Measurement: Sensors 33 (2024) 101076

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

GEP optimization for load balancing of virtual machines (LBVM) in cloud computing

G. Muneeswari [a,*], Jhansi Bharathi Madavarapu [b], R. Ramani [c], C. Rajeshkumar [d], C. John Clement Singh [e]

[a] Department School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India
[b] Department of Information Technology, University of Cumberlands, 6178 College Station Drive, Williamsburg, KY, 40769, USA
[c] Department of Computer Science and Engineering, P.S.R Engineering College, Sivakasi, Tamil Nadu, 626140, India
[d] Department of Information Technology, Sri Krishna College of Technology, Coimbatore, 641042, Tamilnadu, India
[e] Electronics and Communication Engineering, Kings Engineering College, Sriperumbudur, Chennai, 602117, Tamilnadu, India

ARTICLE INFO

Keywords:
Load balancing
Cloud computing
Virtual machines
Bi-LSTM
Genetic expression programming

ABSTRACT

Cloud computing relies heavily on load balancing to distribute workloads evenly among servers, network connections, and drives. The cloud system has been assigned some load which can be underloaded, overloaded, or balanced depending on the cloud architecture and user requests. An important component of task scheduling in clouds is the load balancing of workloads that may be dependent or independent of virtual machines (VMs). To overcome these drawbacks, a novel Load Balancing of Virtual Machine (LBVM) in Cloud Computing has been proposed in this paper. The input tasks from multiple users were collected in a single task collector and sent towards the load balancer, which contains the deep learning network called the Bi-LSTM technique. When the load is unbalanced, the VM migration will begin by sending the task details to the load balancer. The Bi-LSTM is optimized by a Genetic Expression Programming (GEP) optimizer and finally, it balances the input loads in VMs. The efficiency of the proposed LBVM has been determined using the existing techniques such as MVM, PLBVM, and VMIS in terms of evaluation metrics such as configuration latency, detection rate, accuracy etc. Experimental results shows that the proposed method reduces the Migration Time of 49%, 41.7%, and 17.8% than MVM, PLBVM, VMIS existing techniques respectively.

1. Introduction

Cloud services that may be accessible online are how computational hardware and software resources are packaged in cloud computing. There are three different types of cloud computing applications that have been developed: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [1–3]. In order to optimize [4–6] streamline operations and deliver acceptable levels of speed for the consumers, handling massive data sets necessitates a number of strategies. By maintaining effective management of cloud resources, it is possible to accomplish the efficient and scalable properties of cloud computing [7]. Users might cite better utilisation of distributed resources and their application to increase throughput [8], performance [9], and problem-solving at a big scale [10] as the primary objectives of cloud computing (see Figs. 2–7).

The chances of failures that could simultaneously harm the services in cloud systems are reduced by the use of load balancing [11] and redundant mirrored databases in cluster techniques [12], which span several availability zones. The load balancer can move to another resource if one system has an outage [13]. By maximising resource availability and minimising the amount of downtime experienced by organisations during outages, load balancing techniques among the circumstances towards cloud computing help to lower expenses related to document management systems [14,15].

Although load balancing is essentially necessary in a cloud setting, there are several difficulties with it. Scalability is one of the most attractive benefits of both cloud computing and LB, but it is another one of the obvious drawbacks of the latter. In the majority of load balancers, the scalability is limited by a small number of nodes for distributing processes. Performance degradation because load balancers provide different resources with equal or predetermined weights, which can lead to slow speed and high costs [26,27]. Due to the rapid fluctuations in

* Corresponding author.
E-mail address: muneeswari.g.vitap@gmail.com (G. Muneeswari).

https://doi.org/10.1016/j.measen.2024.101076
Received 21 August 2023; Received in revised form 30 January 2024; Accepted 23 February 2024
Available online 29 February 2024

demand for process resources, it is not possible to predict or estimate the load or total number of processes on a site. Security-related issues are also covered, Attackers may take advantage.

Numerous studies have employed machine learning methods to address load balancing and security concerns. Machine learning is the process of gathering and analyzing data using sophisticated algorithms in order to create intelligence [27,28]. In classical machine learning (ML), features are extracted from input using intricate mathematical procedures, and then the features are categorized. ML is inefficient and less successful in producing ideal outcomes since it consists of two phases: the feature extraction phase and the classification phase. To overcome these drawbacks, Load Balancing of Virtual Machine (LBVM) in Cloud Computing has been proposed in this paper. The major contributions of the paper are given as follows.

