National College of Ireland

Enhancing Load Balancing Efficiency In Dynamic Workload Environments Using Enhanced Genetic Algorithm and Machine Learning

MSc Research Project
MSc Cloud Computing

Shantanu Malviya
Student ID: x22248978

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shantanu Malviya
Student ID:	22248978
Programme:	MSc. Cloud Computing **Year:** 2023-2024
Module:	MSc. Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	12th August, 2024
Project Title:	Enhancing Load Balancing Efficiency In Dynamic Workload Environments Using Enhanced Genetic Algorithm and Machine Learning
Word Count:	7747 **Page Count** 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 12th August, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Load Balancing Efficiency In Dynamic Workload Environments Using Enhanced Genetic Algorithm and Machine Learning

Shantanu Malviya
22248978

Abstract

Cloud computing has gained significant popularity in recent times, providing scalable and elastic IT-enabled resources ′as a service′ to customers. There still exists issues with load balancing in cloud computing while handling dynamically dependent workloads. The improper distribution of loads in the cloud environment ultimately results in depreciation of overall system performance. To achieve this, multiple studies utilising various methods of both Enhanced Genetic Algorithm (EGA) and Machine Learning (ML) have been carried out. These methods often lack the vision and execution of integrating both technologies and consider the outcomes of this integration. This study aims to achieve more effective load balancing in cloud computing networks using an Enhanced Genetic Algorithm (EGA) combined with multiple Machine Learning (ML) targeted for the real-time load balancing problems.

An algorithm that has the capabilities to handle dynamic workloads and is equipped with advanced technologies that make use of predictive analysis can act as research area and a concept which can potentially resolve the issues with current present algorithms. This research proposes an EGA with Selection, Crossover and Mutation techniques in a virtual simulation of a Datacenter with Virtual Machines (VMs) that is capable of providing outputs as Resource Utilisation and Execution Time. The algorithm was executed for ‘n’ iterations to gather dataset for ML model training. The various Machine Learning models results showcased Linear Regression outperforming other techniques to achieve the objective.

1 Introduction

Cloud computing provides adaptable solutions that are open to request from a pool of configurable computing resources, providing solutions that come at a reasonable cost and are easily scalable (Mell, 2011). The volume of information that is generated and need for more efficiency has risen dramatically with the progress in cloud computing technologies. However, these benefits have a lot of complex elements involved that fulfil different functionalities: especially with respect to load balancing, the concept is far from being easy to implement. Load balancing is one of the distinct factors that determine how efficient the cloud IT infrastructures are, how fast their responses are, and how stable the systems are to customer needs. This study proposes a study on enhancing the load balancing efficiency in dynamic environments using Enhanced Genetic Algorithm and Machine learning techniques.

Genetic algorithms involving principles drawn from natural selection and genetics bring in high reliability in search and optimization. They can easily scout a vast solution space and make adjustments as required, which renders it (Lambora, 2019) useful in dynamic workload settings. In the same context, machine learning algorithms with the capability to learn from data and make accurate predictions, can complement the deterministic capability of load balancing decision making process by giving recommendations and being able to change based on changes in workload and resource usage.

1.1 Motivation

Software solutions supported by cloud computing are fundamental, ranging from simple storage shelves to data processing systems with artificial intelligence used by large enterprises. Load balancing is very crucial for ensuring that the work load is appropriately divided among servers so that a single server does not become overloaded to the extent that the efficiency and accuracy of the system is compromised (Sajjan, 2017). As cloud data centers expand and computational tasks become more complex, load balancing becomes increasingly crucial. The EGA, an advanced evolutionary algorithm, offers promising solutions for enhancing load distribution and system performance. To achieve optimal resource utilization and meet the diverse needs of cloud computing environments, it is essential to research and adapt the EGA specifically for dynamic and interdependent workloads. The integration of EGA and Machine Learning is an attempt at finding a different method to improve load balancing efficiency.

1.2 Research Question

Can Enhanced Genetic Algorithm and Machine Learning be used to improve load balancing efficiency in dynamic workload environments?

1.3 Research Goals

Load balancing is a key component of cloud computing, ensuring optimal system performance and resource efficiency. In cloud systems, workloads are dynamic and often interdependent, making optimal task distribution challenging (Mishra, 2020).

- The main goal of this research is to improve load balancing efficiency in dynamic workload environments by using an enhanced genetic algorithm (EGA) along with machine learning techniques.

- The EGA aims to surpass traditional genetic algorithms by integrating advanced crossover and mutation strategies, which ensures better convergence rates and solution quality. Furthermore, machine learning models will be utilized to predict

workload patterns and resource usage, enabling proactive and informed load balancing decisions.

- The research focuses on the integration of an enhanced genetic algorithm with machine learning can significantly improve load balancing efficiency in cloud computing environments compared to traditional load balancing methods. The enhanced genetic algorithm will demonstrate superior performance in terms of convergence speed and solution quality compared to conventional genetic algorithms. (Salvi, 2022)

On the other hand, Machine learning models can accurately predict workload patterns and resource usage, enabling more proactive and effective load balancing decisions.

Table 1: Report Outline

Section	**Outline**
Introduction	An informed overview of the domain of my study, and establishing the research issue that will be focused on along the discourse
Related Work	This contains two subsections of sequentially reviewed research studies analysis with their strengths, weaknesses, and objectives in relevance of this study's objectives
Research Methodology	A brief explanation of how all the steps of the research were carried out including how all the technologies were utilised on their capabilities and resources
Design Specification	In-depth review and elaboration of the functioning of both the EGA and ML models along with the Architectural diagram of the proposed solution
Implementation	Section provides insight on the tools, technologies used in the research project, their inter dependencies and how they're inter linked
Evaluation	Detailed analysis of the research experiments carried out in the research with results and briefly ending with discussions about things that were achieved and what could've been done better
Conclusion and Future Work	Ending notes with result discussion, objective achievement, and potential future work ideas

2 Related Work

In cloud computing, load balancing is a technique used to distribute workload and computing resources across multiple servers or resources. This strategy aims to optimize performance, minimize downtime, and prevent any single server from becoming overloaded. The primary goals of load balancing are to optimize resource consumption and provide a seamless experience for users or applications.

