I.J. Information Technology and Computer Science, **2020, 3, 8-18**
Published Online June 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2020.03.02

Load Balancing Optimization Based on Deep Learning Approach in Cloud Environment

Dr. Amanpreet Kaur
Associate Professor, Chandigarh Engineering College, Landran (Mohali)
E-mail: er.amanpreet14@gmail.com

Dr. Bikrampal Kaur
Professor, Chandigarh Engineering College, Landran (Mohali)
E-mail: mca.bikrampal@gmail.com

Dr. Parminder Singh
Associate Professor, Chandigarh Engineering College, Landran (Mohali)
E-mail: singh.parminder06@gmail.com

Mandeep Singh Devgan
Assistant Professor, Chandigarh Engineering College, Landran (Mohali)
E-mail: mandeep.it@cgc.edu.in

Harpreet Kaur Toor
Assistant Professor, Chandigarh Engineering College, Landran (Mohali)
E-mail: harpreettoor.appsci@cgc.edu.in

Received: 01 November 2019; Accepted: 23 November 2019; Published: 08 June 2020

Abstract—Load balancing is a significant aspect of cloud computing which is essential for identical load sharing among resources like servers, network interfaces, hard drives (storage) and virtual machines (VMs) hosted on physical servers. In cloud computing, Deep Learning (DL) techniques can be used to achieve QoS such as improve resource utilization and throughput; while reduce latency, response time and cost, balancing load across machines, thus, increasing the system reliability. DL results in effective and accurate decision making of intelligent resource allocation to the incoming requests, thereby, choosing the most suitable resource to complete them. However, in previous researches on load balancing, there is limited application of DL approaches. In this paper, the significance of DL approaches have been analysed in the area of cloud computing. A Framework for Workflow execution in cloud environment has been proposed and implemented, namely, Deep Learning-based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (DLD-PLB). Optimal schedule for VMs has been generated using Deep Learning based technique. The Genome workflow tasks have been taken as input to the suggested framework. The results for makespan and cost has been computed for the proposed framework and has been compared with our earlier proposed framework for load balancing optimization - Hybrid approach based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (HDD-PLB)" framework for Workflow execution. The earlier proposed approaches for load balancing were based on hybrid Predict-Earliest-Finish Time (PEFT) with ACO for underutilized VM optimization and hybrid PEFT-Bat approach for optimize the utilization of overflow VMs.

Index Terms—Deep Learning, Load balancing, Workflows, Convolution Neural Networks (CNN), Resource provisioning, Framework.

I. Introduction

The heterogeneous cloud resources are distributed across different geographical locations across the globe in various datacenters, so it requires load distribution among available resources to achieve high performance and optimal resource utilization. The dynamic workload of the cloud is distributed evenly among the nodes available for processing. For this, load balancing is done. The load can refer to CPU utilization, memory or storage usage or it can network load. The objective is to distribute the load among machines evenly to improve the overall performance of task execution over cloud resources (processing power, memory, network and storage). Load Balancing techniques are responsible for allocating resources with minimum resource wastage when machines are utilized below their capacity and avoiding overloading/ underloading of Virtual Machines (VMs) (Mittal & Dubey, 2017; Ghomi et al., 2017). When a large number of requests are targeted to a single

virtual or physical machine, then, it gets overloaded with workload above its capacity. This results in increased response time of the applications. This problem can be solved by migrating tasks from one VM (highly loaded) to another which is lightly loaded (Matsumoto & Ezaki, 2011). Load balancing, in cloud environment, is done by migrating load from heavily loaded virtual machines (VMs) to comparatively lightly loaded VMs. Load balancing techniques are used to make sure that each machine in the cloud datacenter performs approximately the equal number of tasks at any point of time (García-Gonzalo & Fernández-Martínez, 2012). Load balancing is important for both cloud service provider as well as cloud service user such that the prior aims to achieve high resource utilization and throughput, while latter wants reduced cost, waiting time, application completion time and VM makespan.

Deep Learning (DL) is a technique which recognizes and remembers objects through training examples. This technique is motivated through the complex structures of the brain neurons and their interconnections. The future predictions about results are made depending on the training data. DL is used when the availability of data is at large extent to train the underlying event or phenomenon. In traditional machine learning, the input is trailed by feature extraction which comprises complex mathematical functions and then classification of the extracted features is done. Since in machine learning, there are two separate phases (Feature extraction and classification phases) which causes inefficiency and less effectiveness towards achieving the optimal results. However, in case of deep learning the two phases are merged in a single stage in which deep learning algorithms perform both feature extraction and classification in different hidden layers.

In cloud computing, Deep Learning (DL) Principles can be used to achieve Quality of Service (QoS) such as to improve resource utilization and throughput; while reducing makespan, latency, response time and cost of balancing load across machines. DL results in effective and accurate decision making of intelligent resource allocation to the incoming requests, thereby, choosing the most suitable resource to complete them. Deep learning techniques are employed to improve the decision making and achieving accurate and effective results (Yang et al., 2018). These techniques result effective decision making for allocating tasks to resources (VMs) and achieving load balancing across available VMs. Deep learning approaches provide effective resource utilization in cloud environment by facilitating resource provisioning (for intelligent resource allocation) while choosing the best-suitable and available resource for the tasks.

Deep Learning techniques are used in resolving the problems that can be elucidated by heuristics. The parameters of heuristic techniques are initialized and then tuned to improve the performance of algorithm. The tuning of parameters is a continuous process until the results converge to optimal values or specific number of iterations is completed. In dynamic environment such as cloud computing, the parameters tuning to optimal values is time –boundless process in heuristic and metaheuristic techniques for scheduling and load balancing, hence, Machine Learning (ML) or Deep Learning based techniques are used for get the optimal results efficiently and effectively. These techniques are capable of providing more precise model and reasoning inferences for scheduling system. Also, ML techniques improve the accuracy of decision making in heuristics algorithms (Abadi et al., 2016). Deadline-based scheduling imposes rules on the task so that only those tasks could be executed by VMs which are completed before their deadlines exhaust.

