Rajkumar and Katiravan
Journal of Cloud Computing (2023) 12:138
https://doi.org/10.1186/s13677-023-00514-7

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Virtualized intelligent genetic load balancer for federated hybrid cloud environment using deep belief network classifier

S. Rajkumar[1*] and Jeevaa Katiravan[1]

Abstract

Load balancing is major issue in federated cloud environment. Various services can be offered by different cloud service providers. As per current working environment cloud computing is used in major applications such as education, online shopping, multimedia services, etc. Dynamic load balancing is required to handle the resources. Federated cloud has various services offering system with computing resources, resource pooling, internet access services and storage. Intelligent Genetic algorithm is proposed to provide efficient load balancing service in hybrid cloud environment. Virtualized Intelligent Genetic Load Balancer algorithm consists of load balancer and resource provisioning system to allocate the resources. Enhanced Load Balancer is used to preserve the load and minimize the span time based on resource provisioning method. In this work we analyse automated virtual machine services by using runtime resource provision. Here we use enhanced load balancer to measure the performance using virtual machine placements, resource utilization and automated quality requirements. We design a deep belief network based on requirements and measure the accuracy using TensorFlow. The simulation results test the accuracy and compare the results. Virtualized Intelligent Genetic Load Balancer system is achieving the accuracy of 95% based on overall capacity requirements. We compare Virtualized Intelligent Genetic Load Balancer system performance with existing simulations results and compared the results.

Keywords Enhanced load balancer, Deep belief networks, Virtual machine, Cloud service provider, Intelligent genetic algorithm

Introduction

Cloud Service Provider (CSP) is offered various cloud services to the user based on requirements. Commercially various CSPs are available based on the requirements. For example Amazon Web Services are providing single touch application service for user and they can access the resources like storage or application level services [1]. Google Provides virtual provisioning based services such as driver for storage, docs for applications, meet of video conferencing services. Also various online platforms are available for booking, ordering, ticketing and shopping [2]. In this case during seasonal or festival time various CSPs are used to provide uninterrupted service to the user. So we need efficient load balancing system for handling these issues [3].

On demand servicing is another issue while implementing federated cloud services. In this case services are offered from remote location [4], access the resource from anyplace while sitting once place with the help of internet [5] and delivery of services [6]. The services are classified as infrastructure, platform and application. It is internet based access model so service level offering need to considered for implementing cloud [7, 8]. User can access the data or resource from cloud pool and verify

*Correspondence:
S. Rajkumar
rajkumar.document1980@gmail.com
[1] Velammal Engineering College, Chennai-66, India

the access privileges based on the requirements while mutual agreement stage [9].

As per cloud surveys, it is not required for physical location access, computing configurations, storage level access and administrative privileges. While setting cloud environment we need to consider the features such as virtualization [10], on demand service [11], pay for use model [12], geographic access [13] and service levels [14]. So internet is major key player for implementing services with dynamic features and reliable service [15]. Figures 1 and 2 show that federated cloud environment access with load balancing access. In this paper we measure the performance of on demand access privilege with new featured services based on usage.

Based on above representation various cloud user can access resource from CSP. In this case we need to consider various service agreement categories such as ranking, predicting user access, profile agreement, cloud broker representations and services. While implementing this at the same multiple users can the CSP means we need effective load balancer to handle it. So our objective is to create effective load balancer to handle the above issue and propose the method to analyze the performance. In this paper, Sect. 2 explains various related works, Sect. 3 gives proposed intelligent approach for handling load balancer, Sect. 4 test the experiments using proposed algorithm, Sect. 5 gives experimental setup and comparison of existing methods and Sect. 6 propose conclusion and future work.

Related works

IBM, VMWare, Amazon and Google are service offering system and provide access level services. Manikandan et al., gives information above Gmail is providing internet mail service and access the resource from GApp. Private cloud model is available for selecting small range of organization and company specific usage [16]. Wong et al., Fully integrated system is available to replace existing networks, intranet access, storage and infrastructure. Amazon and Google provides hybrid mode of cloud service providers with existing IT infrastructure.

Runga et al., cloud deployment model is enables personalized cloud space, configuration of home network optimization and peer-to-peer cloud computing. The dynamic nature cloud space assisting service is required for scaling the networks. In this case, we considered individual service offering system, networking components, geographically accessing modes and data centre optimization [17].

Quang et al, load balancing is handled in various virtual machines which select major priority on task and scheduling. Here the low level task can be allotted to separate broker like cache optimizer. In this case we need consider service operator, storage level, agreements and virtual provision.