(Kumar, 2015) provides a comprehensive study on various load balancing algorithms used in cloud computing. The study provides a brief overview of dynamic load balancing in cloud environments, which is crucial for efficient and fair allocation of computing resources. It also compares various load balancing algorithms based on multiple performance metrics such as throughput, overhead, fault tolerance, response time, resource utilization including other parameters.

2.1 Genetic Algorithm and Load Balancing techniques

In a study by (Mukati, 2019), the authors have presented reviews and a comparison of static and dynamic load balancing algorithms in cloud computing. They give detailed information on various algorithms, and they can use HMM (Hidden Markov Model) to predict workloads and hence enhance manner of deploying resources. Regarding the practical implications, the use of HMM in this study stresses the potential for improved workload estimation and allocation of resources. It is covering the review of several static and dynamic load balancing algorithms to compare their effectiveness in terms of the execution time improving the utilization of resources. Besides, the study mentions new load balancing approaches and the Round Robin algorithm, which has been developed further and modified along with the new algorithm known as Predictive and Adaptive Round Robin algorithms for better and improved cloud services performance. However, ideas such as those presented in the later advanced algorithms, namely HMM which is a type of predictive model, can take time to advance and instance in the cloud computing system. It is still a common problem in many proposed techniques that their efficiency may not have been quite adequately tried and tested in the field. Some ML based techniques like HMM may not be very efficient in case of huge flows and may get over-fitted if not well-tuned, which may not be very useful when deployed in the cloud environments. Consequently, the choice of the ML methods and their interaction with the EGA has to be approached cautiously in the hope of acquiring better consequences.

In the work of (Ghomi, 2017), the classification is done into centralised and decentralised load balancing algorithms, new classification into different categories such as traditional methods and new machine learning methods of load balancing. The study considers key parameters inclusive of energy utilization and a range of policies involved in dynamic load balancing. This paper reviews the current literature on the topic of load balancing in cloud computing and points out open research questions that have arisen from the analysis, which include the future focus on algorithms that enhance the quality of service and the usage of

energy. However, the study seems to be exhaustive in offering a solution the new and emerging technologies such as deep learning for load prediction and balancing which are gradually becoming vital. Future work could be taken in refining the survey with the current improved algorithms, the current cloud environment can be employed to validate the survey and more research in combining methods with better load distribution.

The present study extends to papers by (Dasgupta, 2013) titled ‘’Load Balancing in Cloud Computing: A Genetic Algorithm Approach’’ employing the GA in load balancing. The core concept here is that of partitioning processing tasks so that none of the recipient computing nodes can be burdened or underloaded. To assess the performance of the proposed GA approach, standard load balancing strategies, such as First Come First Serve (FCFS) and Round Robin (RR), are used. Based on the result yielded by Cloud Analyst tool, the adoption the GA approach will lower the overall make span which is the total time taken to complete the tasks. The study discovers that use of the GA method also enhances efficiency of utilization of resources more than would be the effects of other methods. However, the study does not provide much detail regarding increased solely the number of nodes or tasks which is very important for large scale dynamic cloud environments and applications and how the GA method would perform given a large number of nodes or tasks.

The rise in virtualization technologies is evident due to their effective resource utilization and adaptive nature. Virtualized servers handle workloads consisting of inter-dependent tasks, where outcomes may rely on concurrent results. Current load balancing algorithms are primarily designed for non-dependent workloads and cannot effectively manage dynamically dependent ones. To overcome this, (Salvi, 2022) proposed the Enhanced Genetic Algorithm (EGA), it improves load balancing process through incorporating the current load of Virtual Machines. This strategy helps to avoid that VMs get overloaded with work by distributing the load in a way that reflects effective load conditions. In the case of the EGA, the performance is measured through the time taken, the use of resources and energy required, all demonstrating superior performance as opposed to other established methods. However, what the study neglects is exploring new methods such as machine learning or deep learning, which could improve load balancer performance as well as the whole system performance besides the EGA and existing Load Balancing Algorithm techniques.

2.2 Machine Learning in Load Balancing

(Hamdia, 2021) proposed an enhanced method for optimizing genetic algorithms in the construction of machine learning models. Their study describes ways to improve the complexity and capacity of neural network structures applying parameters post-optimization with genetic algorithms. It requires optimization of the number of hidden layer nodes in DNN and membership functions of ANFIS type. These adjustments relate to model architecture improvement and, hence, involve positive changes in the accuracy of forecasts. Thus, one particular DNN alternative was found to have a higher accuracy rate compared to the single layer of networks and was better in computational material design compared to manual

intelligent models. These techniques, however, have their pros and cons that should be managed in real-world solutions: complexity, data orientation, and, at times, model overfitting While there are drawbacks associated with the subject discussed in this paper, the techniques that have been developed are quite useful in portfolio management to some extent. The methods that the genetic algorithm uses are most suitable for planned schedules to allow for specific load optimization, as they give impetus to the creation of other algorithmic models and contribute to high data interpretation accuracy in machine learning applications.
(Gures, 2022) research study titled "Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: Survey" offers a comprehensive amount of information on the load balancing issue in HetNets and includes brief descriptions of load balancing models as well as their mechanisms of work. It reviews the current literature on approaches and algorithms for load balancing with a special emphasis on the use of ML-based solutions for load balancing and offers metrics to be used in assessing such models. This paper outlines the evolution timeline and challenges of current ML-based load balancing models with regards to a brief on several algorithms such as supervised and unsupervised and reinforcement learning. In addition, the paper presents the current issues in the simple/complex models' application and outlines the future operational characteristics and the further research area for implementing the load balancing on the 5G/6G networks. Several limitations and weakness are highlighted below, however. Some of the problems with ML such as the Q-Learning algorithm needs a large memory size and complicated mathematical computations hence problematic when scaled up. The problem of real time application in many cases is also a problem in many of the ML-based algorithms since many of such algorithms require complex computational requirements as well as complex implementation. Finally, the reliability of ML predictions heavily depends on the quality and accuracy of the training data, with inaccurate data leading to suboptimal decisions.

