




Article

A Resource Utilization Prediction Model for Cloud Data Centers Using Evolutionary Algorithms and Machine Learning Techniques

Sania Malik ¹, Muhammad Tahir ¹, Muhammad Sardaraz ^{1,*} and Abdullah Alourani ²

¹ Department of Computer Science, Attock Campus, COMSATS University Islamabad, Attock 43600, Pakistan; fa19-rs-015@cuiatk.edu.pk (S.M.); m_tahir@cuiatk.edu.pk (M.T.)

² Department of Computer Science and Information, College of Science in Zulfi, Majmaah University, Al-Majmaah 11952, Saudi Arabia; a.alourani@mu.edu.sa

* Correspondence: sardaraz@cuiatk.edu.pk

Abstract: Cloud computing has revolutionized the modes of computing. With huge success and diverse benefits, the paradigm faces several challenges as well. Power consumption, dynamic resource scaling, and over- and under-provisioning issues are challenges for the cloud computing paradigm. The research has been carried out in cloud computing for resource utilization prediction to overcome over- and under-provisioning issues. Over-provisioning of resources consumes more energy and leads to high costs. However, under-provisioning induces Service Level Agreement (SLA) violation and Quality of Service (QoS) degradation. Most of the existing mechanisms focus on single resource utilization prediction, such as memory, CPU, storage, network, or servers allocated to cloud applications but overlook the correlation among resources. This research focuses on multi-resource utilization prediction using Functional Link Neural Network (FLNN) with hybrid Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The proposed technique is evaluated on Google cluster traces data. Experimental results show that the proposed model yields better accuracy as compared to traditional techniques.

Keywords: cloud computing; resource utilization; forecasting; neural networks; GA; PSO



Citation: Malik, S.; Tahir, M.; Sardaraz, M.; Alourani, A. A Resource Utilization Prediction Model for Cloud Data Centers Using Evolutionary Algorithms and Machine Learning Techniques. *Appl. Sci.* **2022**, *12*, 2160. <https://doi.org/10.3390/app12042160>

Academic Editors: Eui-Nam Huh and Valentino Santucci

Received: 23 November 2021

Accepted: 26 January 2022

Published: 18 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cloud computing provides customizable platform that allows applications to acquire appropriate resources before execution of required applications on Virtual Machines (VMs) [1]. Cloud service providers typically use a pay-as-you-go pricing model that can lead to cost reduction and flexibility for cloud users. The broad range of developments in cloud computing technology has resulted in significant increase in cloud users and the development of applications in cloud environments to access different services of cloud computing [2]. Several scientific applications use services of cloud computing that result in varied utilization of cloud resources [3,4]. Consequently, it is necessary to manage resources efficiently to handle the fluctuating demand of users. Efficient resources management in cloud computing environment can contribute to optimizing the usage of resources, reducing cost, and improving performance. To achieve efficient resource management, resource utilization prediction is used.

The use of machine learning techniques in research has received more attention in recent years. Neural networks are one of the most widely used methods in machine learning. The key advantage of using neural networks is the ability of accurate decisions. Many applications, including electric flow forecasting, river flow forecasting, etc., use neural networks [5]. Keeping these reasons in view, this article uses neural networks for cloud resource utilization prediction. The key issue in implementing neural networks is training

the network weights. Training the weights of the network is a complex optimization problem. Swarm and evolutionary algorithms are extensively used for such problems. The use of these techniques is preferred over the traditional mathematical methods [5]. To achieve this goal the proposed model uses hybrid Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) (GA-PSO) that takes the advantage of the strengths of both algorithms. Hybrid versions of these algorithms have been used in a variety of domains and impressive results are achieved. The applications include antenna array pattern synthesis [6], mining association rules [7], forecasting electricity demand [8], scheduling resources in cloud computing [9], and process planning [10]. Authors in [11] have proved with the simulation that the hybrid version overcomes the disadvantages of the individual algorithms.

Resource utilization prediction is studied extensively and rich literature is available [1,4,12–14]. Resource prediction models are influenced by some parameters, such as accuracy, time and memory complexities of the model, handling multiple resources, etc. Accuracy of the resource utilization prediction is a challenge due to multiple types of metrics, e.g., CPU and memory usage, disk I/O, network throughput, etc. There may be some implicit relationships, such as the relationship between CPU and memory usage, and the relationship between disk I/O and memory. It is difficult to find and predict the relationship for each type of resource. In this way, the prediction results will not be suitable for practical use. To solve this problem, the auto-scaler based on cloud resource prediction must have the ability to handle different indicators at the same time to make precise scaling decisions.