- The process of Load balancing starts from the collecting set of tasks in which data are given by multiple users.
- In load balancing, the proposed Bi-LSTM and GEP optimization have been performed, and the Bi-LSTM is optimized by the GEP optimizer.
- The optimized cloud resources are given to VMs that perform in the cloud and there the VMs balance the tasks, When the load is unbalanced, VM migration is done.
- The efficiency of the proposed LBVM has been determined using evaluation metrics such as configuration latency, detection rate, accuracy etc.

This paper is structured as follows. The literature review can be found in Section II. The proposed approach is presented in Section III. Results and a discussion are presented in Part IV. Section V presents the conclusions.

2. Literature survey

By deploying the software to the hardware load-balancing device on the virtual machine, the Virtual Load Balancer (VLB) differs from a software load balancer. Many factors resulted in the creation of load balancing. Many studies have been conducted to solve this problem. Among those, some of the techniques have been reviewed in this section.

In 2018, M. Kandhimathi, D.V et al. [16], had proposed in order to allocate the finest virtual machines to fulfil the request in a very efficient and quick manner, the additional virtual machines were added using a genetic technique. The overall energy consumption either when they were idle. The proposed approach outperforms the compared current algorithms, according to simulation findings. This method does not have computational offloading capability.

In 2018, Mehiar Dabbagh, B et al. [17], proposed a system for integrated resource allocation that is energy-efficient for overcommitted clouds. utilizing authentic Google data made up of 29-day sign gathered upon a cluster with in excess of 12K PMs. By reducing the occurrence of unexpected overloads, the suggested framework performs better than prior VM migration techniques and current overload avoidance tactics schemes and significantly diminishing on cloud energy use. The proposed work is unable to decrease costs when the deadline is loose.

In 2019, Wei Guo, W et al. [18], has a modified version of LSTM called N-LSTM is suggested in order to handle the challenges of predicting the workload of virtual machines. This method of problem-solving combines the historical workload of several types of VM across the specified duration period. In comparison to the long-term memory method, the suggested method can produce predictions that are more accurate. The proposed work doesn't satisfy highly dynamic user requirements.

In 2019, Praneeth Gunupati, K et al. [19], proposed Hugo is a model that serves as a distributed, scalable and flexible cloud controller that can control basic operations including the creation, removal, and transfer of virtual machines (VMs). It does it by over-committing the Physical Machine (PM) to get greater usage. As a result, the proposed Load Balancing with VM Migration is successfully executed. The proposed work considers makespan, execution time, and energy consumption as secondary performance indicators.

In 2020, Vincent Kherbache, E et al. [20], proposed migration of Virtual Machines, the migrations were parallelized and sequentialized by mVM according to the memory burden and the network topology to offer schedules with the shortest completion times. The suggested method's findings show that the mVM solving time only makes up roughly 1% of the scheduled execution time. The proposed work is not suitable for real-time applications.

In 2020, Bodrul Alam, T et al. [21], suggested a Markov-based failure prediction model to foresee cloud server failure. Based on its past data, the model predicts that the server base state would deteriorate. The optimization issue is resolved using the Artificial Bee Colony (ABC) technique both optimally and heuristically. As a result, the approach improves dependability and reduces communication lag during VM service migration. The proposed work lacks a large amount of practical data.

In 2021, P. Tamilarasi, D.A et al. [22], designed for the large data cloud environment, a prediction-based load balancing and virtual machine (VM) migration technique (PLBVM) was developed. This paper proposed for the Investigators has provided more consideration towards the harmonizing of the load of complete impact on the system act. The result shows that the proposed PLBVM achieves lesser response delay and execution time. The proposed work does not consider the distribution of resources in dynamic conditions.

In 2021, Deepika Saxena, A et al. [23], proposed an efficient resource management framework to handle the problems and enhance data center performance. The framework's performance is assessed by doing tests on three types of real-world workload data, including the Google Cluster dataset, Planet Lab, and Bit Brains VM traces. The OP-MLB framework outperforms the best fit technique in terms of power savings by up to 85.3%. The proposed work has only fewer characteristics between user requests and cloud service systems.