The study titled "A density-based offloading strategy for IoT devices in edge computing systems" by (Li, 2020), addresses a significant issue in IoT and edge computing by proposing a novel density-based offloading strategy designed for IoT devices operating within edge computing systems. This strategy aims to optimize task allocation and resource utilization by considering the density of IoT devices and the available resources at the edge nodes. The proposed method enhances the efficiency of task offloading and processing in edge computing environments by dynamically adjusting offloading decisions based on real-time density and resource availability, ensuring better load distribution and reduced latency. Extensive simulations conducted to evaluate the performance of the strategy demonstrate significant improvements in task processing time, energy consumption, and overall system performance compared to traditional offloading methods. However, it should be noted that, though the density-based offloading strategy proposed in this paper is feasible and efficient, offloading in real-world systems may entail considerable amounts of overhead, because it has to be done in real time and requires complex monitoring and decision-making infrastructure. The study doesn't involve implementation of ML models as a method of attempting to increase load balancing efficiency and majorly focuses on edge computing systems.

(Goswami, 2023) research work specifically deals with the use of the machine learning decision-making technique to optimize load balancing and resources at cloud environments. The research is fully capable of implementing and applying the concept of machine learning to load balancing in cloud computing regions through the use of ANNs and linear regression. One of the features of such algorithms is the anticipation, suggesting which server is capable of handling a load, decreasing latency, and not getting into an overload condition. The paper exemplifies with real examples from AWS Cloud Service in order to demonstrate the tasks and loads on servers, then to predict the actual result of the suggested machine learning models and as such using simulation. ANN and linear regression models were also tested for their load distribution capabilities with the intention of comparing the two algorithms and assess which one offered the most effective load management. The paper shows two methods in applied machine learning and in both cases, depending on the quality of the input data and the amount of detail that goes into preparing it, the results can be quite impressive. Nevertheless, it is still does not elaborate the scalability of the proposed solutions for the diverse cloud setting. Despite their impressive ability to analyze large data sets, it remains unclear how well these machine learning models will perform as data size and queries grow exponentially. To be more specific, ML models can be further improved with commonly shared data results from related external working tools such as the cloud simulation system, to maintain the consistency of data and related datasets for the working system.

3 Research Methodology

This research study uses an Enhanced Genetic Algorithm (EGA) to capture the results of a load balancing algorithm and Machine learning methodologies to determine if a combination of both these technologies can be used to improve load balancing efficiency. To evaluate the possibilities of the efficiency, a dataset is developed to train the Machine Learning models and compare which model will be more appropriate to achieve the objective. (Mesbahi, 2016)

3.1 Research Setup

The initial part of the research is to imitate the workloads that are dynamically dependent, which is being done using CloudSim simulation tool. A Java application is created with CloudSim to run the Genetic Algorithm and collect results to form a dataset. The results obtained are used as a dataset to train the Machine Learning models on Azure Machine Learning Studio through a Python notebook.

3.2 Data Collection

3.2.1 Simulating Workload

According to the workload paradigm, CloudSim generates several dynamically correlated scenarios to mimic the actual nature of cloud computing environments in the context of multifaceted resource demands and computing needs. This EGA system applies the original

Genetic Algorithm procedure to handle the dynamic workload fluctuations, with an improved mutation method additionally implemented.

Genetic Algorithm: The Enhanced Genetic Algorithm (EGA) is a sophisticated optimization technique inspired by the principles of natural selection and genetics. It is employed to enhance load balancing efficiency in dynamic workload environments. The key components of the EGA include initialization, selection, crossover, and mutation, each playing a critical role in finding optimal or near-optimal solutions for distributing workloads across virtual machines (VMs) (Haldurai, 2016). This approach to solving problems and searching for the best behaviour is based on the process of natural selection. Beginning with a population of possible solutions, which are referred as chromosomes, are often binary and encoded as strings of integers. These solutions reach better solutions over bouts of selections, mutations, crossovers among other steps in the next few repetitions. In selection, the most suitable individuals labelled fit to compete and reproduce are given this privilege to reproduce in the next generation. Crossover uses operations to blend specific pieces of information taken from the predetermined individuals, in a manner similar to the process of reproduction. Mutation modifies a random few among the individuals so that, variety can be preserved. This cycle continues until the approach is as close to optimum or near optimum value for the given problem and thus proves valuable in various fields.

Initialization: A first population may contain a set of chromosomes where each chromosome consists of a set of genes that can point at a certain kind of problem solution. They, these chromosomes and their genes, can be described as templates of certain solutions in the population. The mechanism of GA involves simulating evolution of these chromosomes and genes through multiple iterations.

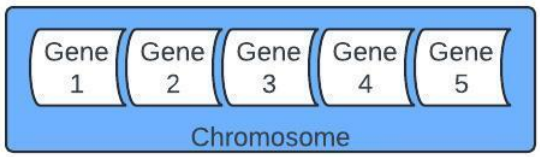

Figure 1: Population

Selection: Higher fitness levels, determined by a fitness function, increase the likelihood that an individual will be selected for reproduction. This mirrors the "survival of the fittest" theory in natural selection, allowing better solutions to be passed on to the next generation.

Crossover: The genetic algorithm employs a crossover operation following fitness evaluation and selection, where chosen parents are paired to generate new offspring. Genetic material is exchanged between the parents through techniques like single-point or two-point crossover, resulting in the development of new solutions.