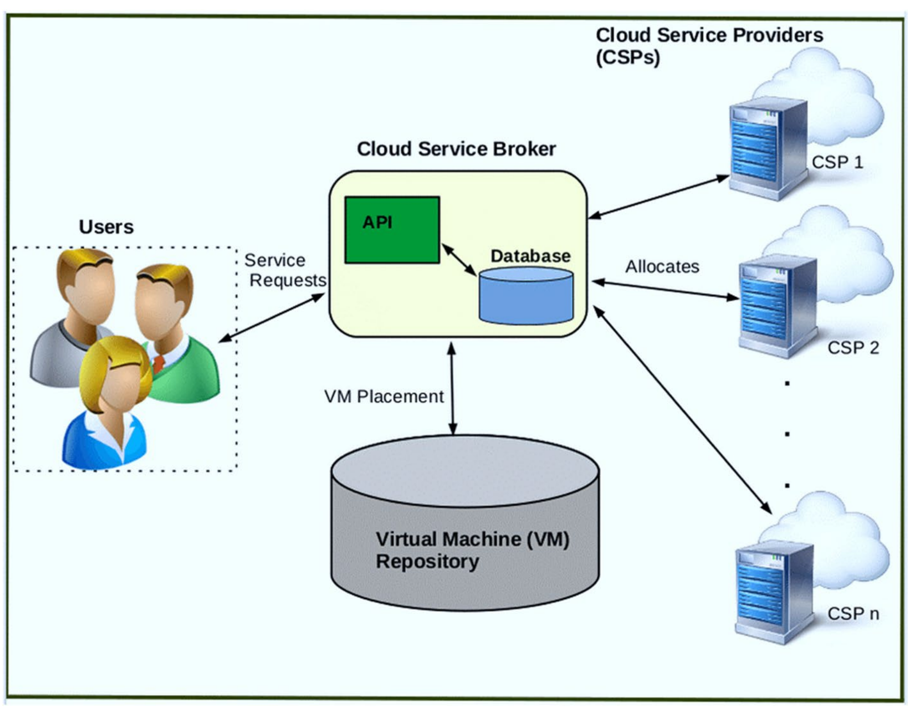


Fig. 1 Cloud User service level management with Broker service using CSP

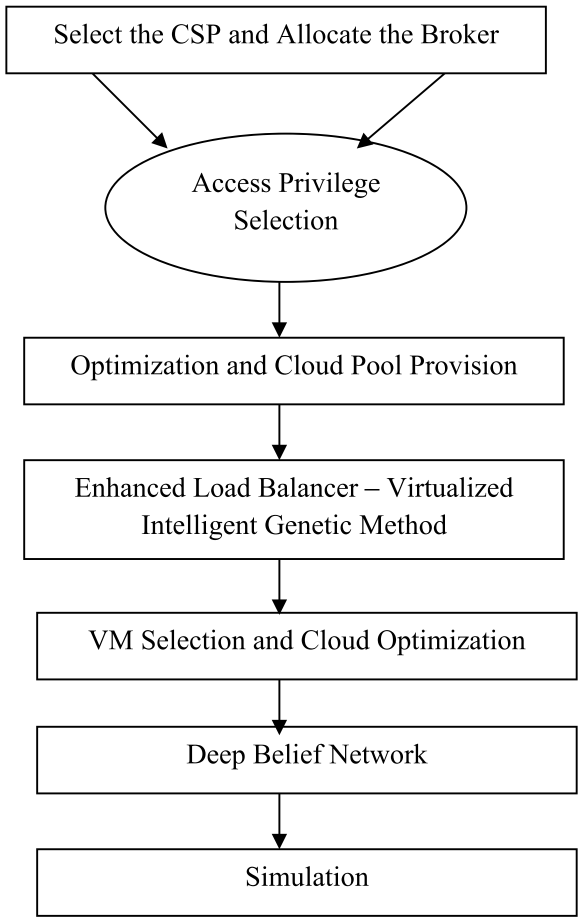


Fig. 2 Flow Chart for Proposed Cloud Optimizer Simulation

Large scale computing or infrastructure is another issues and it can be affected the efficiency. Ant bee virtual provisioning method is proposed by Zinga et al and it is update the feature by dynamic nature. Yung et al, dynamic hill climbing method is used to measure each node transaction results and measure the throughput based on fitness function. Circular based round robin method is used to measure the QoS and each virtual machine selected by using load balancer. So load balancing is major key factor balancing the hybrid cloud and to increase the utilization factor [18].

Virtual machine is to set the federated cloud means we need effective load balancer. As the various studies we propose the enhanced intelligent load balancer for handling VMs and selecting priority based resources from cloud pool. It is a cluster based approach to measure the response time of each node and finds the resource pool [19]. We implement hybrid model for resource utilization and waiting time calculation of each task. So we propose genetic deep belief network model for select-wait-stop based system [20].

Problem statement – load balancing

Load balancing are crucial are often each static and dynamic. Static schemes don't use the gadget info and are a lot of less complicated whereas dynamic schemes can deliver further charges for the device however will modifications because the gadget fame changes. A dynamic scheme is employed right here for its flexibility. The model encompasses a main managing resource and balance the load. Thus, the dynamic control has very little impudence on the alternative running nodes. The system standing then affords a basis for picking the correct load balancing method. The load balancing version given during this story is aimed toward the general public cloud that has numerous nodes with distributed computing assets in several completely different geographic locations. Thus, this model divides the public cloud into several cloud partitions.

Once the environment may be terribly large and complex, these divisions modify the burden balancing. The cloud encompasses a primary controller that chooses the acceptable walls for incoming jobs whereas the balancer for every cloud partition chooses the fantabulous load balancing approach. Trust means that security and privacy and to require the chance at any cost. Cloud shared many sorts of distributed resources to completely different organizations, thence establishing trust between Cloud users and Cloud Service Providers. It's a really huge issue during a Cloud Environment. The amount of tools and models are developed to resolve these problems in distributed systems.

The usually used trust models are Cuboid Trust, chemist Trust, Bayesian Network based mostly Trust Management, cluster Rep, AntiRep, Linguistics net world trust, Peer trust, and so forth however industries don't seem to be settle for the higher than models thanks to lack of standardization and bury operability. The analysis method regarding trust and cloud models are forever inter dependent and to live the safety strength and trust values are their attributes. Cloud Security Alliance service are the challenges and used to assess security of services in cloud model.

The assorted cloud trust mechanisms are used make sure the security and privacy of the users accessing the services. Names supported trust, SLA verification, Policy, proof are the number of the prevailing trust ways within the cloud. Policy based mostly trust may be a real formal trust mechanism utilized in Public Key Infrastructure (PKI). Trust and Reputation are in several and it's involving computation of every trust. The trust reputation is that the collective opinion of a community towards that entity and classified into express trust and indirect trust. Express trust is that the most causative issue for trust computation and that they are wide utilized in e-commerce and P2P networks. A Service Level Agreement(SLA) may be a legal contract between a cloud user and cloud service provider. It's a vital basis of trust management for cloud computing. Obvious based mostly trust, the expected behaviour on the evidence regarding the trustee's attributes of Competency, goodwill and integrity. The federate cloud is the development of multiple internal and external cloud computing services to match the business models and needs. This is services that perform a standard action and multiple cloud service suppliers are operative along in federation.

This is often inter-cloud relationship model and interconnecting the cloud computing environments with additional service providers for managing load balancing, traffic and demand. The service providers could demand the resource supported consumers' agreement and resource management. Federation solves several issues by providing one authority for sign on. The user merely enters the system where they enter it, is documented via the remote system, and is given a federate token that permits them to traverse the network freely, as long because the server they want to access trusts and respects the authority of same federate token.