II. Literature Review

Earlier research on load balancing approaches was mainly based on metaheuristic techniques. Cloud resource provisioning architecture incorporating load balancing is given by Ferretti et al. (2010) which effectively fulfil QoS requirements for application requests by users and optimizes the resource utilization. Mohamed & Al-Jaroodi (2011) resolved the load balancing issue in cloud computing and proposed delay-tolerant dynamic load balancing (DTDLB) technique based on Dual Direction Operations (DDOs) for downloading and executing the parallel applications on independent, heterogeneous and distributed servers. Their methodology does not require constant monitoring or load reallocation at run time. Li et al. (2011) proposed Load Balancing Ant Colony Optimization (LBACO) algorithm for balancing the load across entire system and minimize makespan. The results are found to be superior to basic ACO and FCFS algorithms. The tasks have been considered to be mutually independent with no precedence constraint, non preemptive and computation intensive. The increased complexity of applications based on workflow inputs in heterogeneous environment like cloud can be tackled using hybrid techniques to solve the workflow-based scheduling problem. Another scheduling strategy with load balancing of VMs on hosts based on Genetic Algorithm is presented by Gu et al. (2012). Hung et al. (2012) have suggested an algorithm for cloud environment load balancing based on combination of Max-Min and Max algorithms, viz., LB3M. This algorithm assigns the tasks to nodes dependent on their computing capacity. The researchers have successfully achieved better resource utilization. Hsiao et al. (2013) solved the problem of load imbalance by presenting a novel fully distributed algorithm for load rebalancing in comparison to centralized approach used in Hadoop distributed file system (HDFS) which is prone to single point failure. Fister et al. (2013) improved the original BA by hybridizing it with differential evolutionary (DE) techniques and proposed Hybrid Bat Algorithm (HBA) which produced optimal results for multidimensional optimization problems. The optimization outcomes are subjective to the dimensions of the problem. Jaikar et al. (2014) offered architecture for load balancing in a cloud environment to efficiently assign VMs to improve resource utilization and decrease energy consumption.

Table 1. Summary of literature review

Year	Author	Problem Area	Proposed Technique	Outcome	Deep Learning Approach used
2015	Awad et al.	Task Scheduling and load balancing in cloud computing environment	Load Balancing Mutation (balancing) a particle swarm optimization (LBMPSO) based schedule	Reduced make span, execution time, round trip time, transmission cost than standard PSO, random and LCFP) algorithm	No deep learning approach used
2016	Keshvadi & Faghih	load balancing in IaaS cloud	Multiple Agent-based Load Balancing Algorithm (MA) for dynamic load balancing across VMs	Minimized waiting time of the tasks and makespan, also guaranteed SLA	No deep learning approach used
2017	Duan & Yang	Load balancing across VMs in VPC	Propose a fat-tree data center virtualization framework based on control of all the switching elements in network virtualization	Framework achieves global load balancing on the underlying physical network and delivers predictable performance	No deep learning approach used
2017	Guo	Task Scheduling in cloud computing	cloud computing task scheduling algorithm based on ant colony algorithm and balance the load of the system	Minimize makespan and the total cost of tasks execution,	No deep learning approach used
2018	Hou & Zhao	Resource Scheduling and load balancing in cloud computing	Resource-aware Load Balancing Clonal Algorithm (RLBCA) based on load balancing strategy and clonal selection algorithm	Diverse population as input and faster convergence rate	Deep learning based on Colonal selection algorithm (CSA)
2018	Gomez et al.	load balancing in Heterogeneous Networks	Machine learning (ML) techniques to discover hidden patterns (PCA), learn from the labeled data and make decisions	Enhance the capabilities of an urban IoT network operating under the LoRaWAN standard	ML scheme based on Markov Decision Process (MDP) with both unsupervised supervised techniques
2016	Roman et al.	load distribution and resource utilization in heterogeneous Grids	load balancing achieved through a threshold based hybrid technique for workload distribution in Balanced Partner Based Dynamic Critical Path for Grids (B-PDCPG) algorithm	Effective workload balance and better resource utilization of with decreased makespan	No deep learning approach used

Awad et al. (2015) proposed scheduling and load balancing using Load Balancing Mutation Particle Swarm Optimization (LBMPSO) approach. VMs are assigned optimal load which is being distributed by PSO algorithm. Pacini et al. (2015) proposed and examined ACO-based job scheduler to allocate tasks to suitable VMs hosted on physical machines. Keshvadi & Faghih (2016) proposed a Multiple Agent (MA) -based Load Balancing Algorithm to achieve dynamic load balancing using multiple monitor and mobile agents, and avoiding VM migrations, thus, reducing migration cost and maximizing resource utilization. A threshold-based load balancing method has been projected by Roman et al. (2016) to improve the global throughput of heterogeneous Grid environment, and henceforth, improved utilization of resources. Guo (2017) proposed multi-objective ACO (MO-ACO) based task scheduling with load balancing algorithm with makespan and cost optimization. Better results were attained as compared to Min-min algorithm in terms of balanced load, cost and makespan. Chalack (2017) developed and compared a new task scheduling algorithm based on PSO algorithm. The proposed algorithm has been initialized by randomly selected VMs for execution of tasks. Duan & Yang (2017) have addressed load balancing as one of the important issues of resource management in cloud computing which affects cost, availability and flexibility. The researchers have proposed openflow protocol-based virtualization framework for load balancing supporting heterogeneity of communication network of VMs in virtualized private clouds (VPCs). Deep Learning algorithms are used to solve complex problems related to forecasting, classification, high computational infrastructure based problems over wide area and clustering problems (Singh et al., 2017). The wide area developments in Deep Neural Networks have created a scope for deploying them in distributed environments like cloud computing. For example, Neural Networks (NN) have been adopted into data center management and reduced the overall cooling bill of Google data centers by 40%. Milani, A. S., & Navimipour, N. J. (2016) presents literature review of load balancing techniques with their detailed classification on basis of different parameters. The authors discussed that the mechanisms for solving problem of load balancing further need improvements in terms of response time and performance.