In 2021, P. Joseph Charles, U. L et al. [24], had proposed the Heuristic algorithms which solve the VM consolidation, this approach to solve the issue in bin packing method. To solve the problem, a variety of alternative heuristics algorithms are available. Instead of bandwidth and memory, the CPU utilisation of virtual machines is primarily taken into account when calculating energy consumption. Data in the proposed cloud environments does not reflect real-world conditions.

In 2022, A. Pandyaraj, N. V et al. [25], proposed Using the allocated resources, the virtual machine for infrastructure service in the cloud network is employed. By using virtual climate, the jib may execute the task according to the resources' accessibility and react more quickly. By using an online prediction model, it was possible to estimate the job sizes and so cut down on running time. The proposed work does not consider migration time, and overhead time parameters.

From these literature studies, the author's usage of cloud computing has made it possible for businesses to allocate resources among numerous servers in order to manage workload or application requirements. The authors used some different type of techniques to manage the work load or load balancing such as, Migration of Virtual Machines, PLBVM, Heuristic algorithms etc. The load balancing strategies now in use rely on a number of task parameters. Most methods balance load by using optimization techniques. A review of the literature reveals that deep learning-based load balancing, on the other hand, may greatly enhance the workload distribution that is balanced. To overcome these drawbacks a novel LBVM technique has been proposed in the next section.

3. Proposed method

The proposed Load Balancing of Virtual Machine (LBVM) in Cloud Computing is given detailed in this section. Load Balancing in Cloud is a distributed process for the software-defined, managed service for all of users given tasks traffic. The process of Load balancing is starts from the collecting set of tasks which data are given by multiple users. In load balancing, the proposed Bi-LSTM and GEP optimization are performed, the Bi-LSTM is the optimized by GEP optimizer. The optimized cloud resources are goes to VMs which performs in cloud based and there the VMs balances the tasks, When the load was unbalanced, VM migration was done. Finally, the input tasks and the cloud resources are balanced and executed which is shown in Fig. 1

3.1. Users and tasks

The input data are collected from different kinds of users, the collected data are called tasks. The tasks are stored in a single task collector which collects a number of data from different users.

3.2. Load balancer

The Load Balancing in the Cloud is a process distributed for software-defined, managed service for all the users given tasks traffic. The Load