Enhanced load balancer – virtualized load balancer with optimized cloud pool selection

Load balancing is the important factor in hybrid cloud. It is a mechanism to allocate resource to VM in dynamically from CSP or Cloud resource pool. CSP is having major issue in selecting load balancer and optimizing the cluster. We propose virtualized intelligent genetic algorithm with measuring response time, turnaround time, resource utilization, tolerance against fault and improved in accuracy. We generate enhanced load balancer by using below pseudo code,

Conditions

Load Balancer:

> Case 1: Static Mode – Cloud Environment in stable mode so we fix the cloud services are in constant mode – Compute, Network, Storage are in Constant. We couldn't change their allotted resource.
> Case 2: Dynamic Mode – In this case we can change the requirements based on resource provisioning results. Ii is selected based on demand and elastic property.

We select Virtualized Intelligent Genetic Load Balancer method with static and dynamic based on the request

The above Fig. 3. show that how the load balancer can handle the data centre inputs and optimize the virtual machine. Here we gives the algorithm of balancing the load in virtual machine based on resource provision selected by user.

```
Begin
    Select -> resource_pool()
    Load the CSP/VM Centres
    If(resources == available(n))
       calculate connected_VM()
       select VM()
       return Resource(CSP)
     Else
       Select next()
End
```

Algorithm 1. Load Balancer – Optimizer of CSP

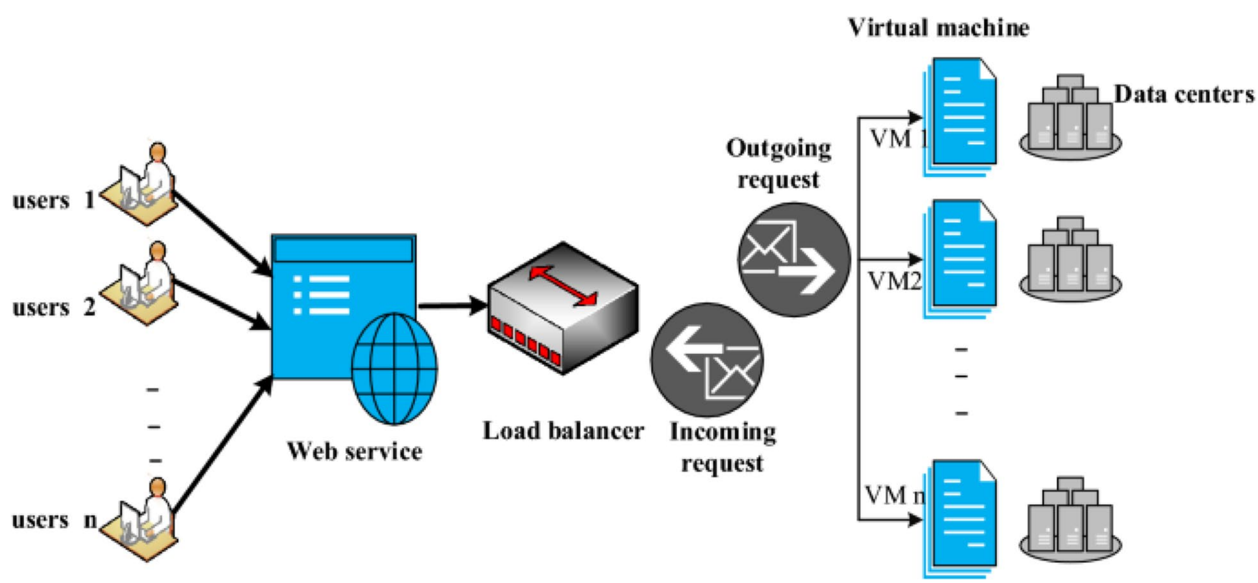


Fig. 3 Load Balancer Optimization – Request from User and Service from CSP/VMs

Virtualized Intelligent Genetic Load Balancer load balancer connect the resource from resource pool and the resource is available means we can set VM. This case once VM is selected means next step to allocate the resource from the user request. The user request can be classified by using Deep belief network classifier. Because it is virtual handler to classify the request as infrastructure, application and platform level services. Then service level agreement is created based on demand if it is required means it can be elastically increased. Our enhanced load balancer transfer the task from one data centre to another VM or CSP. We used virtualized intelligent genetic algorithm with load values and their weight. It is allocated based on Deepq round robin process. Finally the execution and turnaround time is calculated. The below Fig. 4 shows that allocation of loads with resources, completion time, performance index. It is high priority task allocation model with lesser time execution.

Experimental setup – load balancer

Virtualized Intelligent Genetic Load Balancer algorithm can schedule the resource based on user request. Here we are selecting the virtual machine, allocating resources-based resource pool information, data centre for storage and risk assessment. In this paper we implement resource provisioned access control module and simulate the system using Deep belief network optimizer. Resource pool is allocating the resource based on programmed results and access values. The request is sent to broker and apply classification. Here the sources are classified and stored in