Ghomi et al. (2017) presented a detailed literature on task scheduling and load balancing techniques. They discussed these algorithms as important aspects of cloud computing and presented important metrics for load balancing. They gave novel categorization of these algorithms, such as, Hadoop MapReduce load balancing category, Natural Phenomena-based load balancing category, workflow specific category etc. Hou & Zhao (2018) proposed resource scheduling and load balancing

fusion algorithm with deep learning to solve energy consumption problem. Diverse population and faster convergence rate are major concerns of the proposed algorithm for reducing energy utilized in data centers to promote green cloud computing. The authors proposed energy consumption resource scheduling optimization algorithm under cost constraint called Resource-aware Load Balancing Clonal Algorithm (RLBCA) to reduce energy utilized in datacenters. Gimez et al. (2018) projected machine learning techniques based load balancing scheme. The proposed scheme used both unsupervised and supervised methods along with Markov Decision Process (MDP). Their scheme has been applied to improve the urban IoT network. The results of simulation have shown that network packet delivery ratio (PDR) has improved and data delivery -energy cost has decreased. Table 1 shows the summary of the work done (not limited to this) in load balancing for cloud computing environment in recent years. It has been observed that only few works in load balancing has been based on deep learning techniques which give the motivation for present work.

III. Present Work

Load balancing is an important facet of cloud computing which is necessary for uniform load distribution among resources like servers, network interfaces, hard drives (storage) and virtual machines (VMs) hosted on physical servers (Mittal & Dubey, 2017). When tasks are scheduled on VMs hosting on physical nodes, there might arise a situation that some of the VMs are overutilized, while others remain underutilized. When VMs are overutilized, then their makespan (total time taken by a VM to complete all the tasks allocated to it) also increases. However, when the VMs are underutilized, although the makespan decreases, yet it results in increased cost of resource utilization as the available resources (VMs) are not extensively utilized resulting in their wastage. Thus, there is a need to balance the load across the VMs so that both the makespan and cost parameters must be controlled and balanced. The decrease in makespan must not result in increase of cost of resource utilization and vice versa.

In this research, Deep Learning- based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (DLD-PLB) Framework for Workflows has been proposed and implemented. Optimal schedule for VMs has been generated using Deep Learning based technique. The Genome workflow tasks have been taken as input to the proposed framework. The results for makespan and cost has been computed for the proposed framework and has been compared with our earlier proposed framework for load balancing optimization - Hybrid approach based Deadline-constrained, Dynamic VM Provisioning and Load Balancing (HDD-PLB)" framework for Workflow execution (Kaur A et al.,2018a; Kaur A. et al.,2018b). The earlier proposed approaches for load balancing were based on hybrid Predict-Earliest-Finish Time (PEFT) with ACO for underutilized VM optimization and hybrid PEFT-Bat approach for optimize the utilization of overflow VMs.

IV. Methodology for Proposed DLD-PLB Framework

In present work, Keras library for deep learning has been used with TensorFlow for workflow dataset. TensorFlow is a machine learning system which helps to experiment with novel algorithms for training and optimization. TensorFlow has been used to operate for large datasets like workflows and cloud computing like heterogeneous environments, so it has been used in scheduling and load balancing model of proposed DLD-PLB framework. The methodology for proposed framework has been shown in figure 1. The workflow is parsed to collect the tasks. As deep learning approaches are implemented on huge datasets, so Genome workflow has been taken as the input workflow as it has large number of tasks as compare to other workflows. Each task takes computation time and cost to complete its execution. The deep learning regression maps the input function (f) to a continuous output variable (y). Regression predictive model has been used to predict the continuous schedule (unknown variable) of tasks based on their known computation time and cost (known variables). On the other hand, classification model is used to predict discrete output variables from input variables using a mapping function. In present work, classification predictive modeling has not been used as the input tasks are not to be labelled but their schedule is to be generated based on their computation time and total cost. So, deep learning regression model has been used in this work.

Convolution Neural Networks (CNN) training is similar to regular neural network but convolution operations increase its complexity. Convolution is achieved on the input by a filter which generates a feature map. The proposed scheduling model of DLD-PLB framework assumes that the task which takes less time to complete its execution is scheduled first. The CNN of proposed deep learning approach for DLD-PLB framework has three hidden layers:

1. Convolution Layer
2. Pooling layer
3. Relu Function

Convolution layer is used to mix the tasks to eradicate dependencies so that the input to the load balancing model of the proposed framework involves huge dataset of independent tasks. During convolution, task features has been extracted. Multiple convolution operations are done on the input and each operation uses a different filter, resulting in different feature maps. These maps are set together in the convolution layer as a final output. This final output is passed through an activation function (ReLU) to convert it into non-linear form. Each hidden layer is activated by ReLU function which is input to the second layer. The Rectified Linear Unit (ReLU): f(x) =

max(0, x). the output of ReLU function is 'x' if 'x' is positive and otherwise, it is 0.