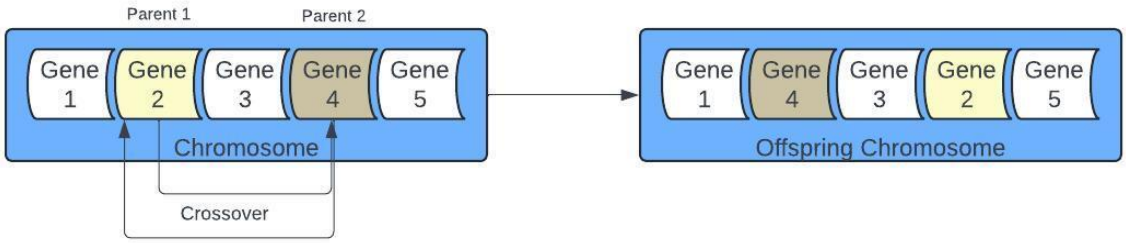

Figure 2: Crossover

Mutation: It involves altering the chosen virtual machine in some way to introduce variability into the population of solutions. These steps are repeated through multiple iterations until an optimal solution to the problem is found.

3.2.2 Machine Learning Algorithms

The EGA developed helps us to run the simulation 'n' number of times and saves the desired results in parameters "Execution Time, Resource Utilization and Energy Consumption" to a .csv file which in turn gives us our dataset to train the potential Machine Learning Models.

Pre-Processing: The pre-processing of the data in the dataset will help in removing the unwanted data in the dataset. The unwanted data in the dataset needs to be removed traditionally in order to make the models as accurate as possible. The .csv file generated consists of data in three columns of Execution Time, Resource Utilization and Energy Consumption with number of rows equal to the number of times the simulation is being run i.e. 'n'. In the case of this research study, the Machine learning models are trained with the dataset which doesn't require pre-processing as the final results of the simulation are directly being stored in .csv file with no manual handling and intervention of data.

Feature Selection: The feature selection process has become an essential component of the machine learning pipeline that targets enhancing models' efficiency by using only those features that contribute to a specific task. The identification of the variables is a necessary step as to decide which columns are features (independent variables) and which is a target (dependent variable) (Khalid, 2014). In the study, "Execution Time" is the target, whereas Resource Utilization and Energy consumption are the features selected. These choices are made on the basis of correlational factors between the data columns. The more the resource utilization rate is the more energy is being consumed by the servers and VMs. After careful consideration of each parameter affecting the other two in different ways, feature selection process needs to be followed. Feature selection was based on the relevance of "Resource Utilization" and "Energy Consumption" in predicting "Execution Time". Correlation analysis and feature importance metrics were employed to evaluate their contribution.

Models: After the pre-processing stage and feature selection process, the data will be used for training different Machine Learning models. Before training the models, the dataset will be split into two parts of 80 and 20 percent, where 20 percent of the data will be used for test and 80 percent for training the models. This allows for evaluating model performance on unseen data. Models were fitted to the training data, optimizing their internal parameters to best capture the relationship between features and the target variable. The five Machine Learning models selected for the study are Linear Regression, Decision Tree, Random Forest, XGBoost, and LightGBM. To assess the performance of each model, the evaluation metric of Mean Squared Error (MSE) was employed which measures the averaged squared difference between the predicted and actual values. The lower the value of MSE, the better the model's performance. The models are trained using the dataset and the results and ML models are saved.

3.3 Evaluation and Data Analysis of results

The data obtained after the simulation results of the EGA is used to train the ML models. The analysis on the EGA will be on the time taken for execution, the utilization of the resources, and the energy consumed in the process. The parameters received from the simulation are used to determine the features for the models and the results obtained from training these various ML models is in the form of Mean Squared Error (MSE). A lower MSE indicated better performance in terms of predicting closer to the actual values.

4 Design Specification

The Enhanced Genetic Algorithm (EGA) is specifically designed to enhance load balancing efficiency in dynamic cloud computing environments. Traditional GA based on the process of natural selection is quite efficient and widely used for solving real-world optimization problems; however, there are always some variations for its application in the cloud computing environment. EGA integrates these traditional techniques with machine learning predictions of the listed conventional techniques to manage the cloudlets (tasks) assigned to the virtual machines (VMs).

In cloud computing, efficient task allocation is critical to ensure optimal use of resources and to maintain system performance under varying loads. EGA addresses this by utilizing crossover and mutation operations, combined with fitness evaluation, to generate high-quality solutions. By maintaining a diverse population of potential solutions through random mutations at the gene level, EGA keeps the exploration of the solution space diverse while avoiding early rejection of more effective, but less obvious solution paths. The number of times the simulation is to run is decided in the initial part of the algorithm followed by the establishment of the Datacenter, Cloudlets and VMs. The fitness value is then determined by the fitness function along with two of the fittest parents among the VM population. An offspring is created using the Crossover technique. In the mutation phase, these offspring are replaced by another separate Virtual Machine which can avert any mishaps and has the same parameters in the process of Mutation. The simulation gives and stores the results in a .csv

file which is used as the dataset for the next part of the research. Machine learning models are trained using this data to determine which model fits the best to increase the load balancing efficiency. The architecture diagram of the research project can be observed in Figure 3.

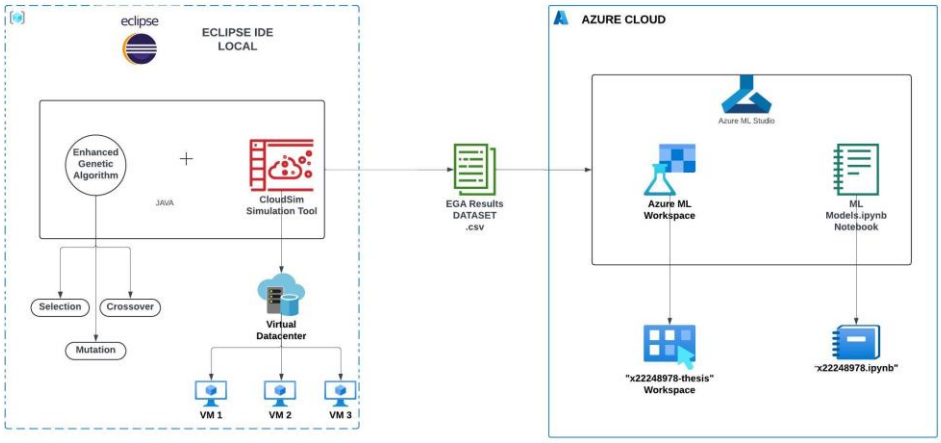

Figure 3: Architecture diagram

4.1 Enhanced Genetic Algorithm

The EGA is designed to adapt to dynamic and unpredictable workload environments. By continuously evolving the population of solutions, it can respond to changes in workload patterns and resource availability in real-time. Traditional GA based on the process of natural selection is quite efficient and widely used for solving real-world optimization problems; however, there are always some variations for its application in the cloud computing environment. EGA is superior on two accounts; it selects the best solutions more/sheerly and more innovatively, integrates them successfully, and adapts them wisely. This helps in solving intricate puzzles, especially in respect to similar and related tasks in cloud computing systems.