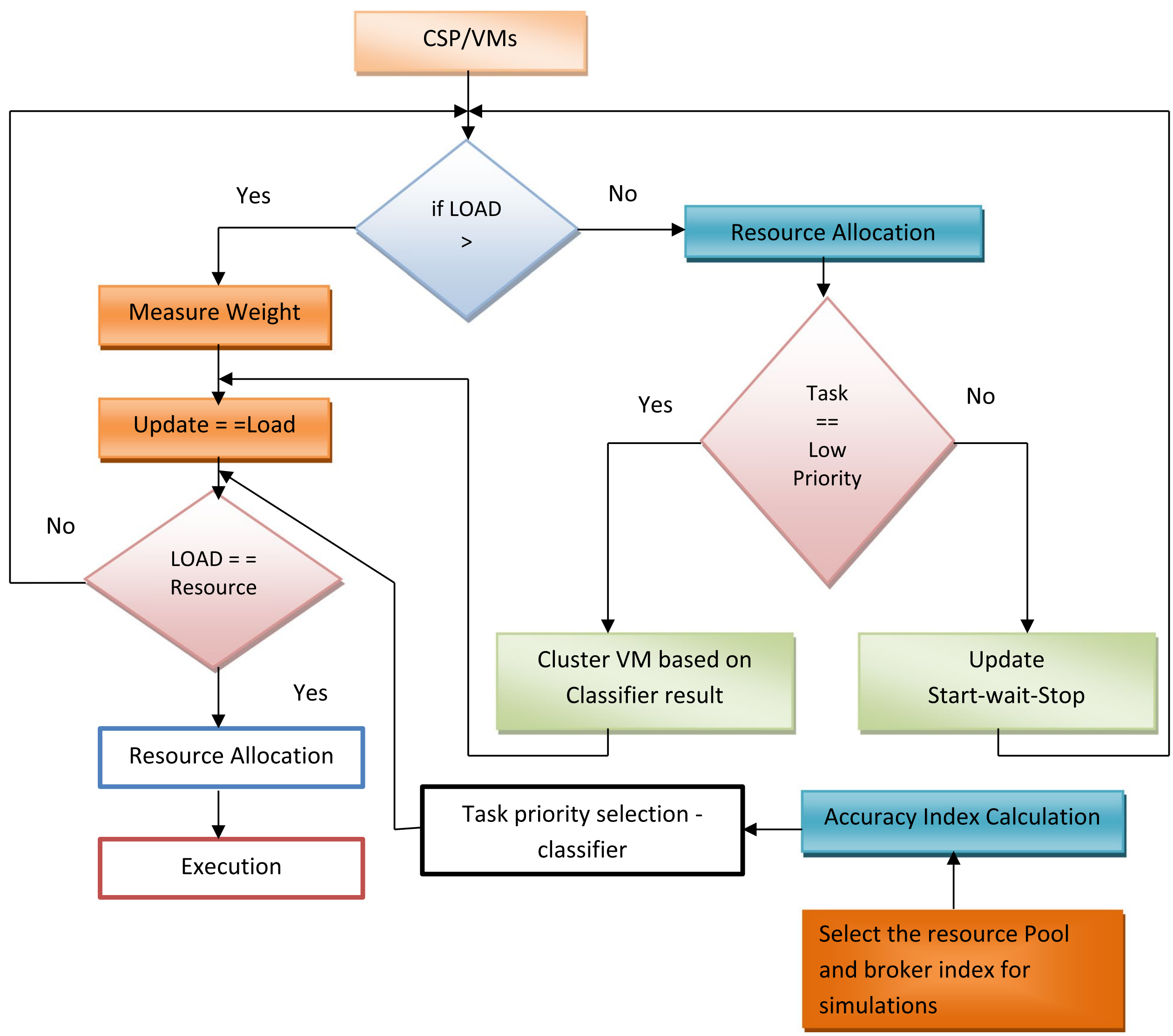


Fig. 4 Virtualized Intelligent Genetic Load Balancer

repository for selection. The data centre features are analyzed by using VM execution time and memory optimization shown in Figs. 5 and 6. In this case R is represented as Resource such as IO operations, capacity, wait, CPU, Load; UR is unrequested, Td is Total Data centre and D is represented as number devices are allotted VM.