In this article, we propose a new model for cloud resource utilization prediction using the hybrid GA-PSO algorithm. The prediction method handles multiple resource indicators simultaneously. This method combines the advantages of the two algorithms and proposes a hybrid GA-PSO algorithm to solve the prediction problem. First, the hybrid model is used to train the neural network, and then the neural network is used for resource utilization prediction. The research contribution of the article can be summarized as follows. A state-of-the-art literature review is presented that covers the analysis of different methods for resource utilization prediction in cloud computing. A hybrid model is proposed that uses GA-PSO to train the network to improve prediction accuracy. Comparative analysis of the proposed model with traditional models is presented that leads to future research directions for resource utilization prediction. The article consists of the following sections. Related work is discussed in Section 2 with a brief description of the recent literature related to the proposed work. The proposed model is presented in Section 3 with details including figures, algorithms, equations, and relevant discussion. Results and discussion are presented in Section 4, followed by a conclusion in Section 5.

2. Literature Review

Multiple services, such as processing, storage, etc., can be accessed over the Internet using a cloud computing paradigm. The Internet is becoming faster and it is easier to access services of cloud computing. Effective resource management in cloud-based computing can bring improvement in the usage of resources, applications performance, and reduce usage costs. There exists strong literature on cloud resource utilization prediction techniques. This section presents details of the related methods. Researchers in [1] proposed a regression integration method for intelligent resource utilization prediction. The proposed method integrates resource usage and function selection to improve utilization and performance. The results show that the accuracy and execution time of the proposed model are improved as compared to existing models. In addition to the better prediction, the technique reduces failures and results in fault-tolerant scheduling. The scalability of virtualization technology for cloud consumers means an excessive demand or fewer resources with respect to time [4]. Depending on this, the efficient utilization of cloud services becomes even more challenging. The model for cloud resource utilization is time-dependent and influenced by cloud resource usage trends. The authors proposed a Learning Automata (LA) theory-based cloud resource usage prediction algorithm. The algorithm uses prediction models

to calculate weights for individual models [4]. The proposed algorithm is validated with prediction of load of many VMs. The results show that the proposed method performs better in comparison to other prediction algorithms. Determination of the exact number of resources for cloud computing applications is a complex task. The methods for workload prediction are based on a single model. Researchers in [13] proposed an adaptive method for workload prediction. This method first classifies workloads into various classes and automatically assigns different prediction models based on the characteristics of the workload. Experiments are performed to validate the proposed model. Automated resource provisioning enables flexible services by adjusting available resources to meet service needs. The accurate prediction plays significant role in reducing the power consumption and ensuring QoS and SLA especially for those services that have rigorous QoS requirements in terms of latency or response time. Authors in [14] proposed a new mechanism designed to precisely predict the processing load of distributed servers and estimate the suitable number of resources. The proposed algorithm targets optimization of service response time, SLAs, while reducing over-provisioning of resources to reduce energy consumption and costs. Experimental results show that the proposed model has better accuracy than the other compared models. Host load prediction is important for the improvement of resource allocation and utilization in cloud computing environments. Since the amount of global variance is more than the variance in the grid, its prediction with a certain amount of accuracy remains a challenge in cloud-based systems. In reference [15], the authors applied a concise adaptive and powerful model called Long Short Term Memory (LSTM) that can predict the average load in advance in consecutive future intervals. Two real workloads are used for the evaluation of performance. The experimental results show that the proposed algorithm is more accurate on both datasets as compared to other models. Despite many benefits of cloud computing, the paradigm also faces some challenges. The key issues of cloud computing consist of dynamic resource expansion and power consumption. Inefficient resource provisioning leads to inefficient cloud systems and expensive workload forecasting. A workload prediction model using neural networks and an adaptive differential evolution algorithm is proposed in [16]. The administrators can find the potential problems in the resource reservation plan and change the plan accordingly. The proposed algorithm makes the decision based on over-provisioning or under-provisioning of the resources. The extracted knowledge during the process is used to analyze the characteristics of resource utilization. Experimental result shows that the proposed prediction of the model is accurate than the compared methods. One of the key functions of cloud computing is scalability, achieved through appropriate resources scheduling. An important question is whether a resource reservation plan can be defined and used for resource scheduling effectively. The plan can allocate new resources while reserving enough available resources. A resource utilization prediction technique based on neural networks and self-adaptive differential evolution technique is proposed [17]. The proposed technique is capable of selecting the best suitable crossover and mutations. The proposed algorithm is evaluated on benchmark datasets and comparative results are presented. Cloud computing faces a few challenges, including dynamic resource expansion and power consumption. Such transactions make cloud systems fragile and expensive. Researchers in [18] proposed an algorithm to solve the problem of workload prediction in cloud data centers. The workload prediction model was developed using LSTM network. The experimental results show that the proposed method achieves better prediction accuracy as compared to other compared models. In addition, to develop models based on different attributes of the workload, the majority of clouds use only user-defined resource usage thresholds to provide automatic scaling capabilities. Few jobs can process multiple indicators simultaneously to predict resource forecasts. Authors in [19] proposed a new cloud-active automatic scaling system prediction model that combines multiple mechanisms. Correlation between different metrics is evaluated to select suitable inputs. A fuzzification technique is proposed to reduce the fluctuation of monitoring data. The prediction model is based on LSTM. Google trace data are used to evaluate the proposed model. Existing cloud resource scheduling meth-