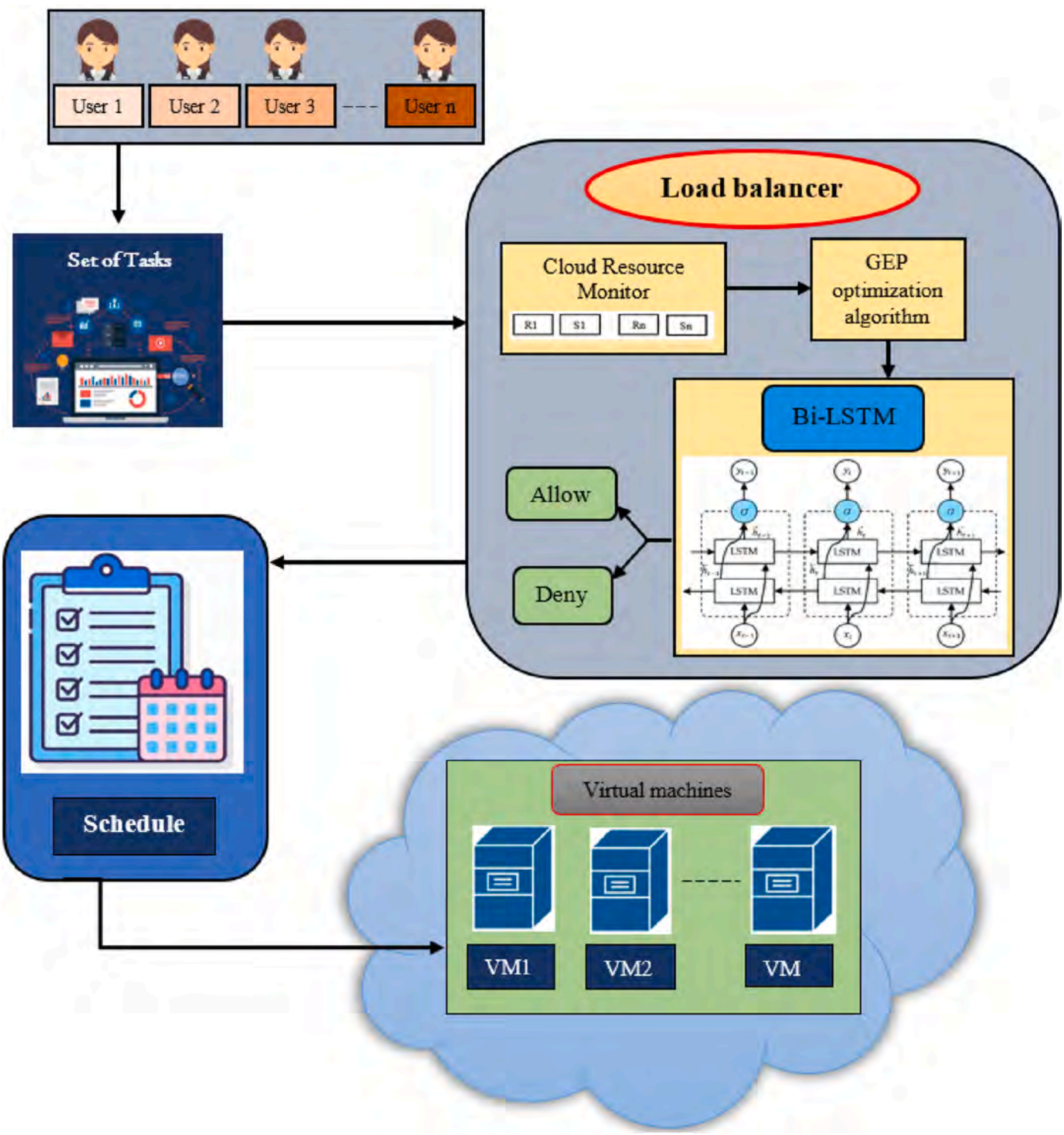

Fig. 1. Proposed load balancing diagram.

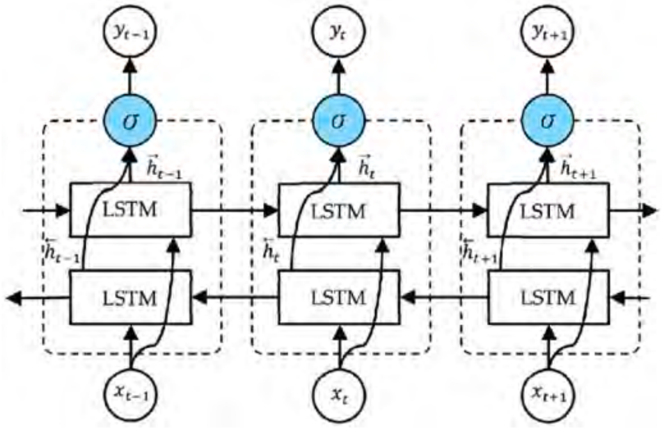

Fig. 2. Bi-LSTM diagram.

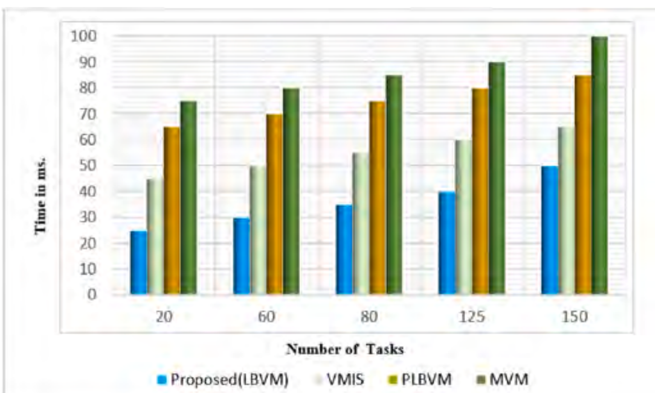

Fig. 3. Times in ms with number of tasks.