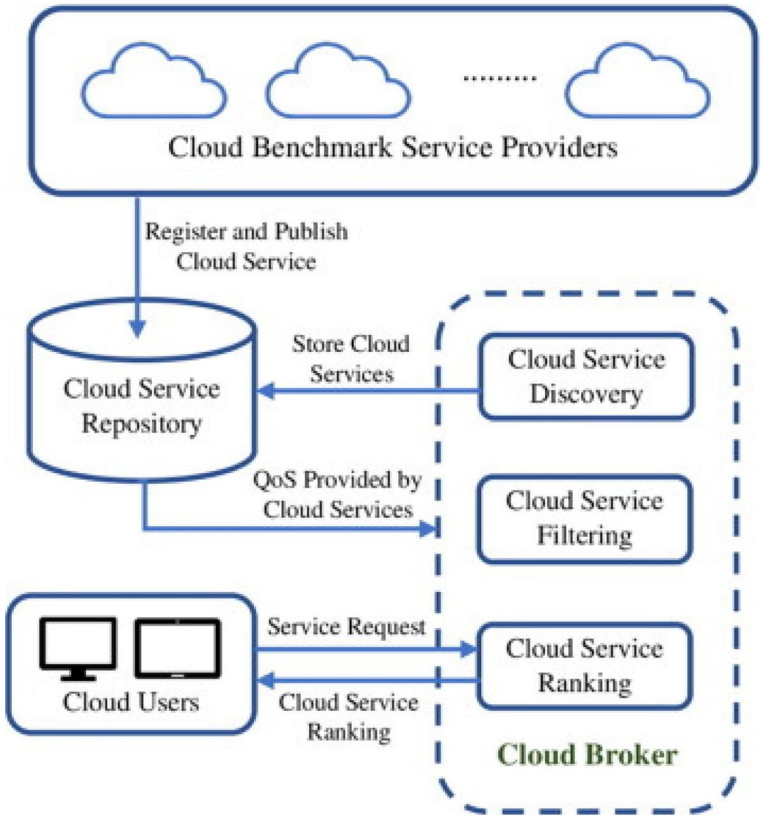


Fig. 5 Load Balancer – Web resource selection and allocation

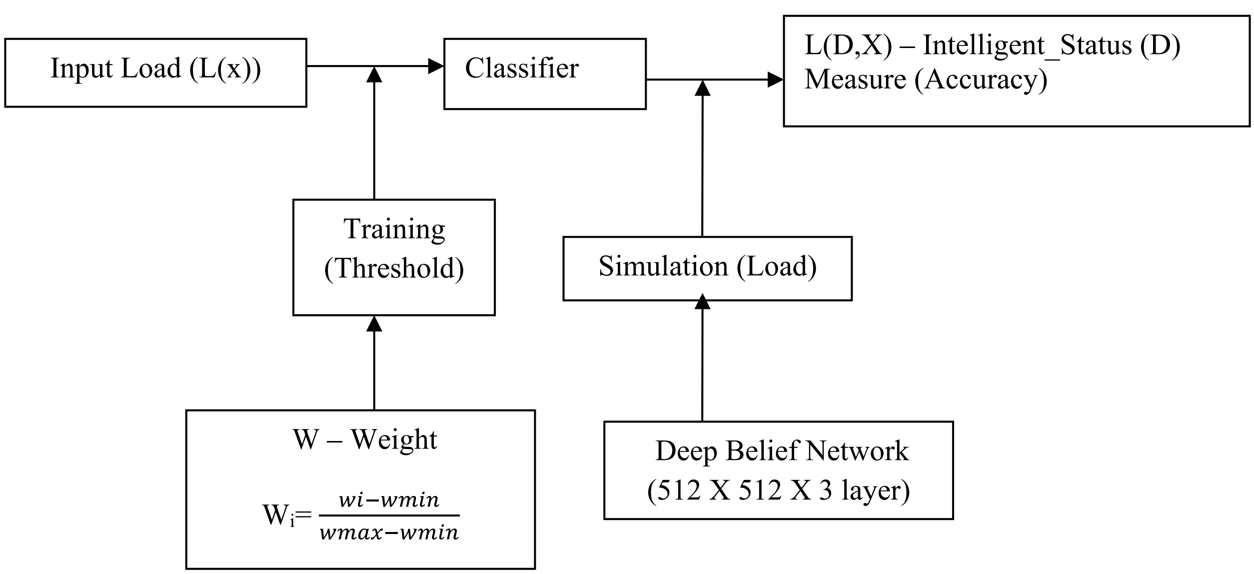


Fig. 6 Deep Belief Network Generation based on the Load

Cloud analytics

Cloud Analytics is applied for analyzing the risk and user level. Because Virtualized Intelligent Genetic Load Balancer system provides a security enabler based on service level agreement. So we identify the risk based on request and unrequest category. It is self and automated system based alert based system. Here resource evaluation can be done based below user selection factor

$$\mathrm{User\ (risk)_i} = \begin{cases} 0 \, if \, \mathrm{R_i} < \mathrm{UR_i} \\ \mathrm{U\ Req_i} - \mathrm{UNreq_i} \, if \, \mathrm{UR_i} < \mathrm{R_i} < \mathrm{FREE\_LOOK_i} \\ 1 \, if \, \mathrm{R_i} > \mathrm{FREE\_LOOK_i} \end{cases} \tag{1}$$

From above equation we idenfied the risk and remove the user while the above condition fails. If the result is in positive case Virtualized Intelligent Genetic Load Balancer algorithm will start work associated with resource provision scale and managing federated cloud.

```
Virtualized Intelligent Genetic Algorithm – Deep Belief Network Classifier

Step 1: Select Virtual machine and Allocate Data centre for task/load
Step 2: Verify the status of current load in VM and Store the value
Step 3: Classify the resource based on VM results and resource availability
Step 4: Organize the Virtual machine [lowest & highest Priority = = Weights]
Step 5:  Select the incoming request and verify the load values in VM
Step 6: Apply Round robin scheduling to select the quantum index for measuring waiting time
Step 7: Calculate the Turnaround time with respect to quantum size {quantum_size >= 1}
Step 8: Start the execustion from lowest to highest from step 4 and until step 7
Enhanced Load Balancer
Step 9: Calculate the Capacity and Load of each VM in execution time
Step 10: if (Load_VM = = Balance)
            Resource_pool();
           Execute();
        else
           Overload() = 1
          Roundrobin();
Step 11: Remove no load VM or less priority load or capacity until complete step 10

Virtualized Intelligent Genetic Algorithm

Step 12: Select the VM Cluster and Update Load, Weight and Capacity
Step 13: Generate Deep Belief Network for classification and sorting the VM
Step 14: Based DBF result: Sort(VM) <- Low to High Priority Selection
Step 15: Calculate: Allocation of VM and their Execution Time
Threshold-High -> VM(D) | min (∑ Threshold(High)) – Minimum_Exe_Time
Threshold-Medium -> VM(D) | min(∑ Threshold(High) + ∑ Threshold(Medium)) –
Minimum_Exe_Time
Threshold-Low -> VM(D) | min ((∑ Threshold(Low)) – Minimum_Exe_Time
Step 16: From above step the result is recorded and until step 15 repeat

Round_Robin(weight) <– Intelligent(Genetic_Status)
Step 17: Measure the VM{Execution Time, Turn Around Time and Completion Time of each
load or task}
Step 18: Specify the lower to higher order execution time values
Step 19: Allocate the task to each virtual machine until Threshold<=0
Step 20: Repeat the process until VM Completion
```

Algorithm 2. Virtualized Intelligent Genetic Algorithm – Deep Belief Network Classifier

By applying distribution factor to eliminate redundant data for classify the weights,

$$\mathrm{D(X,Y)} = \mathrm{F}(\mathrm{x,y}) - \left(\log_i W(i) - \log Wmax\right) \tag{2}$$

Phase 1: Initial process

First phase execution time is calculated for each virtual machine and stored.

Select the number of VM based on availability.

Three Cases: Host, Virtualizer, Guest

Host – Manage the whole VM, Guest – Selecting the Components and specify the communication, Virtualizer – Compute, Network, Storage and application resource handler.

In this case based on L (load) we find estimation index (EI) is represented as

$$\mathrm{EI\ (load)} = \left(\mathrm{EI_{ij}}\right)\mathrm{N} * \mathrm{X}.$$
$$\mathrm{EI\ (load)} = \left\{ \begin{matrix} \mathrm{EI_{11}} & \mathrm{EI_{12}} & \cdots & \mathrm{EI_{1n}} \\ \mathrm{EI_{21}} & \mathrm{EI_{22}} & \cdots & \mathrm{EI_{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{EI_{n1}} & \mathrm{EI_{n2}} & \cdots & \mathrm{EI_{nn}} \end{matrix} \right\} \tag{3}$$

Weight index and transaction factor is calculated as

$$\mathrm{Weight\ Index\ (load)} = \sum_{1}^{n} EI_{ij}(i = 1, 2, \ldots n) \tag{4}$$

$$\mathrm{Weight\ Index\ (x,y)} = W(x,y)_i^* = \frac{X_i Y_i'}{\sum_{\mathrm{i=1}}^{\mathrm{m}} X_i y_i'} \tag{5}$$

$$Transaction(\mathrm{Ti}) = \exp\left(\frac{Time\left(\mathrm{W(x,y)}_i, CSP_j\right) - \min(\mathrm{User}_i)}{\max\left(\mathrm{W(x,y)}_i\right) - \min\left(\mathrm{W(x,y)}_i\right)}\right) \tag{6}$$

From the equation the transaction is stored and allot the resource to cross over section result.