ods mainly focus to bring improvement in resource utilization and power consumption by enhancing traditional heuristic algorithms. The occurrence of peak loads may cause scheduling errors, that may affect the energy efficiency of the scheduling algorithm. The peak loads can produce scheduling errors because no predictive model is able to predict the forthcoming resource usage of the data center by observing the data assembled by the model at the first stage. The effective scheduling algorithms offer a perfect solution to very complex problems while supplying the QoS as well as preventing SLA violations. To solve these problems an algorithm is presented in [20]. The proposed scheduling mechanism is responsible for resource scheduling while minimizing the consumption of energy and guaranteeing QoS. An algorithm predicting the completion time is proposed in [21]. MapReduce packages in cloud computing can personalize the resources allocation and finish the MapReduce jobs in a limited amount of time. Modern systems need cloud users to estimate the number of resources required to run tasks in the cloud. The proposed framework uses scale-out strategy to obtain accurate prediction of the completion time of the jobs. The regression-based performance model can predict and evaluate the execution time of five well-known MapReduce benchmark applications in a private cloud environment. To capture the variability in cloud workloads, an algorithm based on gradient descent and Levenberg–Marquardt (LM) adaptation techniques is proposed in [22]. The authors proposed a sparse framework to adapt the online resource utilization prediction model. The framework uses the concept of sparse networks. Different models are used to introduce sparsity. To achieve the desired level of sparsity, the proposed algorithm eliminates and retrains different parameters. The proposed framework is capable of fast online adaptation of resource utilization prediction models. The algorithm is validated on benchmark datasets in terms of accuracy. To address the issues of excessive power consumption, imbalanced load, and inefficient resource utilization, an efficient resource management algorithm is presented in [23]. The algorithm targets better resource utilization and load management to improve performance. The algorithm uses online resource prediction for each VM to reduce SLA violations and performance degradation. To achieve the desired goals VM placement and migration policies are used. Experiments on different workloads are performed to validate the performance of the proposed algorithm. Another algorithm [24] targets multi-variate resources utilization prediction in cloud data centers. The resources include CPU, memory, and network bandwidth. The algorithm uses a Convolutional Neural Network (CNN) and LSTM models for resource utilization prediction. Initially, the vector autoregression method is used to filter the linear interdependencies between the multi-variate data. In the next step, CNN and LSTM are used for prediction. The proposed model is evaluated with experimental results and comparative results in terms of accuracy are presented. VMs consolidation in cloud environments leads to resource wastage due to instability and high variability of resource utilization. To solve the issue, authors in [25] proposed a resource utilization prediction algorithm using support vector regression technique. The proposed method is best suitable for multi-attributes resources, particularly non-linear workloads. The algorithm is validated with experiments on real workloads and comparative results are presented. Despite many solutions in literature, resource utilization prediction of multi-variate resources still needs sophisticated solutions with improved accuracy and less execution time.

3. Models and Methods

The proposed algorithm is based on hybrid GA-PSO model [9]. The hybrid model integrates single predictive models, which leads to overcome the limitations of the predictive model in terms of cost and complexity of the final model. The main goal is to select and appropriately combine a set of prediction techniques to improve the accuracy of the final forecast. By combining the advantages of each predictive model and minimizing its disadvantages, accuracy can be improved. This section presents details of the different components of the proposed algorithm.