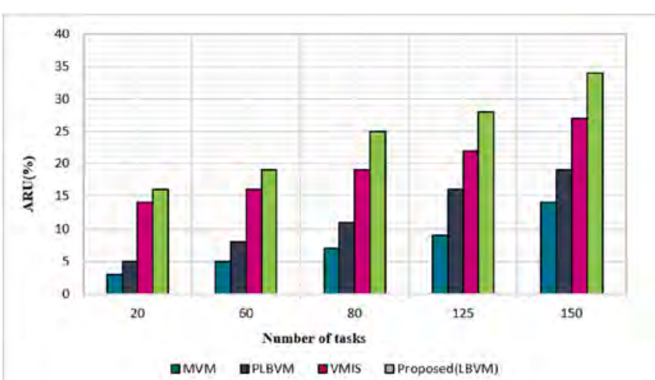

Fig. 4. ARU with number of tasks.

balancer balances the input loads with Bi-LSTM which is optimized by GEP optimizer. At first the cloud resource monitor monitors the tasks which calls as resources R. Then the monitored cloud resource goes to GEP optimizer which optimize the Bi-LSTM.

3.2.1. GEP optimization algorithm

A variant of GP is genetic expression programming (GEP). A collection of terminals, intake variables or constants, and operator functions, GEP encodes programmes or their formulas as binary trees expressed as linear gene expressions. The head and tail of a binary tree make up the level-order traversal of the gene expression. While symbols in the tail can only be terminals, those in the head can represent functions as well.

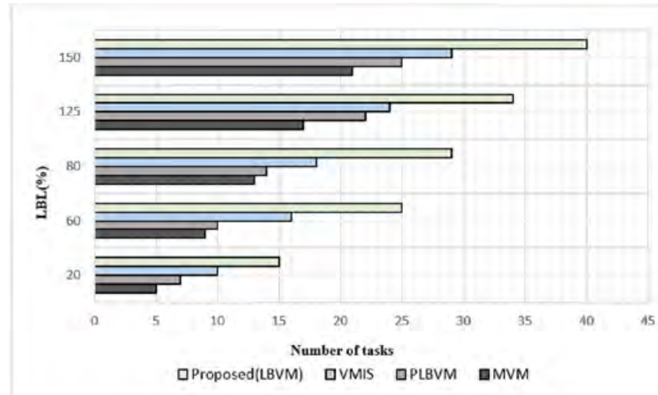

Fig. 5. LBL with number of tasks.

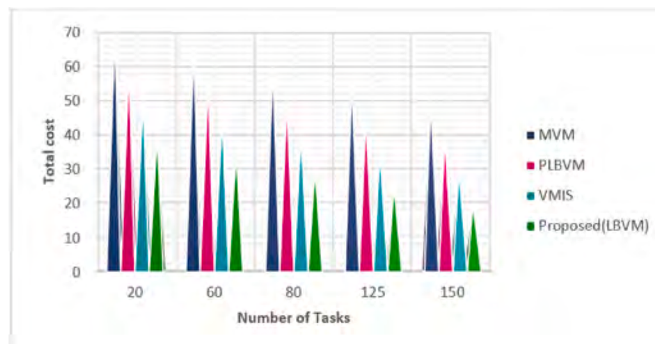

Fig. 6. Total cost with number of tasks.

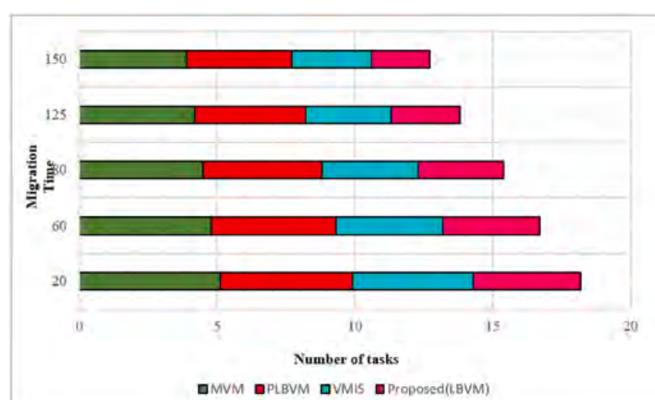

Fig. 7. Migration time with number of tasks.

A gene expression has a length of $h + t$, where h is the user-provided length of the head and $t = h \times (n - 1) + 1$ is the length the length of tail. The most important number of arguments a user can send to a function is represented by the parameter n. In this case, the operators $+, -, \times, \div, \sqrt{}$ and the terminals are the symbol x, y and numbers. There could be some unneeded terminals in the tail portion. Procreation and genetic variation processes are the same as in GA. Expression trees are used in GEP to increase efficiency since a straightforward linear structure makes it easier to apply genetic processes.