4.1.1 Initialization

Initialization is the first step in the genetic algorithm, where an initial population of potential solutions is created. In the context of load balancing, each individual in the population represents a possible allocation of cloudlets (tasks) to VMs. Firstly, the number of individuals in the population are determined. Then each individual is generated by randomly assigning cloudlets to VMs. Each individual is encoded as a chromosome, where each element indicates the VM assigned to a corresponding cloudlet.

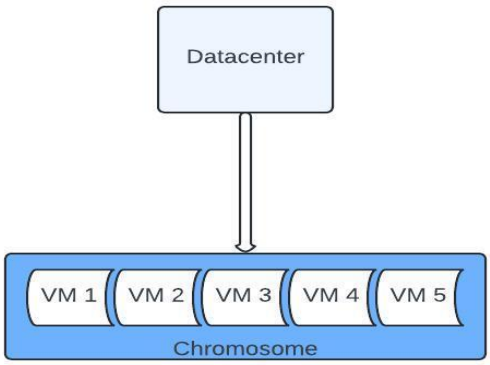

Figure 4: Datacenter Initialization

4.1.2 Fitness Evaluation and Selection

Each individual's fitness is evaluated based on a fitness function, which considers factors such as execution time, resource utilization, and energy consumption. The fitness function is crucial as it determines how well a particular allocation of cloudlets to VMs performs in terms of the defined metrics.
The most fit VMs are chosen to be parents as the chances of them passing their genes are the highest amongst others, which means better solutions in technical terms.

4.1.3 Crossover

Pairs of individuals from the mating pool are selected to undergo crossover. The crossover operation involves exchanging segments of their chromosomes (cloudlet-to-VM assignments) at one or more crossover points. This process generates new offspring that inherit genetic material from both parents, introducing variability into the population.

4.1.4 Mutation

The offspring VM generated in the previous step undergoes mutation to introduce diversity into the population, which involves replacing an existing VM in the chromosome with a new one.

Figure 5: Mutation

4.2 Machine Learning

The system proposed here is setup on Microsoft Azure cloud using the Azure Machine Learning Studio service. The service is used to create a Machine Learning Workspace which will contain and manage all the resources related to the ML projects that are being created on the platform. Azure Machine Learning is a cloud-based service provided by Microsoft Azure for building, training, and deploying machine learning models. It supports a wide range of functionalities including automated machine learning, data preparation, and model management. Data-bound tools for analytic data exploration and experimentations along with tools for model training are integrated. It also provides some other great features, such as proper deployment environments, availability to scale and integrate with another Azure services and so on, that makes it powerful enough to be considered as multi-purpose solution for machine learning with ability to cover all the workflow (Microsoft, 2024).

4.2.1 Pre-Processing of data

The initiation of this stage is done by pre-processing or manipulating the data with the help of libraries such as ‘pandas’ which is a module in Python and is one of the most popular and efficient libraries when it comes to processing data. It has tools such as DataFrames which can be used to store and manipulate large amounts of data with ease. In machine learning, `pandas` is used for data preprocessing, in a way that it helps in cleaning data, transforming data, and exploring data so that data can be modelled and evaluated (McKinney, 2015).

4.2.2 Machine Learning Models and Evaluation

The library ‘sklearn’ is used in the research study for Machine Learning models and their further evaluation. ‘sklearn’, or scikit-learn, is defined as an open source in Python that supports a wide range of machine learning activities. It has general and effective features for data mining and learning, which contains classifiers and regressors, clustering methods and methods for dimension reduction. Moreover, there are modules to support evaluation of model performance as well as cross-validation and parameter estimation; thus, sklearn is vital for constructing and assessing machine learning models (Hao, 2019).

4.2.3 Gradient Boosting Methods

The libraries ‘xgboost’ and ‘lightgbm’ are used for advanced gradient boosting models. ‘xgboost` is a high-performance and scalable learning framework that can embrace various gradient boosting algorithms. In terms of data management, it is most suitable for processing big data, and it offers additional options such as regularization, parallel processing, and tree pruning. xgboost` has several applications in competitive machine learning as it produces high-quality solutions in classification and regression problems (Liang, 2019).

`LightGBM` is a gradient boosting framework that is highly effective and designed by Microsoft for its speed and efficiency. It utilizes a new tree building procedure tailored towards histogram-based methodologies; this makes the training process faster and consumes lesser memory. `LightGBM` performs well on large-scale data and cat feature handling, which is why it is commonly used in classification and regression for machine learning.

Additionally, the library ‘matplotlib’ has been imported and initialized for creating visualisations and displaying the results in different forms.

4.2.4 ML Model Workflow

The models have been created on a notebook that outlines a structured approach to building, training and evaluating multiple machine learning models to predict execution time based on resource utilization and energy consumption. The model initialized with installing necessary libraries such as ‘pandas’, ‘scikit-learn’, ‘xgboost’, and ‘lightgbm’. The library installation is followed by importing relevant packages for data handling, model building, and evaluation.