3.1. Neural Networks

Neural networks inspired by biological brains are one of the most versatile and prominent machine learning approaches. Neural networks have been widely applied in various domains including regression, clustering, classification, prediction, pattern recognition, learning, and robotics [5]. Standard feedforward neural network consists of the input layer, one or more hidden layers, and an output layer. In principle, a neuron is a strategic signal processing unit that reads numerous signals as input and then uses the activation function (such as sigmoid) to generate a signal as an output. A sensory route for a network to detect the environment is formed by reading the normalized signal into the input layer. For instance, in the CPU utilization prediction problem, this signal refers to the host CPU demand from previous time steps. This signal is first sent forward via weighted connections through the interconnected layers of neurons. As soon as the signal is propagated through the network, the network then generates an output signal through the final output layer. The output signal correlates to the CPU demand in the future time steps. Several examples in the literature show the effective use of neural networks to solve problems like CPU utilization prediction. It is noteworthy to mention that one of the most challenging tasks in the implementation of neural networks is training the network weights. Training network weights, such that the network could generate a highly accurate output for any given input is a complicated optimization problem, and, therefore, involves the potential use of swarm and optimization algorithms. Functional Link Neural Network (FLNN) is a class of neural networks that utilize higher combinations of inputs. Many applications used FLNN including classification and prediction [26]. The neuron input signal can be expressed as shown in Equation (1).

$$v_j = \sum_{i=1}^n w_{i,j} a_i \quad (1)$$

where i th layer is preceding to j th layer containing N neurons, v_j denotes the input at j th layer, a_i is the output in the i th layer, $w_{i,j}$ represents the weights assigned to each output signal when passed to every neuron in the layer. The output value of every neuron always lies in the range between $[0, 1]$, and it is calculated by using the activation function. The activation function (such as sigmoid) is expressed as shown in Equation (2).

$$a_i = \frac{1}{1 + \exp(-v_i)} \quad (2)$$

3.2. Genetic Algorithm

GA is the optimization algorithm typically used in complex and large systems, for the determination of values close to the global optimal value. Therefore, GA is suitable algorithm for training neural networks. A classical GA, inspired by natural selection, is a population-based search algorithm based on the idea of survival of the fittest. The new populations are generated by iteratively applying the genetic operators to existing individuals of the population. The core elements of GA are the representation of chromosomes, selection, mutation, crossover, and evaluation of fitness function.

GA plays the role of training the network. Here, the bias of network with a chromosome that may be taken as real value vector and weights are encoded by employing an encoder component. On the other hand, there is also a decoder that performs the reverse job of decoding chromosomes in form of bias of network and weights. As GA function needs fitness function, so the Mean Absolute Error (MAE) is calculated taking the errors in training data into account as shown in Equation (3). In Equation (4), the fitness function is calculated. Algorithm 1 shows GA processes in the training model.

$$MAE = \frac{1}{N} \sum_{i=0}^N (y_i^p - y_i^a) \quad (3)$$

$$fitness = \frac{1}{MAE} \tag{4}$$

Algorithm 1: Genetic algorithm

Input : p size of population
 p_c crossover probability
 p_m mutation probability
 gen_{max} max number of generations

Output: best chromosome

initialize population $p = E_1, \dots, E_{p_\theta}$, each entity is a d -dimension vector
 $E_i = (e_{i1}, \dots, e_{id})$
 $gen = 1$
while ($gen \leq gen_{max}$) **do**
 calculate fitness with equation 4
 best chromosome $E_{best} =$ fittest chromosome
 selection (according to the fitness function)
 crossover (according to crossover probability)
 mutation (according to mutation probability)
end
return best chromosome

3.3. Particle Swarm Optimization

PSO is a well-known optimization algorithm that belongs to the swarm intelligence category. Several examples are available in the literature where neural networks are successfully trained using PSO. This algorithm essentially consists of numerous particles that float through the problem space analyzing the possible solutions and progressing to the best ones. The initialization of the algorithm is performed by generating N particles with arbitrary positions x_i moving with the velocities v_i . Any i th particle’s velocity vector can be denoted by $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{in})$ and the position vector $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ in N -dimensional space. Every particle maintains a record of the best-found positions/solution in the problem space called personal best denoted by $pbest_i$, as well as the best position found by all particles in the neighborhood represented by $gbest_i$. Let $pbest_i = (x_{i1}^{pbest}, x_{i2}^{pbest}, x_{i3}^{pbest}, \dots, x_{in}^{pbest})$, and $gbest = (x_1^{gbest}, x_2^{gbest}, x_3^{gbest}, \dots, x_n^{gbest})$ are, respectively, the best position of a particle i and the global best positions found by the particles. At every iteration k , every particle relocates itself to a new position depending on its previous location and velocity. On the contrary, the typical evolutionary algorithms evaluate the new solutions by using the operator’s mutation and crossover. The fitness is determined via objective functions, and eventually, this fitness evaluates which positions are the best possible solutions. Every particle position is updated depending on the best personal position of its own, as well as of its neighborhood. After (predefined) numbers of calculation steps, the particles should be converging toward the best solution/position by using equations of motion as shown in Equation (5).