3.2.2. Bi-LSTM

A sequence processing model called Bi-LSTM consists of two LSTMs. One processing the information forward and the other processing it backward. The network can access more data with the aid of Bi-LSTMs, which is advantageous for the context of the algorithm. The two hidden LSTM layers are linked to the output layer by bidirectional LSTM (Bi-LSTM). In the application, using two LSTM as one layer encourages improving the learning long-term dependency, which subsequently will increase model performance.

The output sequence $\overleftarrow{h}$ of the backward LSTM layer is computed using the inverse input from time $t-1$ to $t-n$. Since the output sequence $\overrightarrow{h}$ of the forward LSTM layer is acquired in the same manner as in the unidirectional case. The σ function receives these output sequences, which are then merged to create the output vector y_t. Like the LSTM layer, the vector $Y_t = y_{t-n}, \ldots, y_{t-1}$ may be used to describe the Bi-LSTM layer's final output. The expected blood pressure for the subsequent iteration is found in the last element, y_{t-1}.

Given input tasks $X = (x_1 \ldots, x_T)$, the hidden vector tasks $h = (h_1, \ldots, h_T)$ and the output vector tasks $Y = (y_1, \ldots, y_T)$ by iterating the following equations from $t = 1$ to T.

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = W_{hy}h_t + b_0 \tag{2}$$

The weight matrix is denoted by the W term, the bias vector is shown by the B term, and the hidden layer function is represented by the H term. H is usually a sigmoid function applied element by element.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1}b_f\right) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{5}$$

$$O_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{6}$$

$$h_t = O_t \tanh\left(c_t\right) \tag{7}$$

The logistic sigmoid function in this case is denoted by σ, and the input, forget, output, and cell activation vectors, f, o , and c, are all of the same size as the hidden vector h. The input/output gate matrix is W_{x0}, while the hidden input gate matrix is W_{hi}.

The optimized Bi-LSTM then go to the scheduling section, the Bi-LSTM send the optimized task and there the schedule will decide to send to the VMs whether it taken or not. While running as a process on the host operating system, a virtual machine functions as a distinct, independent machine. When the load was unbalanced, VM migration was done. It is an easy approach to provide a share of cloud resources to a specific task. In order to allocate tasks to virtual machines (VMs), a forecast timetable for the input task is finally produced using the suggested model.

4. Result and discussion

Simulations have been performing to evaluate the proposed LBVM in cloud computing using GEP optimization techniques. Here, the Bi-LSTM technique is used because it offers additional training of tasks and thus Bi-LSTM based modelling offers better prediction. The input tasks from multiple users were collected in a single task collector and sent towards to the load balancer, there it performs the Bi-LSTM technique by using Genetic Expression Programming (GEP) optimization algorithm and the collected schedules sent to Virtual Machines (VMs) and finally it balanced the input loads in VMs, When the load was unbalanced, VM migration was done.

The optimized Bi-LSTM then go to the scheduling section, the Bi-LSTM send the optimized task. While running as a process on the host operating system, a virtual machine functions as a distinct, independent machine. It's an easy approach to provide a share of cloud resources to a specific task. According to the results, better prediction is provided by Bi-LSTM based modelling than by traditional LSTM based models because it is based on extra data training, provides better prediction than conventional LSTM based models. Tensor Flow is used as the backend to store task training data in the Python (PyCharm) implementation of the suggested load balancing approach. The suggested framework incorporates TensorFlow into its load balancing and scheduling models to handle massive datasets.

4.1. *Performance metrics*

This section presents the simulation results of proposed Load Balancing of Virtual Machine (LBVM) performance metrics namely, Times in ms, ARU (%), LBL (%), Total cost and Migration time.

4.1.1. *Times in ms*

The number of tasks is Compared with other three existing techniques which are MVM [20], PLBVM [22], VMIS [24] and the proposed Load Balancing of Virtual Machine (LBVM). Compared with other existing techniques the times in ms performance is lower.