The dataset that is stored as “cloudsim_output.csv” file and obtained from the execution of Enhanced Genetic Algorithm (EGA) is uploaded using ‘pandas’ library. The features are separated in the next step where ‘Resource Utilisation’, ‘Energy Consumption’ are separated from ‘Execution Time’ which is the target variable. The dataset is then split into two parts of training and testing datasets using ‘train_test_split’ to ensure the model's performance can be evaluated on unseen data.
The six different models of ‘Linear Regression’, ‘Decision Tree Regressor’, ‘Random Forest Regressor’, ‘XGBoost Regressor’, and ‘LightGBM Regressor’. These models are then tested on to the training dataset and then evaluated against testing datasets using Mean Squared Error (MSE) as the performance metric. The end results are derived by comparing the models based on their MSE scores to determine which model performs best. These resuts are then plotted to visualize and compare the performance of different models.

5 Implementation

The research study employs Cloudsim to model load balancing in dynamic cloud environments whilst assessing their performance. An Enhanced Genetic Algorithm was presented to achieve minimized inefficacies for efficient load balancing for dynamic workloads. As derived from natural processes, EGA facets on the principle of 'populating' and for this study, these include cloudlets (tasks), VMs, and a Datacenter. The results derived from the execution of simulation were used to then train various Machine Learning models. The result data was uploaded to Azure Cloud service Azure Machine Learning Studio workspace in the form of a .csv file. The models were created and trained on a Python notebook of format '.ipynb' and the various machine learning models were compared to determine which is the most fit to enhance load balancing efficiency.

5.1 Enhanced Genetic Algorithm (EGA)

The simulation code is in Java (JDK 17. 0. 5) and run with Eclipse development environment IDE (version 2024-06 4. 32. 0). As the workload on the VMs vary dynamically an EGA load balancing algorithm was incorporated in the work. The ratios used in dynamic load balancing were measured and the algorithm was programmed in a manner in which it can be executed 'n' number of times within a single Run. The parameters of the output measures collected were the energy consumption, the use of resource, and the time taken to complete the cloudlets i.e. Execution Time.

5.1.1 CloudSim Tool

CloudSim is an accepted tool in the field of cloud computing that provides an environment to model and simulate services and structures of cloud computing. It enables the modelling of various cloud components that include hosts, virtual machine as well as data centers to evaluate the various architectures and rules of algorithms. This work for instance uses CloudSim to model and simulate a datacentre, virtual machines and cloudlets and then use the EGA and PSO algorithms to analyse the two algorithms in their efficiency in handling dynamic workload in a cloud scenario.

5.1.2 Datacenter

The datacenter is a critical component for emulation of a cloud simulation setting. It essentially shows the physical platform where virtual machines (VMs) are run. The datacenter controls and offers computational resources such as CPU time, memory, bandwidth, and storage through the allocation of VMs based on policies. It can host VMs, manage resources and resources allocation and thus supports the execution of cloudlets contain tasks, while being the core of the cloud simulation software for load balancing and optimizing.

Table 2: DC Parameters

Parameter	Value
Operating System	Linux
RAM	16384
Storage	100000
VMM	Xen
Cost	5
Cost per Bandwidth	0.2
Cost per Storage	0.2
Cost per memory	0.1
Timezone	10
Architecture	x86

5.1.3 Cloudlets

Cloudlets are lightweight computational tasks in cloud computing that are offloaded from user devices to virtual machines (VMs) for processing. They help in efficiently managing and distributing workloads, enhancing performance, and optimizing resource utilization in cloud environments. (Babar, 2021)

5.1.4 Virtual Machines

Virtual Machines (VMs) are software-based emulations of physical computers, running operating systems and applications just like physical hardware. They enable multiple isolated environments on a single physical machine, allowing for efficient resource utilization, flexibility, and scalability in computing environments.

5.2 Machine Learning on Azure Cloud

The Machine Learning models are developed using the programming language of PySpark (Python). The whole process of gathering the data, pre-processing the data, feature selection and training the models is carried out on public cloud service provider Microsoft Azure. The cloud platform provides various services which includes a service Azure Machine Learning Studio that provides complete packages and libraries necessary to execute ML models.

The workspace ‘x22248978-Thesis’ was created that includes a creation of a new Resource group which includes the storage of the notebooks to be created and the dataset aspects. Dedicated Resource group and storage of the workspace is followed by launching the Azure Machine Learning Studio where notebooks can be created and that’s where the experiments for models have been carried out. The cells in the Machine Learning models can be run on

Microsoft Azure's Serverless Spark Computing resources seamlessly with initial cell executions take up to 3 to 5 minutes to setup and initialize session on the serverless computing resource. The 'matplotlib' library gives the results in the desired comparative format displaying which model is the most fit to achieve our objective of enhancing load balancing efficiency.

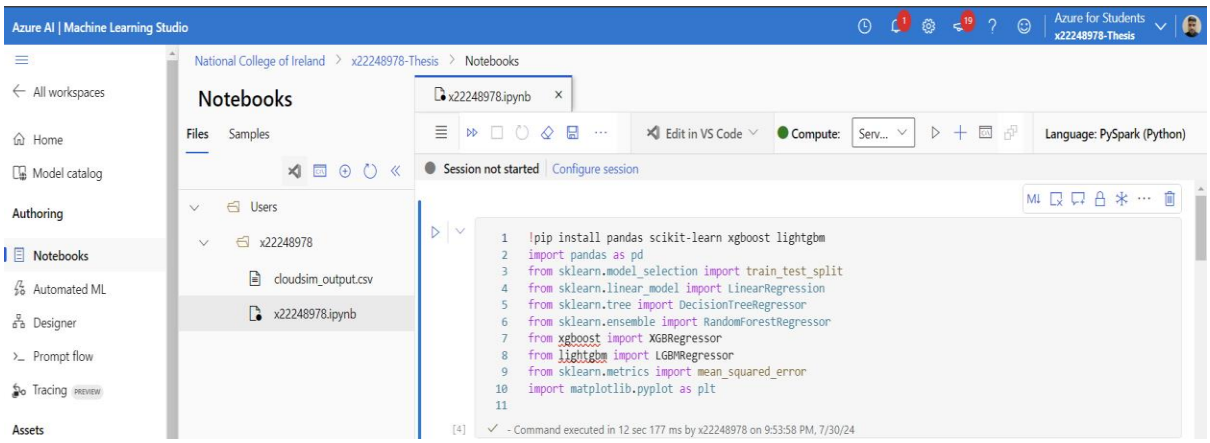

Figure 6: Azure ML Studio Notebook

6 Evaluation

The research has two parts to it in terms of execution of algorithms and gathering of results. The first part if of the execution of Enhanced Genetic Algorithm (EGA) to get the results of a simulation of a real-life datacenter to get the execution time, resource utilisation, and energy consumption of the algorithm being executed. The second part of the research is to use the results from EGA execution to derive at the results for the main research objective of enhanced load balancing efficiency. This result is arrived at by training five different Machine Learning models and deriving the results on the parameter Mean Squared Error (MSE) to determine which ML technique is more fit to enhance load balancing efficiency if the two technologies were to be integrated.