$$v_i^{k+1} = wv_i^k + c_1r_1(pbest_i^k - x_i^k) + c_2r_2(gbest_i^k - p_i^k) \tag{5}$$

In Equation (5), v_i^k represents the velocity of the dimension d of the particle i in time t . Equation (6) is used to update the position of a particle. In Equation (6), p_i^k refers to the position of particle i at time t in d th dimension and v_i^{k+1} is the velocity calculated in Equation (5) [27]. Algorithm 2 shows the steps of PSO.

$$p_i^{k+1} = p_i^k + v_i^{k+1} \tag{6}$$

Algorithm 2: Particle Swarm Optimization

Input : initial population
Output: best particles
 P =population size
 $p=i_{th}$ particle in P
 calculate fitness of p according to Equation (4)
 calculate velocity of p according to Equation (5)
 $gbest$ =global best position
 $pbest$ =particles' best position
for each particle p in P **do**
 initialize X_{ij}^t randomly
 initialize velocity v randomly
 evaluate p_i
 update $pbest$ and $gbest$
end
 return best particle

3.4. Proposed Algorithm

The proposed resource utilization prediction algorithm is trained with hybrid GA-PSO for better resource management. Due to the huge size of user applications and large number of resources, resource management becomes a challenge. The accuracy of the resource prediction model strongly influences resource utilization and other factors related to cloud performance. The prediction model should have the ability to handle multiple resources. GA-PSO algorithms are used to combine the two training models to help the effective management of cloud resources. Literature shows that increasing the number of iteration of GA-based algorithms leads to improved solutions in comparison to other algorithms. However, large number of iterations leads to more execution time for convergence to best solutions. On the other hand, PSO yields solutions in less time as compared to other algorithms. However, PSO suffers from the problem of fast convergence and local optima. The proposed algorithm takes advantage of the strength of both GA and PSO to produce better results. The hybrid model uses diversity and fast convergence to optimal solutions and reaches the best solution comparatively faster than other algorithms. Hybrid GA-PSO based algorithm consists of attributes of both GA and PSO. The flow of the GA-PSO algorithm is shown in Figure 1. The procedure starts by generating a random population and applying GA to the initial population. After the procedure reaches half of the number of iterations, the procedure stops and the generated solutions are passed to PSO where local and global variables are calculated and variables are updated accordingly. The procedure stops when the maximum number of iterations reaches. The initialization in the hybrid GA-PSO algorithm is set up by certain number of iterations. In the first iteration, the solution is started randomly. A series of new solutions are generated at the end of the first iteration. These solutions are used in a recursive way to form a series of new solutions. In a standard GA, the population is represented as a set of chromosomes. Each chromosome consists of genes. In GA, the evolution of chromosomes occurs via crossover and mutation operators and the selection of offspring for the next generations. Roulette wheel selection is used in the proposed algorithm. The selection procedure uses spinning a wheel based on the fitness value of each chromosome. For crossover clustered crossover operator is used. In this procedure breaking points in each chromosome are selected and new chromosomes are generated by mixing the selected points. For mutation operator swap mutation is used where a pair of genes is selected and genes at both points are swapped. The remaining iterations are performed with PSO taking the population generated with GA. The position and velocity of the particles in the previous iteration are used to create new particles. The

position and velocity are changed on the basis of pbest and gbest parameters calculated at each iteration. These values change continuously with every iteration. In the first iteration, pbest is equal to solutions produced by GA. The gbest solution is the one with the lowest fitness value. In addition, a fitness-value based comparison is carried out between the previously created particles and newly created particles at each iteration. The pbest contains the particle having the best fitness value. At each iteration, the gbest stores the best particle among the entire generation by comparing the fitness value with the best value. This comparison at every step guarantees that all particles are marching towards the best solution to reach an optimal solution. Fitness is calculated in both cases with MAE. Algorithm 3 illustrates the implementation of the hybrid GA-PSO algorithm whereas Figure 2 shows the flow of the hybrid model in training the network.

