggested method prioritises the present state and past data to quantify its impact on system load, when sufficient VM resources have been assigned in advance. Then, the operators of crossover and mutation will choose the distribution with the least detrimental effect. The test results demonstrated that the proposed strategy approach improves load balancing when historical data is taken into account and that it decreases the need for dynamic migration.

An enhanced method for cost-based scheduling was also presented in [9]. Planning groups of jobs on a cloud computing platform with resources that charge for different types of resources and conduct different types of computations is the main focus of this tool. Communication between jobs and resources improves the computation/communication ratio when jobs are clustered. This technique measured both computer efficiency and the number of resources used. By combining different jobs at the time of execution, efficiency was increased, as was the transmission of data between them. In most cases, the capabilities of numerous resources and their processing are evaluated before tasks are combined. CloudSim was used to perform the simulation.

Linear programming was proposed by Akintoye and Bagula et al. [10] to establish the cloud atmosphere's resource allocation problem, and they built a Binding Policy Based on the Hungarian Algorithm (HABBP) to optimise it. Tasks can be completed more quickly using HABBP because a load balancing policy is used to link cloudlets to the right virtual machines. Both (1) to demonstrate a unique binding approach based on the HABBP algorithm and (2) to provide an interactive interface for configuring cloudlets parameters, they submitted HABBP modules for inclusion in CloudSim. When compared to CloudSim's standard binding strategy, the HABBP speeds up the execution of tasks. It also demonstrated that the HABBP approach might optimise cloud-based virtual machine (VM) allocation and solve this problem.

The effects of virtualization on energy usage in the cloud were studied by Atiewi et al. [11]. The results of their experiment using a PSSA in a green cloud simulator contrasting virtualized and non-virtualized servers were presented. The comparison is based on three metrics: datacenter workload, total energy usage, and Make Span. The findings demonstrated that (1) virtualized datacenters perform better than non-virtualized ones when it comes to data centre load, (2) non-virtualized servers consume more energy than their virtualized counterparts, and (3) datacenters with non-virtualized servers performed better when it came to Make Span than datacenters with virtualized servers did.

Using double-fitness load balancing and the Job Completion Cost Genetic Algorithm (LCGA), Yin et al. [12] proposed a powerful approach to work scheduling. The proposed method for optimising cloud-based task scheduling aims to (1) maximise resource utilisation while satisfying user demands at the lowest possible job execution cost. To meet the needs of service providers, it is necessary to: (2) Offer load balancing to ensure proper job allocation in a dynamic cloud setting. They tested the proposed optimised approach to two other algorithms—the Load balancing Genetic Algorithm (LGA) and the task completion Cost Genetic Algorithm (CGA)—to ensure its efficacy. The comparison results showed that while LGA is successful at load balancing, it does not acquire the influence of task completion cost. The results also highlighted the fact that CGA results in the lowest task completion cost but has no discernible effect on load balancing. The results of the LCGA showed that optimal task scheduling achieves both load balancing and minimal completion costs.

In their paper, Deepika and Rao et al. [13] examined how virtualization could be used to enable dynamic cloud resource allocation. By moving VMs from a busy (hot) server to a less active (cold) server, the suggested method efficiently distributes many virtual resources. The "skewness" method was used to calculate server resource utilisation, which involves looking at historical data to predict future needs. Through optimising workloads (meeting VM requests depending on server capacity) and green computing (saving server energy by optimising unnecessary server utilisation), the proposed solution enables dynamic resource allocation.

III.METHODOLOGY

In cloud computing, users have the flexibility to increase or decrease their resource use as required. Virtualization technology allows for the multiplexing of resources between physical and virtual machines. As a result, in an ARM system that use virtualization technologies for dynamic resource allocation, the fundamental purpose for each physical machine is to prevent overloading.

A system that makes use of virtualization technology to enable green computing by lowering the number of servers in operation and to dynamically allocate data centre resources in response to application demands. To ascertain which physical machine will be overworked next, a powerful load-prediction algorithm will be employed. Traditional queuing analysis rarely takes into account systems of this magnitude, yet a cloud data centre can have thousands or even hundreds of thousands of facility (server) nodes. There could be a great deal of variety in the time it takes to complete a task. Given the ever-changing nature of cloud settings, the variety of user requests, and the time-dependent nature of load, data centres in the cloud must maintain a consistent level of service quality despite significant fluctuations in demand

These restrictions apply to currently used methods
The lower priority jobs will have to wait a lengthy period due to the priority and rank-based scheduling algorithms.