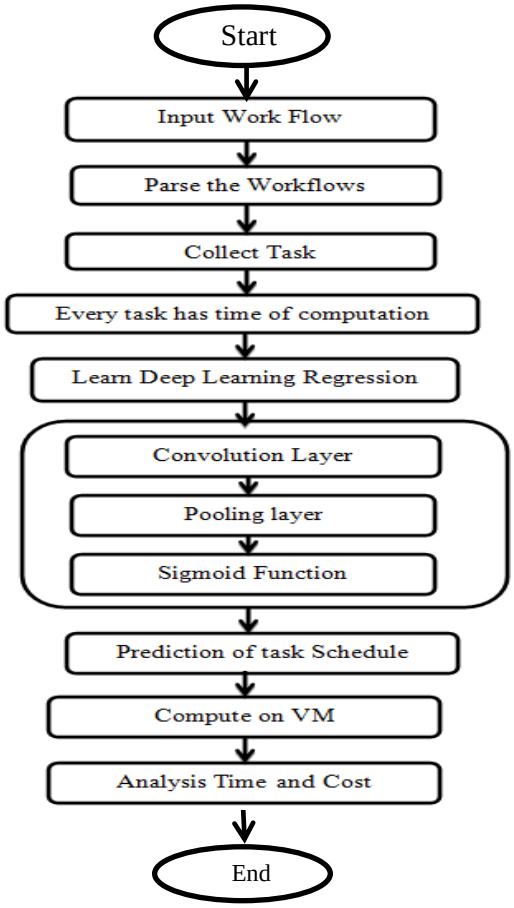

Fig.1. Methodology for Proposed Work

In present work, two prominent features have been considered, task computation time and cost. These extracted features have been allocated weights for further training and learning. After convolution layer is the pooling layer which lessens the computation complexity of learning process and speed up the training phase of input tasks. The predicted task schedule is allocated to VMs and then total time and cost incurred taken to complete their execution has been computed. The results are compared with the previously proposed hybrid heuristic- metaheuristic based framework for load balancing.

V. Design of Proposed DLD-PLB Framework

The proposed framework for load balancing optimization based on deep learning approach is shown in fig 1. The proposed DLD-PLB framework has four models- IAAS model, workflow scheduling model, load balancing model and performance evaluation model.

A. IAAS Model

This model of proposed framework includes the hardware (processor, memory and disk) of the single physical host upon which virtual machines have been incorporated. Dynamic number of VM instances has been created on the single host machine. The Infrastructure as a Service (IAAS) cloud model includes hardware layer and virtualization layer. It contains infrastructure components like processor cores, memory and storage which comprise a cloud datacenter. The single host hardware is virtualized to form virtual machines (VMs) using hypervisor. For the proposed framework, the cloud environment has been simulated by Cloud workflow simulator (CWS) for experiments. The number of VMs taken for workflow execution is increased exponentially. The VMs are created on a single physical machine (host). The resources (CPU cycles and memory) of the host are shared among the VMs generated on its hardware. As the number of VMs increase, a high degree of resource sharing is carried out by the hypervisor which enhances the utilization of physical machine (host) resources.

B. Workflow Scheduling Model

A workflow with large number of tasks has been taken as input to the workflow scheduling model. The input workflow has tasks with dependencies and has been considered as the training data. The convolution layer extracts the feature of tasks- computation time and cost. The convolved features (extracted features) have been generated by computing the dot product of input with the filter. In other words, the feature map generated as output of convolution has undergone filter operation. This operation has been applied on the tasks such that the tasks which do not fulfil the deadlines of total completion time and cost have been filtered. This has been done by applying multiple convolutions and filtering operations. Thus, a huge collection of tasks with an initial schedule has been created on the basis of time and cost. These parameters have been taken as the extracted features of the input task. The input tasks have been assigned weights on the basis of features extracted. Genome is a scientific workflow application used for experiment purposes. The workflow task are signified as a direct acyclic graph (DAG) G(T,E) such that 'T' is the set of n tasks $\{t_1,t_2,\ldots.t_n\}$ to be executed on a VM and 'E' is the set of edges representing the dependencies among tasks of workflow. Further, an entry task t_{entry} in DAG is without any parent and an exit task t_{exit} is without any child nodes. The execution time from/to these tasks is zero.

As shown in figure 2, the edges from t_{entry} to t_1 and t_2 have weight 0. Similarly edges from t_6 and t_7 to t_{exit} have weight zero.

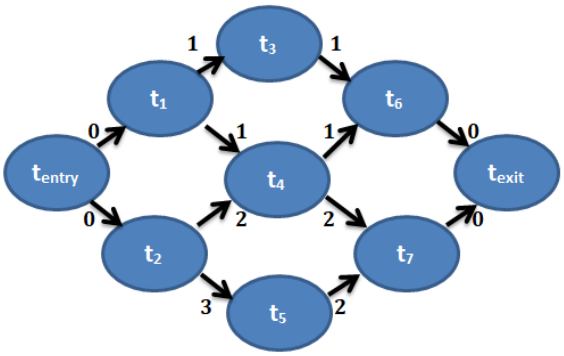

Fig.2. Sample Workflow Model

C. Load balancing Model of Proposed DLD-PLB Framework

In this model of proposed Deep Learning based load balancing optimization framework, firstly, the schedules generated by the workflow scheduling model has been taken as input i.e. the training data. Pool layer is used to reduce the spatial dimensions ad computational complexity of the training data. The most effective schedule (on the basis of extracted features) has been selected for the pooling phase. The pooling has been done on the basis of the extracted features-time and cost. Max pooling is used in which the schedule with maximum task length but minimum time and cost parameters has been selected for pooling.

Relu layer is use to map the negative values to zeros by applying and activation function. The VMs are assigned schedule generated by the proposed scheduling and load balancing model optimized by Gradient descent optimizer. The objective is to find the best weights of features which result in minimum loss of tasks due to deadline violations.

D. Gradient Descent Optimization

The goal of all supervised machine learning algorithms is to best estimate a target function (f) that maps input data (X) onto output variables (Y). These functions are basically the linear regression functions. In proposed deep learning approach for load balancing optimization of DLD-PLB framework, coefficients, like, features extracted (in terms of time and cost of input tasks), how many features extracted (i.e., their count), have been used to characterize the optimization algorithm find the best estimate for the target function (f). Thus, gradient descent optimization has been used to obtain the best schedule of tasks based on optimal values of time and cost parameters (coefficients).In proposed framework, the parameters are calculated using the proposed deep learning algorithm based on gradient descent optimization. The algorithm executed over the entire dataset of workflow tasks and optimal values of parameters are computed during iterations.

E. Performance Evaluation Model

This model evaluates the performance of the proposed framework on the basis of gradient descent which is an optimization approach to find the minimum of the function. In other words, gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes time and cost functions. The evaluations are done on the basis of the results computed for time and cost parameters of input tasks.