4.1.2. *ARU with number of tasks*

The number of tasks is Compared with other three existing techniques which are MVM, PLBVM, VMIS and the proposed Load Balancing of Virtual Machine (LBVM). Compared with other existing techniques the ARU performance is higher.

4.1.3. *LBL*

The number of tasks is Compared with other three existing techniques which are MVM, PLBVM, VMIS and the proposed Load Balancing of Virtual Machine (LBVM). Compared with other existing techniques the LBL performance is higher.

4.1.4. *Total cost*

The number of tasks is Compared with other three existing techniques which are MVM, PLBVM, VMIS and the proposed Load Balancing of Virtual Machine (LBVM). Compared with other existing techniques the total cost is lower.

4.1.5. *Migration cost*

The number of tasks is Compared with other three existing techniques which are MVM, PLBVM, VMIS and the proposed Load Balancing of Virtual Machine (LBVM). Compared with other existing techniques the migration time is lower.

The results section shows that the suggested LBVM technique outperforms previous load balancing methods by utilizing the cloud environment, which offers limitless computing resources, virtualization, scalability, and the ability to store large amounts of structured data. It is unquestionably superior to unstructured data. Nevertheless, the suggested work does not utilize dependent works. We intend to enhance this type of load balancing for buildings that have dependent activities in the future.

5. Conclusion

Load Balancing of Virtual Machine (LBVM) in Cloud Computing has been proposed in this paper. The input tasks from multiple users were collected in a single task collector and sent towards to the load balancer, there it perform the Bi-LSTM technique by using Genetic Expression Programming (GEP) optimization algorithm and the collected schedules sent to Virtual Machines (VMs) and finally it balances the input loads in VMs, When the load was unbalanced, VM migration was done. According to the results, better prediction is provided by Bi-LSTM based modelling than by traditional LSTM based models because it is based on extra data training, provides better prediction than conventional LSTM based models. In load balancing, the proposed Bi-LSTM and GEP optimization are performed, the Bi-LSTM is the optimized by GEP optimizer. The optimized cloud resources are goes to VMs which performs in cloud based and there the VMs balances the tasks. The proposed method has been evaluated in terms of MVM, PLBVM, VMIS and the proposed Load Balancing of Virtual Machine (LBVM). The proposed method reduces the Migration Time of 49%, 41.7%, 17.8% than MVM, PLBVM, VMIS existing techniques. Because cloud computing offers virtualization, scalability, infinite computational resources, and the capacity to store massive volumes of both organized and unstructured data, it is perfect for deep learning. We intend to enhance this type of load balancing for buildings that have dependent activities in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