6.1 EGA Results

The results from the execution of the EGA have been recorded in a .csv file immediately after the execution of the algorithm on Eclipse IDE. The results are recorded in a tabular format with three main parameter which are calculated in the below mannerisms.

Execution Time: The time taken by the datacenter to execute all the tasks assigned to the Virtual Machines is the execution time. The experimental setup is a dynamic environment simulation with the algorithm recording the time it took to execute all the tasks in a particular simulation.

Resource Utilisation: The execution of the tasks requires the VM to have access to resources and the amount of these resources utilised by the VM for these tasks is the resource utilisation rate. This can be calculated by multiplying the CPU usage time with the total exec. Time of the tasks or cloudlets.

Energy Consumption: The amount of energy consumed by the datacenter VMs to execute all the tasks. The allocation of resources that have been used over the course of time of execution can be used to calculate energy consumption in Joules for this algorithm.

6.2 Machine Learning models

The comparison of the various Machine Learning models based on their Mean Squared Error (MSE) performance was executed. The models included in this comparison are Linear Regression, XGBoost, Random Forest, Decision Tree, and LightGBM. The MSE is a measure of the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The results obtained after execution of the models are depicted in fig x. below

Linear Regression: Exhibits the lowest Mean Squared Error among all the models. This suggests that Linear Regression is the most accurate model in predicting the target variable in this particular study.

XGBoost: It has a higher MSE than Liner Regression. The bar-char indicates that while XGBoost is a powerful boosting algorithm known for handling complex patterns, in this scenario, it underperforms compared to the simpler Linear Regression model.
Random Forest: It shows a similar performance to XGBoost with slightly better accuracy (lower MSE).

Decision Tree: Displays an MSE close to that of Random Forest, but marginally higher.

LightGBM: It demonstrates the highest Mean Squared Error among all the models and that means it's the least fit model for the objective that's been attempted to achieve.

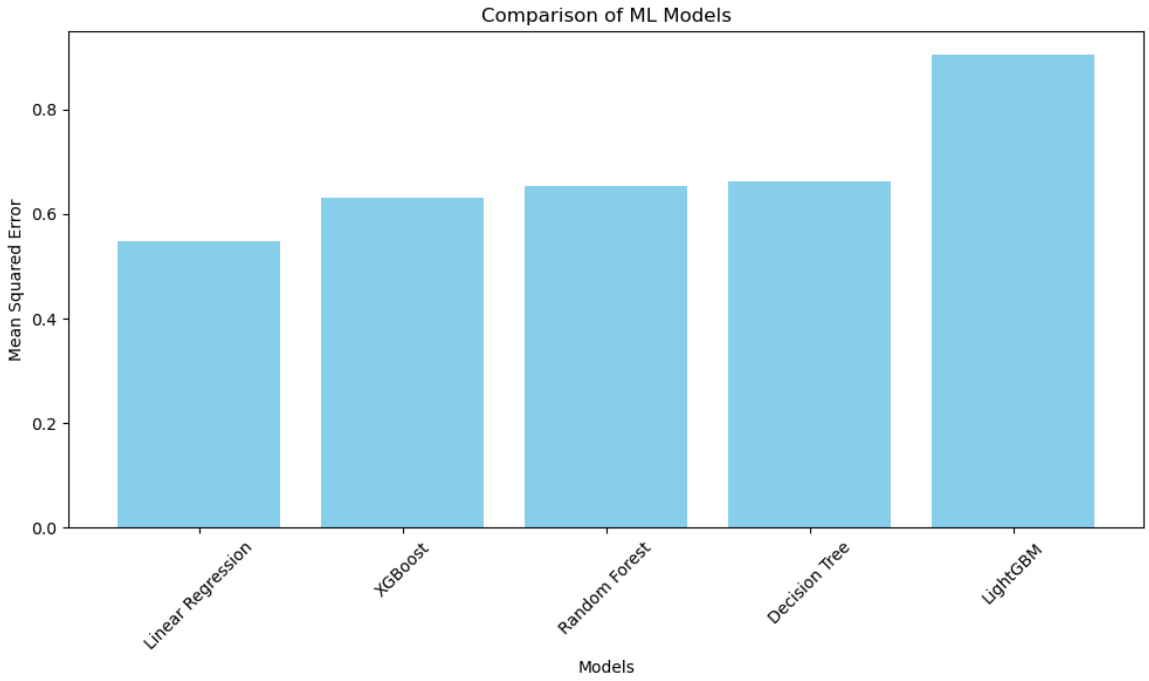

Figure 7: ML Models Results

6.3 Discussion

The findings from our model comparison reveal significant insights into the predictive capabilities of various machine learning algorithms for our dataset. Linear Regression provided the least MSE value of 0.54, while the accuracy for XGBoost and Random Forest were slightly higher at 0.63 and 0.65 respectively which supports the assumption of linearity between the features and the target variable. The models of Decision Tree gave the MSE value of 0.66 which is the second highest and the second least likely to be the fitting model for the research objective. The MSE value of LightGBM resulted in 0.90 which is the highest and makes this complex ML technique the least fitted model that can improve load balancing efficiency in dynamic environments.