Algorithm 3: Hybrid GA-PSO algorithm

Input :network weights

Output:best solutions

p_c crossover probability

p_m mutation probability

$iter_{max}$ max number of iterations

initialize population $p=E_1, \dots, E_{p_\theta}$, each entity is a d-dimension vector

$E_i = (e_{i1}, \dots, e_{id})$

$iter=1$

while ($iter \leq iter_{max}/2$) **do**

 calculate fitness with equation 4

 best chromosome E_{best} = fittest chromosome

 selection (according to the fitness function)

 crossover (according to crossover probability)

 mutation (according to mutation probability)

 select best chromosome ($chromosome_i$)

end

set particle $p=(chromosome_i)$

calculate fitness of p according to Eq. 4

calculate velocity of p according to Eq.5

$gbest$ =global best position

$pbest$ =particles' best position

while not reach $iter_{max}$ **do**

 linitialize X_{ij}^t randomly

 initialize velocity v randomly

 evaluate p_i

 update $pbest$ and $gbest$

end

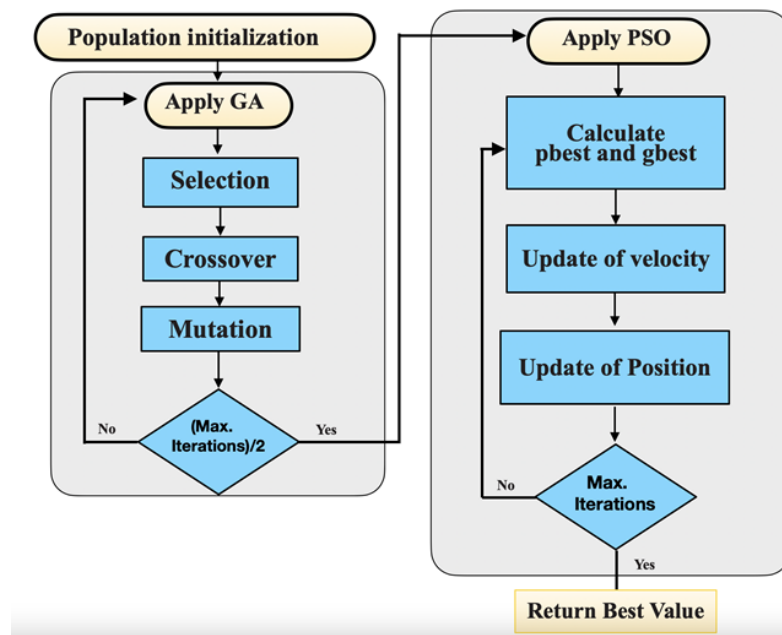


Figure 1. Illustration of the flow of GA-PSO algorithm.

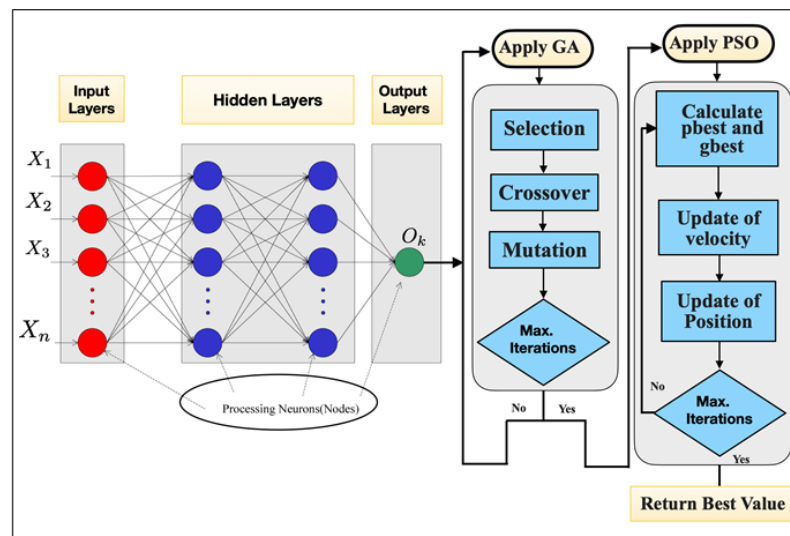


Figure 2. Illustration of the flow of the hybrid model in network training.