References

[1] S. Malla, K. Christensen, HPC in the cloud: performance comparison of function as a service (FaaS) vs infrastructure as a service (IaaS), Int. Technol. Lett. 3 (1) (2020) 137.
[2] R. Yasrab, Platform-as-a-service (Paas): the Next Hype of Cloud Computing, 2018 arXiv preprint arXiv:1804.10811.
[3] R.S.R.J. Raghavan, R.V. Nargundkar, Impact of software as a service (SaaS) on software acquisition process, J. Bus. Ind. Market. 35 (4) (2020) 757–770.
[4] D. Ardagna, M. Ciavotta, R. Lancellotti, M. Guerriero, A hierarchical receding horizon algorithm for QoS-driven control of multi-IaaS applications, IEEE Trans. Cloud Comput. 9 (2) (2018) 418–434.
[5] H. Yang, C. Lou, L. Sun, J. Li, Y. Cai, Z. Wang, W. Li, G. Liu, Y. Tang, admetSAR 2.0: web-service for prediction and optimization of chemical ADMET properties, Bioinformatics 35 (6) (2019) 1067–1069.
[6] D. Hendrycks, N. Mu, E.D. Cubuk, B. Zoph, J. Gilmer and B. Lakshminarayanan, Augmix: A Simple Data Processing Method to Improve Robustness and Uncertainty. arXiv preprint arXiv:1912.02781.
[7] A. Rashid, A. Chaturvedi, Cloud computing characteristics and services: a brief review, Int. J. Comput. Sci. Eng. 7 (2) (2019) 421–426.
[8] H. Rajab, T. Cinkler, T. Bouguera, IoT scheduling for higher throughput and lower transmission power, Wireless Network 27 (2021) 1701–1714.
[9] O.M. Ahmed, L.M. Haji, H.M. Shukur, R.R. Zebari, S.M. Abas, M.A. Sadeeq, Comparison among cloud technologies and cloud performance, J. Appl. Sci. Technol. Trends 1 (2020) 40–47.
[10] B. Langmead, A. Nellore, Cloud computing for genomic data analysis and collaboration, Nat. Rev. Genet. 19 (4) (2018) 208–219.
[11] J. Zhang, F.R. Yu, S. Wang, T. Huang, Z. Liu, Y. Liu, Load balancing in data center networks: a survey, IEEE Commun. Surveys & Tutor. 20 (3) (2018) 2324–2352.
[12] P. Govender, V. Sivakumar, Application of k-means and hierarchical clustering techniques for analysis of air pollution: a review (1980–2019), Atmos. Pollut. Res. 11 (1) (2020) 40–56.
[13] A. Balalaie, A. Heydarnoori, P. Jamshidi, D.A. Tamburri, T. Lynn, Microservices migration patterns, Software Pract. Ex. 48 (11) (2018) 2019–2042.
[14] B. Yi, X. Wang, K. Li, M. Huang, A comprehensive survey of network function virtualization, Comput. Network. 133 (2018) 212–262.
[15] D. Ageyev, O. Bondarenko, T. Radivilova, W. Alfroukh, Classification of existing virtualization methods used in telecommunication networks, in: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), IEEE, 2018, pp. 83–86.
[16] M. Kanthimathi, D. Vijayakumar, An enhanced approach of genetic and Ant Colony based load balancing in cloud environment, in: 2018, International Conference on Soft-Computing and Network Security (ICSNS), IEEE, 2018, pp. 1–5.
[17] M. Dabbagh, B. Hamdaoui, M. Guizani, A. Rayes, An energy-efficient VM prediction and migration framework for overcommitted clouds, IEEE Trans. Cloud Comput. 6 (4) (2016) 955–966.
[18] W. Guo, W. Ge, X. Lu, H. Li, Short-term load forecasting of virtual machines based on improved neural network, IEEE Access 7 (2019) 121037–121045.
[19] P. Gunupati K. Boyidi, A. Kumar, S. Earla, S. Gugulothu, A. Singh, Distributed controller for load balancing with VM migration, in: 2019 4th International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2019, pp. 488–495.
[20] V. Kherbache, E. Madelaine, F. Hermenier, Scheduling live migration of virtual machines, IEEE Trans. Cloud Comput. 8 (1) (2017) 282–296.
[21] A.B. Alam, T. Halabi, A. Haque and M. Zulkernine, Optimizing virtual machine migration in multi-clouds. In 2020, International Symposium on Networks, Computers and Communications *(ISNCC)*, 1. IEEE.
[22] P. Tamilarasi, D. Akila, Prediction based load balancing and VM migration in big data cloud environment, in: 2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM), IEEE, 2021, pp. 123–127.
[23] D. Saxena, A.K. Singh, R. Buyya, OP-MLB: an online VM prediction-based multi-objective load balancing framework for resource management at cloud data center, IEEE Trans. Cloud Comput. 10 (4) (2021) 2804–2816, m.
[24] P.J. Charles, U.L. Stanislaus, Secure virtual machine migration using ant Colony algorithm, in: 2021 Fifth International Conference on I-SMAC(IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, 2021, pp. 1571–1575.
[25] A. Pandiaraj, N. Vinothkumar, R. Venkatesan, Virtual machine migration for infrastructure service in cloud network, in: 2022 Smart Technologies, Communication and Robotics (STCR), IEEE, 2022, pp. 1–5.
[26] T. Raj, C. Pushpalatha, A. Ahilan, An optimized profound memory-affiliated de-noising of aerial images through deep neural network for disaster management, Signal, Image Video Process. (2023) 1–9.
[27] A. Agasthian, Rajendra Pamula, L.A. Kumaraswamidhas, Integration of monitoring and security based deep learning network for wind turbine system, Int. J. Syst. Design Comput. 1 (1) (2023) 11–17.
[28] A. Jegatheesh, N. Kopperundevi, M. Anlin Sahaya Infant Tinu, Brain aneurysm detection via firefly optimized spiking neural network, Int. J. Curr. Bio-Med. Eng. 1 (1) (2023) 23–29.