VI. Proposed Deep Learning Algorithm for Predicting Schedule

In the proposed algorithm, the cloud tasks refer to the tasks of Genome workflow. This workflow has huge dataset in terms of number of tasks so considered suitable for applying deep learning technique in proposed framework. The labels are the parameters taken for evaluation i.e., time and cost. The number of features decides number of weights, so here, 2 weights have been taken. The labels are assigned weights W_1 (time) and W_2 (cost). "Epoch" refers to one pass during full training set. During convolution, each hidden performs feature extraction by following a non-linear activation function. Y[k] is the output of a single hidden layers each with 'k' nodes $[h_1,h_2,\ldots h_k]$ during an epoch is computed as:

$$y[k]=v*h[k] \quad (1)$$

where 'v' is the vector of weights of edges connecting hidden node to output node and $v = [v_1, v_2, \ldots, v_k]$. Two hidden layers are taken for the propose deep learning algorithm. These layers are initialized with weights for time

(W_1) and cost (W_2) metrics. Gradient descent optimization has been used to initialize the member of epoch. Further, each hidden node's value for 'n' inputs and is computed as:

$$h[k]= f\left(\sum_{i=1}^{n} w[k] * x[i]\right) \quad (2)$$

CNN with 'l' hidden layers, $h^l = f(w^l * h^{l-1})$, $l \geq 2$

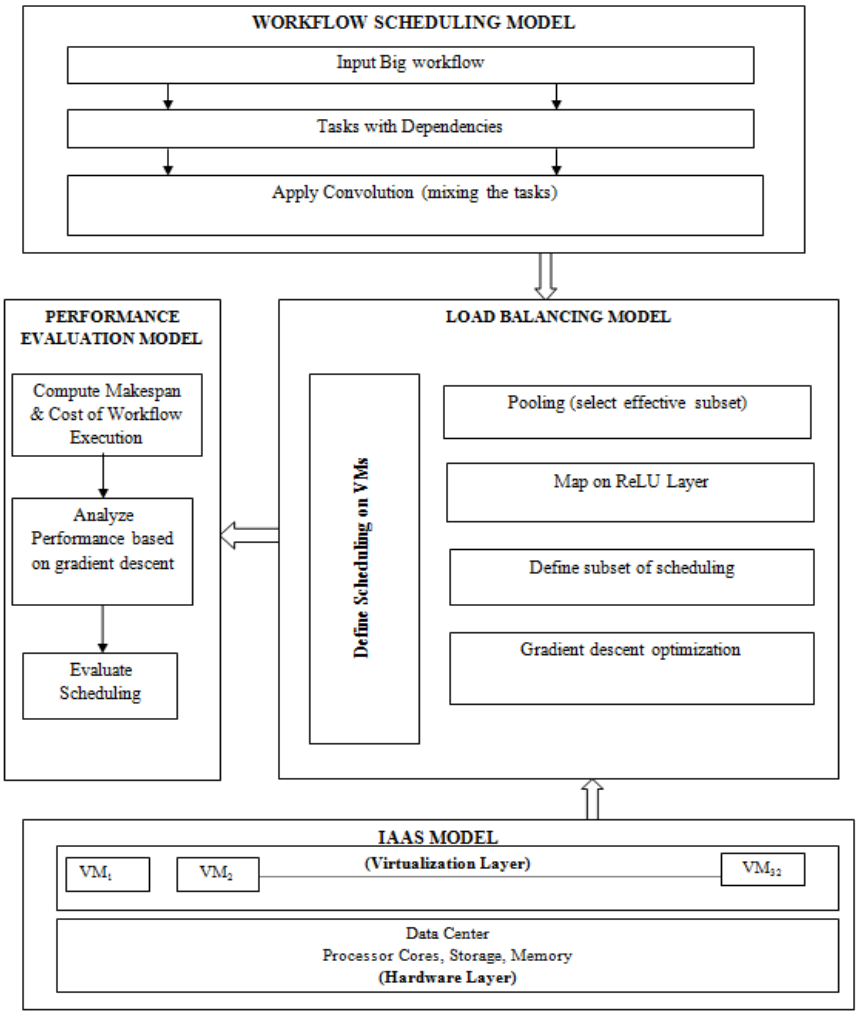

Fig.3. Proposed Framework based on Deep Learning: DLD-PLB Framework

A. Rectified Linear Unit (ReLU) Activation Function for proposed Deep Learning Algorithm

In proposed algorithm, the activation function is Rectified Linear Unit (ReLU) is currently the most popular as it is cheap to compute due to simpler mathematical operations. So, this function is considered best activation function in deep Neural Networks. The ReLU mapping is done to convert the non-linear ReLU function to linear function. This has been done in present work as follows:

$$f'(x) = \frac{d\left(\ln\left(1+x^{x[n]*h[n]}\right)\right)}{dx} \quad (3)$$

Then the second layer calculates weighted sum on that input and it in turn, fires based on another ReLu activation. However, the other activation functions like sigmoid and tanh functions are costly due to dense activations resulting in large computations to be processed so avoided in the proposed algorithm. Ideally, a few neurons in the network are required to be activated, thus making the activations sparse and efficient.

Algorithm: Prediction of schedule with effective resource utilization for deadline constrained workflow tasks

Input: Cloud tasks with deadline constraint
Output: predicting schedule with effective resource utilization
Step 1: Initialize weight of labels W_1 and W_2 and member of epoch with gradient descent optimization.
Step 2: For (K ≤ epoch)
Do
$y[k] = X[k] * h[n]$ (feature extraction)
$y[n] = \sum_{k=1}^{n} x[K] * h[K]$ (feature extraction for all the input tasks)
$x[n] \leftarrow Numberoffeatures$
$f'(x) = \frac{d\left(ln\left(1+x^{x[n]*h[n]}\right)\right)}{dx}$
$f'(x) \leftarrow ReluLayerMapping$
end
Step 3: Update weight by optimization

$$\Delta W_{x[n]}^{K} = d\left(I_K^{x[n]}\right) y_x^n \quad (4)$$

Δ W ← Update Weights
Step 4: Make Model

B. Gradient Descent Optimization in Proposed Algorithm

The proposed deep learning algorithm is based on the Gradient descent optimization algorithm which is used to compute the parameter (coefficient) values while minimizing the cost function. This algorithm is best to compute the extracted features (cost and time) of the workflow tasks since these features cannot be computed analytically using linear algebra. the objective is to compute the best cost values are predicted based on the current values of the coefficients. In each iteration, new coefficient values are better than that generated in the previous iterations.