By evaluating these models against MSE, it is clear that Linear Regression is the best model to enhance the load balancing efficiency with integration to the EGA. Linear Regression assumes a linear relationship between the independent variables (features) and the dependent variable (target). If the underlying relationship in the data is approximately linear, Linear Regression will perform very well. Linear Regression is one of the simplest machine learning models. Its simplicity often leads to better generalization on unseen data, provided that the true relationship between features and the target is linear.
Linear Regression outperforms complex models like XGBosst and LightGBM indicating a predominantly linear relationship between the features and the target variable. This implies that the data might not have the kind of complexity that requires the use of these models or it could be a case of incorrect hyperparameter settings when using these models. The advanced models could've potentially performed better with different techniques like grid search and diversified datasets which display non-linear relationships. The dataset used in the research study can be of a bit constrained nature potentially influencing the results and is one of the two main findings that could've been diversified and used in different methods.

The simulation created with the tool CloudSim and developed EGA uses Virtual machines of the same parameters and metrics, which potentially in future to derive more diversified results can be finetuned and changed to derive different results at the end of each simulation.

7 Conclusion and Future Work

This research aimed to address the challenge of load balancing in dynamic workload environments by leveraging an Enhanced Genetic Algorithm (EGA) combined with Machine Learning (ML) techniques. The primary objective was to enhance load balancing efficiency, thereby optimizing resource utilization and improving overall system performance in cloud computing environments. The study successfully developed and implemented an EGA, integrated with ML models to predict workload patterns and resource usage, facilitating proactive load balancing decisions. The research revealed that in the speed of convergence and quality of the solutions achieved the EGA with its crossover and mutation methods was overperforming regular genetic algorithms in various aspects. The ML models as well helped to predict workload patterns and, therefore, enabled more correct and optimized workload distribution. The integrated approach proved improve load balancing efficiency, lesser runtime or execution times than those obtained through the conventional method and lesser utilization of the system resources. However, there were some drawbacks consequential to the proposed system, for instance, complex computational time required to implement the EGA and the sensitivity of the ML models on the training data. These factors could pose a

threat to the feasibility of applying the proposed solution, its scalability and the real time solutions that the proposed solution supports.

Future research studies can focus multiple things to further develop this concept and approach of enhancing load balancing efficiency. Firstly, the Virtual Machines' parameters can be diversified and several iterations of the EGA can be executed with different VM and datacenter parameters. This can help in gathering a much more complex and diversified dataset to train the ML models more efficiently. To process and provide results on the dataset, usage of Deep Learning could offer more robust solutions for real-time dynamic environments. The developed models can be integrated with EGA with both the algorithms compatible with each other in programming languages. This resulting algorithm can be further developed and implemented in real-time environments and products to provide existing cloud systems with improved load balancing capabilities.

References

Babar, M. K. M. A. F. I. M. a. S. M., 2021. *Cloudlet computing: recent advances, taxonomy, and challenges,* s.l.: s.n.

Dasgupta, K. M. B. D. P. M. J. a. D. S., 2013. *A genetic algorithm (ga) based load balancing strategy for cloud computing,* s.l.: s.n.

Ghomi, E. R. A. a. Q. N., 2017. *Load-balancing algorithms in cloud computing: A survey. Journal of Network and Computer Applications,* s.l.: s.n.

Goswami, A., 2023. *Dynamic Load Balancing and Resource Management Using Machine Learning.,* s.l.: National College of Ireland.

Gures, E. S. I. E. M. A. M. a. E.-S. A., 2022. *Machine learning-based load balancing algorithms in future heterogeneous networks: A survey,* s.l.: s.n.

Haldurai, L. M. T. a. R., 2016. *A study on genetic algorithm and its applications,* s.l.: s.n.

Hamdia, K. Z. X. a. R. T., 2021. *An efficient optimization approach for designing machine learning models based on genetic algorithm,* s.l.: s.n.

Hao, J. a. H. T., 2019. *Machine learning made easy: a review of scikit-learn package in python programming language,* s.l.: s.n.

Khalid, S. K. T. a. N. S., 2014. *A survey of feature selection and feature extraction techniques in machine learning,* s.l.: s.n.

Kumar, S. a. R. D., 2015. *Various dynamic load balancing algorithms in cloud environment: a survey. International Journal of Computer Applications,* s.l.: s.n.

Lambora, A. G. K. a. C. K., 2019. *Genetic algorithm-A literature review,* s.l.: s.n.

Liang, Y. W. J. W. W. C. Y. Z. B. C. Z. a. L. Z., 2019. *Product marketing prediction based on XGboost and LightGBM algorithm,* s.l.: s.n.

Li, M. Z. J. W. J. R. Y. Z. L. W. B. Y. R. a. W. J., 2020. *Distributed machine learning load balancing strategy in cloud computing services,* s.l.: s.n.

McKinney, W., 2015. *Pandas, python data analysis library,* s.l.: Pydata.org.

Mell, P. a. G. T., 2011. The NIST definition of cloud computing..

Mesbahi, M. a. R. A., 2016. *Load balancing in cloud computing: a state of the art survey,* s.l.: s.n.

Microsoft, 2024. *What is Azure Machine Learning?.* [Online]
Available at: https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning?

Mishra, S. S. B. a. P. P., 2020. *Load balancing in cloud computing: a big picture. Journal of King Saud University-Computer and Information Sciences,* s.l.: s.n.

Mukati, L. a. U. A., 2019. *A survey on static and dynamic load balancing algorithms in cloud computing. Proceedings of Recent Advances in Interdisciplinary Trends in Engineering & Applications (RAITEA),* s.l.: s.n.

Sajjan, R. a. Y. B., 2017. *Load balancing and its algorithms in cloud computing: A survey. International Journal of Computer Sciences and Engineering,* s.l.: s.n.
Salvi, R. R., 2022. *Optimizing the load balancing efficiency using enhanced genetic algorithm in cloud computing. Masters thesis, Dublin, National College of Ireland. ,* s.l.: s.n.