Phase 2: Virtualized Intelligent Genetic Cross Over Validation function

For the each cloud attenuation index is measured based on time with respect to load priority.

$$\mathrm{AI(load)} = \frac{Time\ (csp)_i^{\tau}}{\sum_{\mathrm{j=1}}^{\mathrm{n}} Time\ (csp)^{\tau}} \tag{7}$$

where.

$\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{AI(Load)} = 1$ and Time – execution time of VM.

Based on above representation the trust can be recorded as.

$$TRr = \sum_{1}^{\mathrm{m}} Tensor_{\mathrm{Time,Weight}} \times Time(i)_{\max} \times AI(load) \tag{8}$$

Quality factor is measured by

$$Quality = \left(TR_j^*\right)^T * Tensor(Time, Weight) = \begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1k} \\ Q_{21} & Q_{22} & \cdots & Q_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ Q_{j1} & Q_{j2} & \cdots & Q_{jk} \end{bmatrix} \quad (9)$$

$$Tensor\ (Time\ (i) = Quality * \mathrm{Estimate}(Q)^T \quad (10)$$

From the above result tensor time is calculated for measuring accuracy.

Phase 3: Measuring Execution Time

The virtual machines are selected based on finite number of iteration and set least probability factor. The intelligent genetic algorithm is used to select the VM from group D and transaction factor is measure for each completed process,

$$\phi(load) = Quality(Load) * \left[\frac{1}{\left(Transaction(i) + EI^{-n}\right)}\right] \quad (11)$$

In this case minimum transaction or less probability values VM are not considered and remove form D. The weight function is calculated based on each completed transaction execution time and turnaround time. The Time factor is calculated as,

$$Time\_Transaction_j\left(\mathrm{Weight_j}\right) = \frac{\mathrm{Quality}\left(\mathrm{W_j}\right)}{\sum_{j=1}^{m} \mathrm{Tensor}\left(\mathrm{W_j}\right)} \quad (12)$$

Table 2 Input values of each processing VM and their dataset

Input (Load and Weight)	Deep Belief Network – Classifier
Deep Belief Network	512 X 512 X 3 Layer – TensorFlow
Connected Layers	3 Layer with VM selection
VM Placement	Based on classifier accuracy X:iterations
Hidden Points	2,4,8,16,32 Connected nodes

Based on above test conditions, we calculated VMs turnaround time and execution time is calculated. Below Table 3 shows that the result of VMs execution factor.

From above time values we need to calculated accuracy factor for each VM using same load and weight factors. The below formula shows that calculation of accuracy, precision, recall and measure values. From this success index is calculated.

$$\mathrm{Quality}(\mathrm{Accuracy}) = \frac{Wmax + Whidden}{N} \quad (13)$$

$$\mathrm{Quality}(\mathrm{Precision}) = \frac{Wmax}{N - Quality(Accuracy)} \quad (14)$$

$$\mathrm{Quality}(\mathrm{Recall}) = 1 - \frac{Quality(Accuracy) - X}{N} \quad (15)$$

$$\mathrm{Quality}(\mathrm{Measure}) = \mathrm{Quality}\left(\mathrm{Accuracy}\right) + \sum Quality(Precision * Recall) - N \quad (16)$$

From above the result processing time of CPU and execution times are recorded. Same process is continued for all the VMs and CSPs. For the below Table 1 shows that CSP status and their execution time calculated based on above formulas,

TensorFlow is used to simulate our system with the configuration of 3.75 GHz GPU computing system, 1 TB HDD and 8 GB RAM. For running operating system we used windows 10 for execution. From the below Table 2 shows that input for processing the simulations,

Table 1 CSP Status and their execution time results

CSP	Date Centres	VM-ID	Start Time	End Time
CSPid1	5	1	2.12 ms	4.52 ms
CSPid2	10	2	3.21 ms	6.33 ms
CSPid3	15	3	3.57 ms	7.27 ms
CSPid4	20	4	4.15 ms	8.23 ms
CSPid5	50	5	5.37 ms	9.28 ms
CSPid6	100	6	7.21 ms	11.39 ms

From this sentiment index is calculated as

$$Sucess\_Index = \frac{Quality\left(Accuracy + Precision\right) - N}{Quality(RecallXMeasure)} \quad (17)$$

From this formulas are taken for calculating index, the below Table 2 represents trained and test data values with iterations.

From the above table we calculated accuracy factor and other deep learning results. Multiple iterations are done simulating cloud environment. Based on this over proposed Virtualized Enhanced Load Balancer with Intelligent Genetic algorithm gets average accuracy result of 95%. Also the below Table 4 shows that comparison of proposed with existing load balancer methods.

From above Table 5 and Figs. 7 and 8 shows that the comparison of existing methods with Virtualized Intelligent Genetic Load Balancer virtualized intelligent genetic algorithm. Compare with existing method Virtualized Intelligent Genetic Load Balancer system

Table 3 Result of VMs Execution and Turnaround Time from capacity, load and weight values

Virtual Machine	Capacity	Load	Weights	Execution Time	Turnaround time
5	10,20,50,100	100	50,100,200,500	0.92,0.91,0.93,0.92	0.94,0.94,0.95,0.94
10	10,20,50,100	100	50,100,200,500	0.91,0.92,0.93,0.93	0.91,0.92,0.93,0.93
15	10,20,50,100	100	50,100,200,500	0.92,0.91,0.92,0.93	0.94,0.93,0.94,0.94
20	10,20,50,100	100	50,100,200,500	0.94,0.93,0.94,0.95	0.94,0.93,0.94,0.95
25	10,20,50,100	100	50,100,200,500	0.93,0.95,0.95,0.94	0.93,0.94,0.94,0.95
50	10,20,50,100	100	50,100,200,500	0.92,0.94,0.94,0.94	0.94,0.94,0.95,0.94
100	10,20,50,100	100	50,100,200,500	0.94,0.94,0.95,0.94	0.91,0.92,0.93,0.93
150	10,20,50,100	100	50,100,200,500	0.93,0.94,0.94,0.95	0.93,0.94,0.94,0.95
200	10,20,50,100	100	50,100,200,500	0.92,0.92,0.95,0.94	0.94,0.93,0.94,0.95
500	10,20,50,100	100	50,100,200,500	0.94,0.93,0.94,0.94	0.94,0.94,0.95,0.94
1000	10,20,50,100	100	50,100,200,500	0.94,.0.95,0.94,0.94	0.94,0.93,0.94,0.94