Figure 3 shows the predictive system of resources for the cloud [12]. The system consists of four modules, i.e., manager, preprocessor, trainer, and predictor. The Manager module is responsible for collecting the monitoring data regarding resources. The collected data are stored in a repository. A database is used to store the latest monitoring data of resources. The monitoring data stored in the database are created as historical monitoring data and time series. The preprocessor module creates these forms of monitoring data. The time-series data in raw form are transformed into supervised data by the preprocessor module to adjust the input by the neural network. A variety of mechanisms are available to be deployed that act to process the data regarding cloud workload. These include the normalized data, averaging data in the long run, grouping into multivariate time-series, and sliding window. After processing in preprocessor module, the data in output as historical resources data are fed into database collector. This historical resources data creates a model for predictions that are completed in the trainer module. For predicting the utilization of resources, the data are given to predictors. The proposed methodology includes a learning method added in the trainer module that uses FLNN. The GA-PSO mechanism trains the

network one by one for increasing the speed of convergence and enhancing the accuracy of forecasting. The designed learning mechanism is named FLGAPSONN. When the training process is completed, the trained module predicts the utilization of resources in the future which is done in the predictor module.

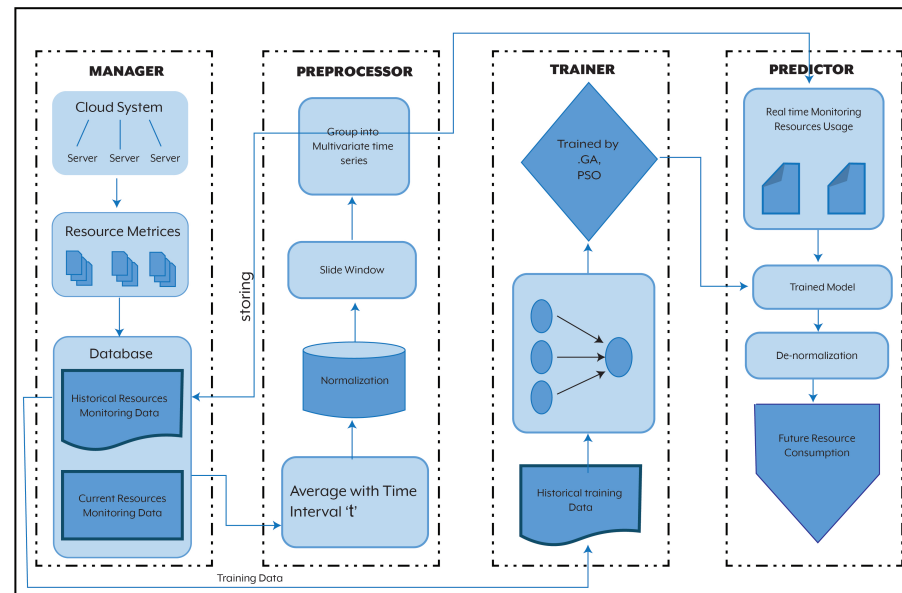


Figure 3. Architecture of the predictive systems.

4. Results and Discussion

This section presents details of the evaluation of the proposed hybrid model. The evaluation is carried out with publicly available Google cluster workload traces dataset [12,28]. Each job in the dataset consists of multiple concurrent tasks that run on different machines. The dataset consists of parameters such as CPU utilization, memory consumption, disk utilization, etc. According to the previous studies [12,28], less than 2% of the jobs take longer than one day. As used in previous experiments, a long-running job (ID 1617658948) consisting of 60,171 tasks was selected for evaluation. The job covers a 20 day period. The data of the first 15 days are used for training and the remaining data are used for testing. The evaluation consists of multivariate input where both CPU and memory are considered and univariate input where CPU or memory are considered. To evaluate the performance of the proposed hybrid GA-PSO model, we compared the results with traditional FLNN, FLGANN, and FLPSONN. The compared models consist of one input layer and one output layer with Exponential Linear Unit (ELU) as activation function.

Table 1 shows the comparative results of the proposed model with other models in terms of MAE. The results are calculated with sliding window of size 5. The results show that the hybrid GA-PSO model yields smaller values for both univariate and multivariate input cases. For univariate CPU utilization prediction, the MAE of the proposed model is 0.25 which is smaller than the 0.32, 0.29, and 0.36 produced by FLPSONN, FLGANN, and FLNN models, respectively. In case of univariate memory consumption prediction, the proposed model produced MAE of 0.018 which is smaller than the compared methods. In case of multivariate CPU utilization prediction, the hybrid model produced MAE of 0.33 which is smaller than the compared methods. Similar results can be observed in case of multivariate memory consumption prediction.