- The energy consumption time must be known before an execution begins in an energy-aware scheduling technique.
- Genetic algorithm based resource allocation is not optimal in all settings.
- The time lag procedure of a scheduling method was not discussed.

Therefore, a different approach is needed to ensure that all requests and resources are treated with the same importance.

In this research, we demonstrate how SaaS providers might use LSTM [14] to create a cloud-based workload prediction module (see Figure 1). Using real-world web server request logs, LSTM [15] was put to the test to see how accurately it could forecast future workload. According to the forecasted demand, servers will be distributed primarily through cloud sim. Results from simulations show that our approach is able to accurately predict the future need for requests based on historical data, allowing for optimal utilisation of available resources at no additional expense. However, it can be challenging to achieve the QoS with an economically viable quantity of resources due to the time-varying nature of workloads. Priority-based server allocation will help us free up resources when they are no longer needed.

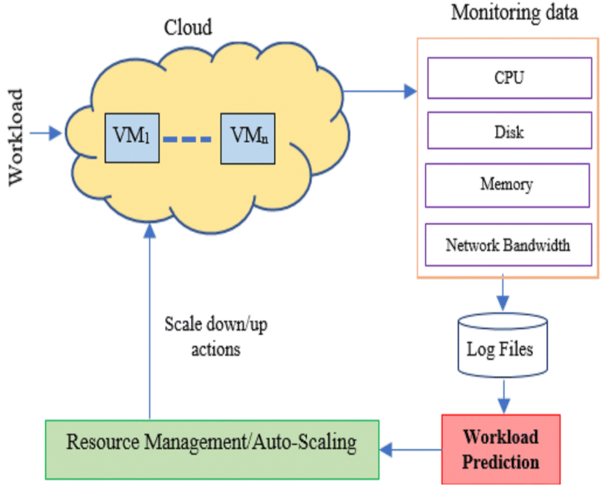


Fig. 1 Architecture diagram for the proposed system

To create cloud system components [16] (services, host, data centre, broker, VMs), communicate between components, manage the simulation clock, and queue and process events, the lowest layer relies on the core functionalities provided by the discrete event simulation engine (SimJava).

In order to avoid overprovisioning or under provisioning, the suggested method allocates resources to the user, checks the utilisation at regular intervals, and changes the allocated resources accordingly. The Cloud Service Provider (CP) can take the user's first request and immediately begin allocating the necessary resources [17]. Virtual machines (VMs) are made accessible and assigned on demand based on a mapping between user requirements and the number of VMs necessary. A virtual machine (VM) is an emulated machine that functions identically to the underlying physical machine. As many virtual machines (VMs) as can be supported by a single physical computer are possible. A computer with 16 GB of Memory, an octa-core processor running at 3.4 GHz, 4 GB of VRAM, and a RAID array of four 1 TB hard drives can run five low-end virtual machines with little trouble. Severalvirtual machines (VMs) can be created for a single user if necessary. When allocating a brand new virtual machine (VM), time is of the essence. When a user requests a fresh virtual machine (VM), the process can take anywhere from one to four minutes.

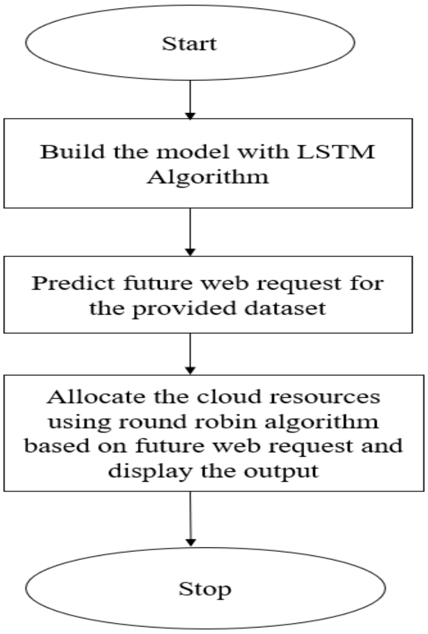

Start
Build the model with LSTM Algorithm
Predict future web request for the provided dataset
Allocate the cloud resources using round robin algorithm based on future web request and display the output
Stop