In step 3 of the proposed deep learning based optimization algorithm, the weights are updated based on gradient descent Optimization.

Initially, the coefficient $\left(I_K^{x[n]}\right)$ is initialized to a small random value (0.0). This is the coefficient for the function. The cost of the coefficients is evaluated by putting them into the function and calculating the cost (or time) i.e.

$$\text{cost} = f(\text{coefficient}) \quad (5)$$

Then, the derivative of the cost is calculated. The derivative refers to the slope of the function at a given point. The slope is predicted to know the direction (+ve or -ve) in which the coefficient values are to be computed in order to lower the cost (or time) in the next iteration. We need to know the slope (uphill) so that the direction (sign) in which the coefficient values have to be moved in order to get a lower cost on the next iteration. Then, the derivative of the function is taken which act as the downhill of the function, thus the updated coefficient values are given as follows:

$$\text{delta} = \text{derivative}(\text{cost}) \quad (6)$$

In present work, this updation has been taken as updation in weights as:

$$\Delta W_{x[n]}^{K} = d\left(I_K^{x[n]}\right) y_x^n \quad (7)$$

where, $\Delta W \leftarrow$ Update Weights

Further, a learning rate parameter (a) has been specified which value by which coefficients can change on each update, given as:

$$\text{coefficient} = \text{coefficient} - (a * \text{delta}) \quad (8)$$

This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough.

Finally, the schedule of the input tasks has been predicted based on proposed Model. The model is based on the optimization function (gradient descent) which minimizes the cost and time parameters of the input class (tasks). The proposed model has been analyzed on the basis of time and cost computed during execution of workflow tasks.

VII. Results and Discussions

The performance of the algorithms has been analyzed on the basis of the simulation results generated after implementing the deep learning based framework for predicting an optimal schedule of the input workflow (Genome) tasks. The deep learning model has been implemented in Python (PyCharm) and "Tensorflow" has been used as the backend to store the training data of tasks. The schedule predicted through deep learning approach is compared with the earlier approaches proposed in this dissertation. The hybrid-heuristic-metaheuristic approaches have been implemented in cloud workflow simulator.

The makespan results of proposed framework based on deep learning and hybrid-heuristic- metaheuristic approaches has been shown in table 2. The number of VMs has been taken dynamically as 2, 4, 8, 16 and 32. It has been analysed that the results of the proposed load balancing model based on deep learning algorithm are better than the previous model for load balancing in proposed framework which was based on hybrid-heuristic-metaheuristic approaches.

Table 2. Makespan (milliseconds) based comparison between DLD-PLB Framework based on Deep Learning and HDD-PLB Framework for Workflow based on Hybrid Heuristic-Metaheuristic implemented in Cloud Workflow Simulator (CWS)

Makespan Comparison		
No. of VMs	DLD-PLBFW (Deep Learning)	HDD-PLBFW (CWS)
2	311	6426.21055
4	165	19246.99905
8	102	33886.77655
16	81	61839.59345
32	79	259757

Table 3. Cost (in Rs.) based comparison between DLD-PLB Framework based on Deep Learning and HDD-PLB Framework based on Hybrid Heuristic-Metaheuristic implemented in Cloud Workflow Simulator (CWS)

	COST Comparison	
No. of VMs	DLD-PLBFW (Deep Learning)	HDD-PLBFW (CWS)
2	32.655	31.72
4	34.65	71.66
8	42.84	373.42
16	68.04	706.34
32	132.72	1266

The results of both the techniques show similar pattern such that the makespan increases with increased number of VMs. The increase in makespan with increased number of VMs is because of the more workflow tasks have been executed by the proposed framework when more VMs (as a resource) are available for execution. The VMs has been increased exponentially with initial 2 VMs till 32 VMs. Makespan results of proposed framework based in DL approach show substantial

decrease than the other based on metaheuristic approach. It shows that the training of input data pertains to optimized results as compare to the results based on hybrid-heuristic-metaheuristic approaches. Although, deep learning algorithm takes more time in training the input tasks, but the execution of the proposed framework with trained data produces far better results of the parameters. Similarly, the cost results for the proposed framework implemented using two different techniques have also been analysed. The results have been presented in table 3. It has been analysed that the proposed framework with load balancing model based on deep learning approach are more cost optimal as compare to the other proposed hybrid approach.

The figure 5 shows the cost result patterns for the two separate approaches followed for the implementation of the proposed framework of workflow scheduling and load balancing.

VIII. Conclusion and Future Work

Novel approach for load balancing optimization in cloud environment has also been proposed based on Deep Learning Algorithm. It has been concluded that the results of the proposed load balancing model based on deep learning algorithm are better than the previous model for load balancing in proposed framework which was based on hybrid-heuristic-metaheuristic approaches. . The proposed DLD-PLB framework based on deep learning algorithm in this research work has raised a number of challenges for further research. Since, cloud computing leverages advances in computing technology, particularly, internet-based technology and provides services in the form of IAAS, PAAS and SAAS. Other than these cloud service, Machine Learning-as-a-Service (MLAAS) is an emerging paradigm of cloud computing. The cloud services can be used to store, process huge dataset to train the tasks and perform deep learning. Cloud computing architecture is well suited to its key players (Google, Amazon and Microsoft) who are using deep learning techniques to effectively handle huge amount of data i.e. Big Data. The three cloud services giants have already launched their MLAAS versions- Amazon Machine Learning services, Azure Machine Learning, and Google Cloud Artificial Intelligence that allow for fast model training and deployment with little to no data science expertise. Since cloud computing architecture provides unlimited computing resources and supports virtualization, scalability and stores huge data both structured and unstructured so it act as a right platform for deep learning.