Table 4 Result of VMs Accuracy, Precision, Recall, Measure and Success Index

Virtual Machine	Accuracy	Precision	Recall	Measure	Success Index
5	0.94	0.12	0.86	0.75	94
10	0.94	0.10	0.87	0.77	94
15	0.95	0.12	0.86	0.76	93
20	0.96	0.13	0.85	0.76	96
25	0.95	0.14	0.88	0.78	96
50	0.94	0.12	0.87	0.76	97
100	0.94	0.13	0.86	0.78	95
150	0.95	0.14	0.86	0.76	96
200	0.94	0.12	0.87	0.78	96
500	0.95	0.13	0.87	0.77	97
1000	0.94	0.12	0.87	0.78	96

accuracy index is high and better execution time. The comparison result shows that Virtualized Intelligent Genetic Load Balancer system has better accuracy results. In this case, we compared our virtual intelligent genetic method with existing methods such as CNN Classifier, Virtual Loader, Sentiment Analyser, Eigen Vector. From the above table VMs are increased and accuracy rate is compared. Dynamic Load balancer is observed in this case in three phases such as guest, virtualizer and host. Each phase performance compared and over proposed system has produced better success index factor. Multiple iterations, load and weight factors also compared in each stage. Load optimiser is used to measure the probability factor. Finally the success index is achieved as 95%.

Table 5 Comparison of various existing methods with Virtualized Intelligent Genetic Load Balancer method

Virtual Machine	Success Index (Accuracy Rate in %)				
	CNN Classifier	Virtual Loader	Sentiment Analyser	Eigen Vector	Virtualized Intelligent Genetic Method
5	72	81	67	77	94
10	72	80	67	73	94
15	73	79	68	77	93
20	74	78	69	73	96
25	75	77	70	77	96
50	71	81	68	76	97
100	76	78	69	75	95
150	77	79	67	74	96
200	76	81	68	77	96
500	76	79	71	75	97
1000	74	78	69	72	96

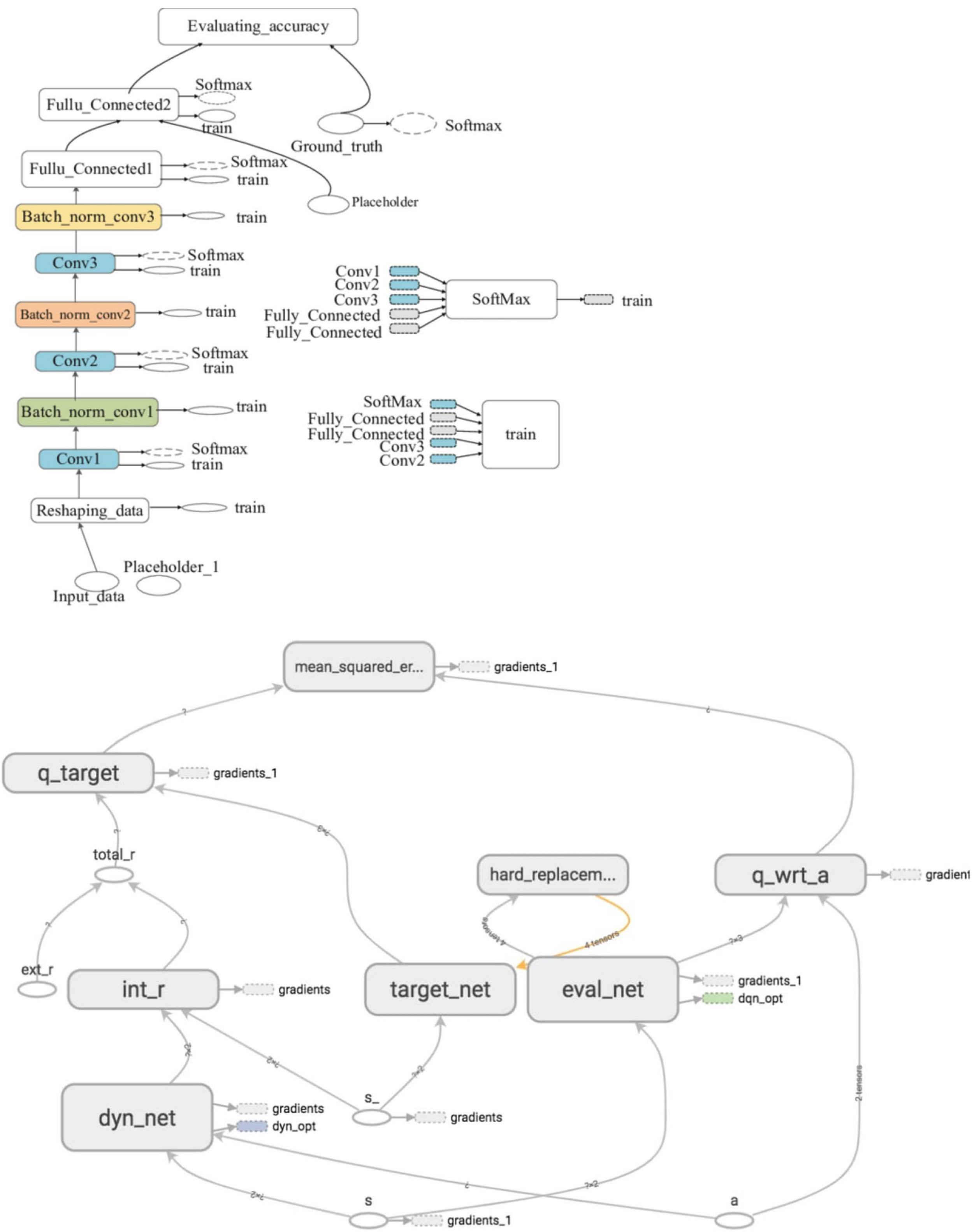
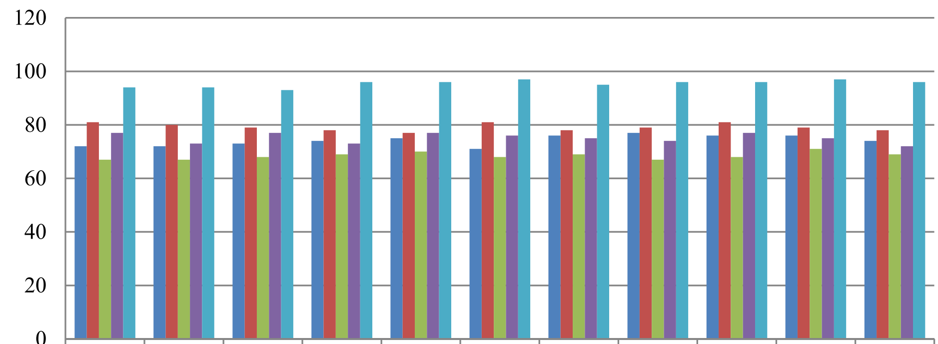