Fig. 2 Flow chart of the proposed Algorithm

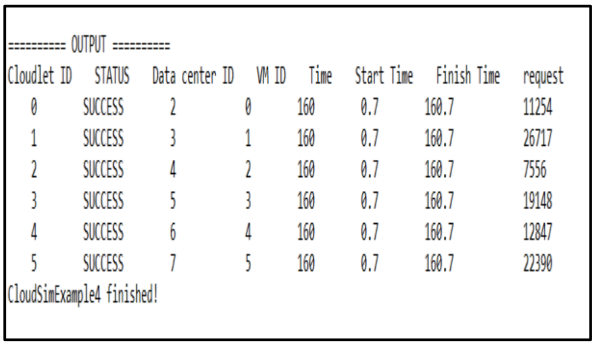
In this flowchart Fig 2, we have suggested two algorithms: the LSTM algorithm for forecasting future web requests, and the Round Robin algorithm for allocation of resources. Here, we have utilised the Wikipedia dataset to forecast the web requests for seven different languages on that specific date, and the round robin algorithm has been utilised in cloudsim to determine which cloudlets should be allocated. The predicted output was utilised as an input in the cloudlet assignment process.

IV. Results and Discussion

Experiment has been setup using cloud Sim simulator where Wikipedia dataset has been used for prediction of future web request using Lstm Algorithm. Table1 displays the results of the trained model. The model used two dense layers in its prediction procedure, and it was trained using a weekly cycle. We settled on using Date and Views as model characteristics. And if there is a pattern, it will be implemented as a new function. In this case, we've extrapolated from a sample of pageviews. Users will save money thanks to Cloudsim, which will be used to distribute datacenter resources in anticipation of future requests. It has used its training data to project that it will receive this many requests over the next upcoming days. The date has been passed as input and future request for different languages on that particular date has been printed. Based on the output allocation of vm and datacenter has been done for first language first datacenter and first vm has been allocated using round robin algorithm. The experiment starts from predicting future request and then follows the allocation policy. The output shows the allocated vmid, datacenter, cloudletid, starttime, finishtime, request as shown in figure 3. Since the output has differentlanguages, we allocated a separate datacenter and separate vm for each language and separate cloudletid for each language based on all the available packages in cloudsim. For vm allocation, vmallocation policy has been used, and for datacenter their respective package datacenter and datacenter broker have been used.

Table 1 Trained model Summary

Layer (type)	**Output Shape**	**Param #**
Lstm (LSTM)	(None, 7, 400)	652800
Lstm_1 (LSTM)	(None, 7, 500)	1802000
Lstm_2 (LSTM)	(None, 500)	2002000
Layer1 (Dense)	(None, 700)	350700
Layer2 (Dense)	(None, 100)	70100
Dense (Dense)	(None, 7)	707
Total params: 4,878,307 Trainable params: 4,878,307 Non-trainable params: 0		



```
========== OUTPUT ==========
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time   request
    0        SUCCESS        2             0      160      0.7         160.7        11254
    1        SUCCESS        3             1      160      0.7         160.7        26717
    2        SUCCESS        4             2      160      0.7         160.7        7556
    3        SUCCESS        5             3      160      0.7         160.7        19148
    4        SUCCESS        6             4      160      0.7         160.7        12847
    5        SUCCESS        7             5      160      0.7         160.7        22390
CloudSimExample4 finished!
```

Fig. 3 Output for allocated resources for future request

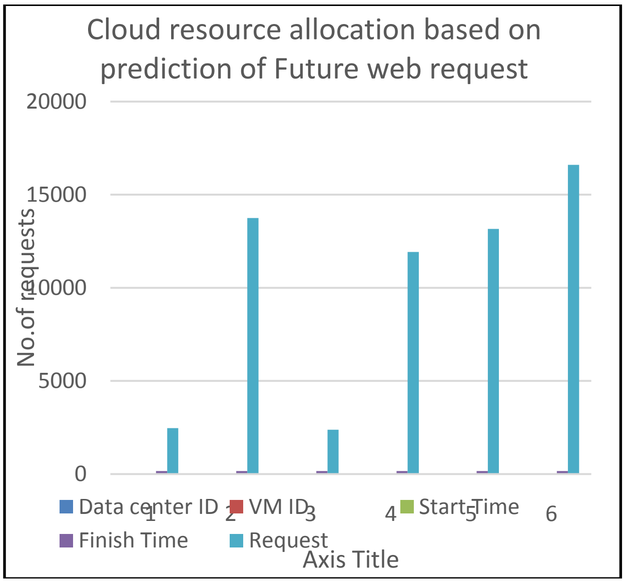


Fig. 4 Allocation of resources based on request

In the above fig.4 the datacenter and Vm has been allocated for the request with the time execution from start to end. The VmId and datacenter has been submitted to the request from 0 to 7.Since start time and end time are short it has been ended at the zero level. The vertical line represents the number of requests and horizontal line represents their respective Id.