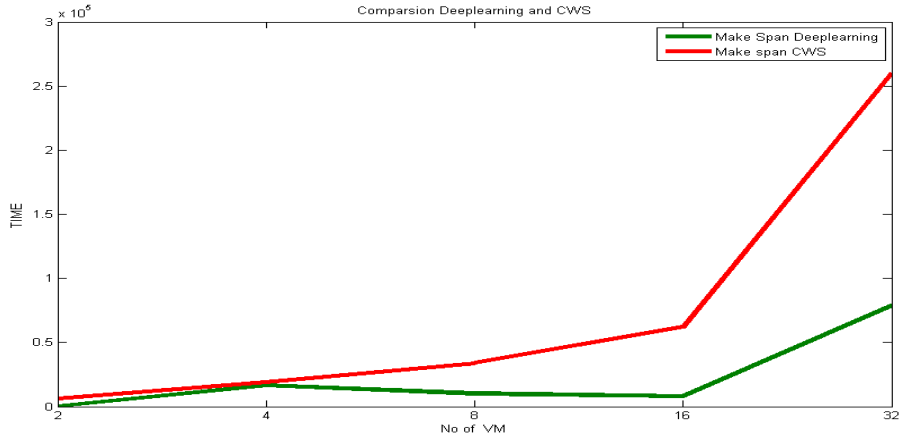

Fig.4. Average Makespan (milliseconds) Analysis of proposed HDD-PLB Framework and DLD-PLB Framework

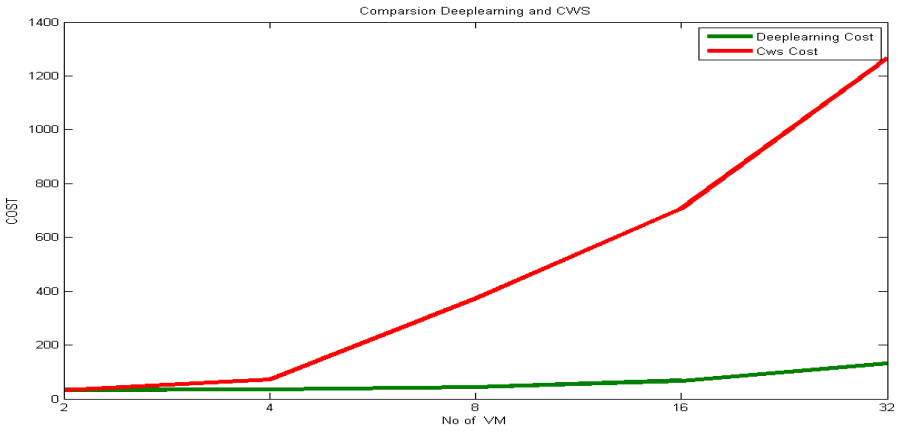

Fig.5. Average Cost (in Rs.) Analysis of proposed HDD-PLB Framework and DLD-PLB Framework

REFERENCES

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J. & Kudlur, M. (2016), "Tensorflow: A System for Large-scale Machine Learning", In OSDI, Vol. 16, pp. 265-283.

[2] Awad, A. I., El-Hefnawy, N. A., & Abdel-Kader, H. M. (2015), "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments", Procedia Computer Science, 65, pp. 920–929.

[3] Chalack, V. A. (2017), "Resource Allocation in Cloud Environment using Approaches based Particle Swarm Optimization", International Journal of Computer Applications Technology and Research, 6(2), pp. 87–90.

[4] Duan, J., & Yang, Y. (2017), "A Load Balancing and Multi-Tenancy Oriented Data Center Virtualization Framework", IEEE Transactions on Parallel and Distributed Systems, 28(8), pp. 2131–2144.

[5] Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., &Turrini, E., (2010), "QoS–Aware Clouds", in Proceedings of IEEE 3rd International Conference on Cloud Computing, pp. 321-328.

[6] Fister, I., Fister, D., & Yang, X. S. (2013), "A Hybrid Bat Algorithm", Elektrotehniski Vestnik / Electrotechnical Review, 80(1–2), pp. 1–7.

[7] Garc á-Gonzalo, E., & Fern ández-Mart ínez, J. L., (2012), "A Brief Historical Review of Particle Swarm Optimization (PSO)", Journal of Bioinformatics and Intelligent Control, 1(1), pp. 3–16.

[8] Ghomi, E. J., Rahmani, A. M., & Qader, N. N., (2017), "Load-balancing algorithms in cloud computing: A survey", Journal of Network and Computer Applications, 88, pp. 50-71.

[9] Gomez, C., Shami, A., & Wang, X. (2018)., " Machine Learning Aided Scheme for Load Balancing in Dense IoT Networks", Sensors, 18(11), 3779.

[10] Guo, Q. (2017), "Task Scheduling Based on Ant Colony Optimization in Cloud Environment", Proceedings of AIP Conference, Volume 1834 AIP Publishing.

[11] Gu, J., Hu, J., Zhao, T., & Sun, G., (2012), "A New Resource Scheduling Strategy based on Genetic Algorithm in Cloud Computing Environment", Journal of Computers, 7(1), pp. 42–52.

[12] Hou, X., & Zhao, G. (2018), "Resource Scheduling and Load Balancing Fusion Algorithm with Deep Learning Based on Cloud Computing", International Journal of Information Technology and Web Engineering (IJITWE), 13(3), 54-72.

[13] Hsiao, H. C., Chung, H. Y., Shen, H., & Chao, Y. C. (2013), "Load Rebalancing for Distributed File Systems in Clouds" IEEE Transactions on Parallel and Distributed Systems, 24(5), pp. 951–962.

[14] Hung, C., Wang, H., & Hu, Y. (2012), "Efficient Load Balancing Algorithm for Cloud Computing Network", Proceedings of International Conference on Information Science and Technology (IST 2012), pp. 28–30.