Fig. 7 TensorFlow result of Deep Belief Network Generation and Accuracy Index Calculation



	1	2	3	4	5	6	7	8	9	10	11
CNN Classifier	72	72	73	74	75	71	76	77	76	76	74
Virtual Loader	81	80	79	78	77	81	78	79	81	79	78
Sentiment Analyser	67	67	68	69	70	68	69	67	68	71	69
Eigen Vector	77	73	77	73	77	76	75	74	77	75	72
Virtualized Intelligent Genetic Method	94	94	93	96	96	97	95	96	96	97	96

Fig. 8 Comparison results of various methods with our proposed algorithm (Virtualized Intelligent Genetic Method) – Success Index (%)

Conclusion

Load balancing is important issue in cloud optimization problem. Cloud service providers and virtual machine are the major factor measuring success index. We proposed enhanced load balancer for balancing the loads and weights with respect to user requirements. Virtualized Intelligent Genetic method is applied for optimizing the cloud and measure the execution time at various iterations. Each virtual machine is classified by using Deep belief Network. TensorFlow is used to simulate the cloud environment and measuring the success index and accuracy factor. The various experiments are done to measure the accuracy and Virtualized Intelligent Genetic Load Balancer system achieves 95% as accuracy and compared with existing methods. In future this method can be implemented for automated and dynamic dataset.

Acknowledgements
Not applicable.

Authors' contributions
Only my contribution in my full paper.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 8 March 2023 Accepted: 30 August 2023
Published online: 02 October 2023

References

1. Manikandan S, Dhanalakshmi P, Priya S, Odilya Teen AM (2021) "Intelligent and Deep Learning Collaborative method for E-Learning Educational Platform using TensorFlow." Turkish J Computer Mathematics Education 12(10):2669–76 (E-ISSN: 1309–4653, 2669–2676)
2. Manikandan S, Chinnadurai M (2022) Virtualized Load Balancer for Hybrid Cloud Using Genetic Algorithm. Intelligent Automation Soft Computing 32(3):1459–1466
3. Manikandan S, Chinnadurai M (2019) 2019, 'Intelligent and Deep Learning Approach OT Measure E-Learning Content in Online Distance Education.' Online J Distance Educ e-Learning 7(3):2147–6454
4. Anton Beloglazov and CanturkIsci, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing" IBM T. J. Watson Research Center Hawthorne, NY 10532, 2018
5. Luiz SO, Perkusich A, Lima AMN (2010) Multisize Sliding Window in Workload Estimation for Dynamic Power Management. IEEE Trans Computers 59(12):1625–1639
6. Chunfeng Lv, Jianping Zhu & Zhengsu Tao, "An Improved Localization Scheme Based on PMCL Method for Large-Scale Mobile Wireless Aquaculture Sensor Networks", Arabian Journal for Science and Engineering volume 43, pages1033–1052(2018), https://doi.org/10.1007/s13369-017-2871-x
7. Huang W, Ho T, Langberg M, Kliewer J (2018) Single-unicast secure network coding and network error correction are as hard as multiple-unicast network coding. IEEE Trans Inf Theory 64(6):4496–4512. https://doi.org/10.1109/TIT.2018.2820686
8. Rouvier M, Favre B (2016) SENSEI-LIF at SemEval-2016 Task 4 : Polarity embedding fusion for robust sentiment analysis. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). San Diego, Association for Computational Linguistics
9. Cliche M, BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs, arXivPrepr. arXiv1704.06125
10. Lei T, Joshi H, Barzilay R, Jaakkola T, Tymoshenko K, Moschitti A, Marquez L. Semi-supervised Question Retrieval with Gated Convolutions. arXivPrepr. arXiv1512.05726, 2015

11. Yin Y, Yangqiu S, eta Zhang M (2017) NNEMBs at SemEval-2017 Task 4: Neural Twitter Sentiment Classification: a Simple Ensemble Method with Different Embeddings. Proc 11th Int Work Semant Eval 621–625
12. Rodrigo N, Calheiros Rajiv Ranjan, Beloglazov Anton, De Rose César A. F, Buyya Rajkumar (2015) CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. Softw Pract Exp 41(1):23–50
13. Shinde V, Dange A, Lambay MA (2016) Load Balancing Algorithms in Cloud Computing. Int J Comput Sci Trends Technol (IJCST)
14. Sekara K, Kosal KR (2017) SIQ Algorithm for Efficient Load Balancing In Cloud. IEEE 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)
15. Dobber M, van der Mei R, Koole G (2016) Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment: A Comparison. IEEE Trans Parallel Distrib Syst 20(2):207
16. Chaczko Z, Mahadevan V, Aslanzadeh S, Mcdermid C (2011) Availability and Load Balancing in Cloud Computing. 2011 International Conference on Computer and Software Modeling IPCSIT 14. IACSIT Press, Singapore
17. Christian S et al (2016) Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition
18. Vazquez C (2019) On the Performance Variability of Production Cloud Services. Delft University of Technology, Parallel and Distributed Systems Group
19. Sulistio A (2020) Performance and Power Management for Cloud Infrastructures. Department of Mathematics and Computer Science, Distributed Systems Group
20. Priyanka CP, Subbiah S (2017) Comparative Analysis on Virtual Machine Assignment Algorithm. 2017 IEEE International Conference on Computing and Communication Technologies

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.