V.CONCLUSION

In this paper cloud resources have been allocated based on prediction of future web request so that the cost can be minimized by prior allocation of resources. The resouces has been allocated priority for that based on prediction. This will make sure the resources are fully utilized and on demand cost can be avoided by this method. Using Round robin algorithm allocation has been done and in future deallocation of this resources can be done. This method is more efficient since it predicts the future workload and does the allocation part also by this way two works are done simultaneously.

REFERENCES

[1] Wang CF, Hung WY, Yang CS. A prediction based energy conserving resources allocation scheme for cloud computing. In2014 IEEE International Conference on Granular Computing (GrC) 2014 Oct 22 (pp. 320-324). IEEE.

[2] Saraswathi AT. a, Kalaashri. Y. RA b, Dr. S. Padmavathi,“Dynamic Resource Allocation Scheme in Cloud Computing”.

[3] Rajasegarar S, Leckie C, Palaniswami M, Bezdek JC. Quarter sphere based distributed anomaly detection in wireless sensor networks. In2007 IEEE International Conference on Communications 2007 Jun 24 (pp. 3864-3869). IEEE.

[4] Chen J. A cloud resource allocation method supporting sudden and urgent demands. In2018 Sixth International Conference on Advanced Cloud and Big Data (CBD) 2018 Aug 12 (pp. 66-70). IEEE.

[5] Rengasamy R, Chidambaram M. A novel predictive resource allocation framework for cloud computing. In2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) 2019 Mar 15 (pp. 118-122). IEEE.

[6] Bhardwaj T, Upadhyay H, Sharma SC. Autonomic resource allocation mechanism for service-based cloud applications. In2019 international conference on computing, communication, and intelligent systems (ICCCIS) 2019 Oct 18 (pp. 183-187). IEEE.

[7] Than MM, Thein T. Energy-saving resource allocation in cloud data centers. In2020 IEEE Conference on Computer Applications (ICCA) 2020 Feb 27 (pp. 1-6). IEEE.

[8] Khodar A, Al-Afare HA, Alkhayat I. New scheduling approach for virtual machine resources in cloud computing based on genetic algorithm. In2019 International Russian Automation Conference (RusAutoCon) 2019 Sep 8 (pp. 1-5). IEEE.

[9] Selvarani S, Sadhasivam GS. Improved cost-based algorithm for task scheduling in cloud computing. In2010 IEEE International Conference on Computational Intelligence and Computing Research 2010 Dec 28 (pp. 1-5). IEEE.

[10] Akintoye SB, Bagula A. Optimization of virtual resources allocation in cloud computing environment. In2017 IEEE AFRICON 2017 Sep 18 (pp. 873-880). IEEE.

[11] Atiewi S, Abuhussein A, Saleh MA. Impact of virtualization on cloud computing energy consumption: Empirical study. InProceedings of the 2nd International Symposium on

Computer Science and Intelligent Control 2018 Sep 21 (pp. 1-7).

[12] Yin S, Ke P, Tao L. An improved genetic algorithm for task scheduling in cloud computing. In2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA) 2018 May 31 (pp. 526-530). IEEE.

[13] Deepika T, Rao AN. Active resource provision in cloud computing through virtualization. In2014 IEEE International Conference on Computational Intelligence and Computing Research 2014 Dec 18 (pp. 1-4). IEEE.

[14] R. Nareshkumar and K. Nimala, "An Exploration of Intelligent Deep Learning Models for Fine Grained Aspect-Based Opinion Mining," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-7, doi: 10.1109/ICSES55317.2022.9914094.

[15] R. Nareshkumar and K. Nimala, "Interactive Deep Neural Network for Aspect-Level Sentiment Analysis," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10083812.

[16] M. Jananee and K. Nimala, “Allocation of cloud resources based on prediction and performing auto-scaling of workload,” 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Jan. 2023, doi: 10.1109/iceconf57129.2023.10083865.

[17] M. Omkar and K. Nimala, “Machine Learning based Diabetes Prediction using with AWS cloud,” 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Jul. 2022, doi: 10.1109/icses55317.2022.9914160.