[15] Jaikar, A., Dada, H., Kim, G. R., & Noh, S. Y. (2014), "Priority-based Virtual Machine Load Balancing in a Scientific Federated Cloud", IEEE 3rd International Conference on Cloud Networking, CloudNet 2014, pp. 248–254.

[16] Kaur A., Kaur B., Singh D. (2018), "Meta-heuristics based Load Balancing Optimization in Cloud Environment on Underflow and Overflow Conditions", Journal of Information Technology Research (JITR) (IGI Global), Vol.11(4) pp. 155-172.

[17] Kaur A., Kaur B., Singh D. (2018), "Comparative Analysis of Metaheuristics Based Load Balancing Optimization in Cloud Environment", Smart and Innovative Trends in Next Generation Computing Technologies- Communications in Computer and Information Science (CCIS) Springer, Singapore, vol. 827. pp. 30-46.

[18] Keshvadi, S., & Faghih, B. (2016), "A Multi-agent based Load Balancing System in IaaS Cloud Environment, International Robotics & Automation Journal, 1(1), pp. 1-6.

[19] Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011), "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization", Proceedings of Sixth Annual Chinagrid Conference, pp. 3–9.

[20] Matsumoto, H. & Ezaki, Y., (2011), "Dynamic Resource management in cloud Environment". Fujitsu Science Technology Journal, 47(3), pp. 270-276.

[21] Milani, A. S., & Navimipour, N. J. (2016). "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends", Journal of Network and Computer Applications, 71, 86-98.

[22] Mittal, S., & Dubey, P. M., (2017), "AMO Based Load Balancing Approach in Cloud Computing", IOSR Journal of Computer Engineering, 19(2), pp. 62–66.

[23] Mohamed, N., & Al-Jaroodi, J. (2011), "Delay-Tolerant Dynamic Load Balancing", Proceedings of IEEE International Conference on High Performance Computing and Communications, pp. 237–245.

[24] Pacini, E., Mateos, C., & Garc á Garino, C. (2015), "Balancing Throughput and Response Time in Online Scientific Clouds via Ant Colony Optimization", Advances in Engineering Software, 84, pp. 31–47.

[25] Roman, M., Habib, A., Ashraf, J., & Ali, G. (2016), "Load Balancing in Partner-Based Scheduling Algorithm for Grid Workflow", International Journal of Advanced Computer Science and Applications, 7(5), pp. 444–453.

[26] Singh, A. B., Bhat, S., Raju, R., & D'Souza, R. (2017), "Survey on Various Load Balancing Techniques in Cloud Computing", Advances in Computing, 7(2), pp. 28-34.

[27] Yang, R., Ouyang, X., Chen, Y., Townend, P., & Xu, J., (2018), "Intelligent Resource Scheduling at Scale: A Machine Learning Perspective", in Proceedings of 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 132-141.

Authors' Profiles

Amanpreet Kaur is pursuing PhD in the area of cloud computing from IK Gujral Punjab Technical University, Jalandhar. She has outstanding academic record with merit positions in B.Tech (CSE) and M.Tech (IT) from Guru Nanak Dev University, Amritsar. She has contributed more than 40 research papers in renowned International journals. She has more than 15 years of teaching experience in professional institutes has also received faculty excellence awards in the year 2010 and recently in 2018. She is life member of many professional bodies of technical education and research. Her interests are in areas of cloud computing, Fog Computing and SDN.

Dr. Bikrampal Kaur is Professor in Chandigarh Engineering College, Landran, Mohali. She holds the degrees of B.Tech., M.Tech, and M.Phil. She completed her Ph.D.in the field of Information Systems from Punjabi University, Patiala. She has more than 19 years of teaching experience and served many academic institutions. She is an Active Researcher who has supervised many MCA/M.Tech. dissertations and guiding Ph.D. research scholars. She contributed more than 80 research papers in various national & international journals/conferences. Her areas of interest are Information System, ERP.

Dr. Parminder Singh is a young dynamic personality with a proven record of a good academician and researcher having an outstanding academic record. He has been working as Associate Professor in Information Technology Department and has more than Fourteen years of rich experience as an academician and researcher. He has published over 70 Journal and conference papers in the areas of Networking, Wireless Networks, sensor computing and Network security. He holds two patents deriving from his research. Dr. Singh has published three books on his research activities. He completed three research projects, including one DST project. He has conducted Webinar Sessions related to Network Programming and Software Defined Networks in association with CISCO, Systems to have academia-industry Interaction. He has won best-paper awards including the IEEE "Best Paper Award" in the Year 2012 and 2014. He received "Young Teacher Award" in International Conference ICIC-2018. He has also received faculty excellence and research awards in the year 2011, 2013, 2015, 2016, 2017, 2018 and 2019 from different organizations for excellence in research, teaching and service.

Mandeep Singh Devgan has been working as Assistant Professor in CGC Landran, Mohali in department of Information Technology. He has done his M.tech (CSE) degree from BBSBEC, Sri Fatehgarh Sahib and pursuing Ph.D from IKGPTU. He has published more than 30 research papers in well-known International journals and Conferences. He has more than 13 years of teaching experience and received faculty excellence awards in the year 2015. His areas of interest are cloud computing, network security, Software Architecture.

Harpreet Kaur Toor is an Assistant professor in CEC, Landran, Mohali. She has 7 year of experience in field of teaching in mathematics. She has published various paper in International Journals and filed 2 patents in last 2 years. She has got Best Teacher Award in year 2018.

How to cite this paper: Amanpreet Kaur, Bikrampal Kaur, Parminder Singh, Mandeep Singh Devgan, Harpreet Kaur Toor, "Load Balancing Optimization Based On Deep Learning Approach in Cloud Environment", International Journal of Information Technology and Computer Science(IJITCS), Vol.12, No.3, pp.8-18, 2020. DOI: 10.5815/ijitcs.2020.03.02