Table 1. Comparative results of the proposed model with other models. The results are reported in terms of MAE.

| Input | Model | CPU | Memory |
|---------------|-----------|------|--------|
| Univariate | FLGAPSONN | 0.25 | 0.018 |
| | FLGANN | 0.29 | 0.021 |
| | FLPSONN | 0.32 | 0.024 |
| | FLNN | 0.36 | 0.027 |
| Multiivariate | FLGAPSONN | 0.33 | 0.026 |
| | FLGANN | 0.41 | 0.033 |
| | FLPSONN | 0.43 | 0.035 |
| | FLNN | 0.47 | 0.039 |

Figures 4 and 5 show graph plots of univariate CPU and memory utilization prediction for different models. The results include actual and predicted values. The results show that the values produced by the hybrid models are closer to the actual values as compared to other compared models. The results of multivariate CPU and memory utilization prediction are shown in Figures 6 and 7, respectively. Like previous results, the hybrid model produced closer results to the actual values.

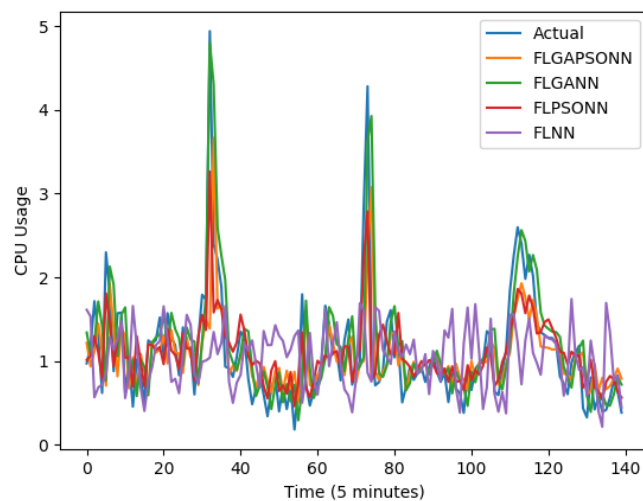


Figure 4. Univariate CPU utilization prediction for different models.

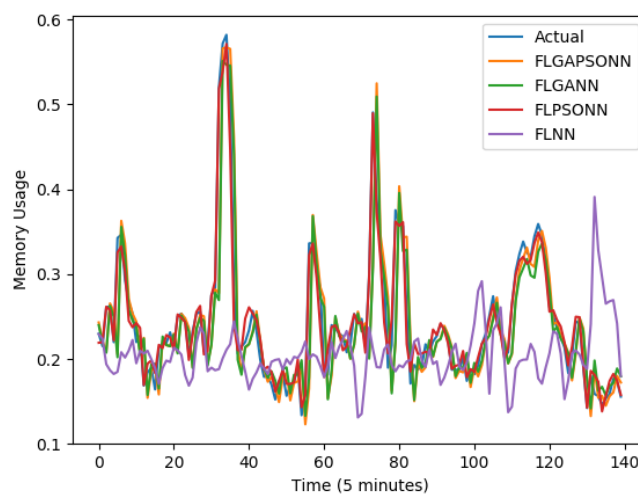


Figure 5. Univariate memory utilization prediction for different models.

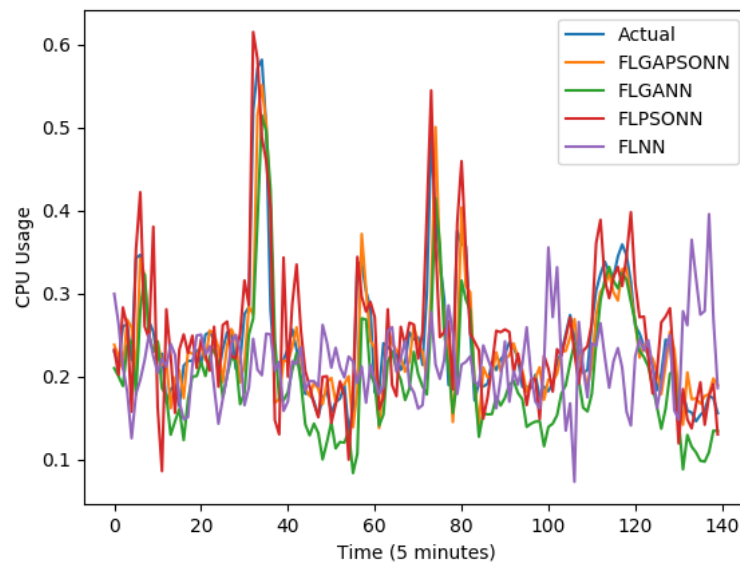


Figure 6. Multiivariate CPU utilization prediction for different models.

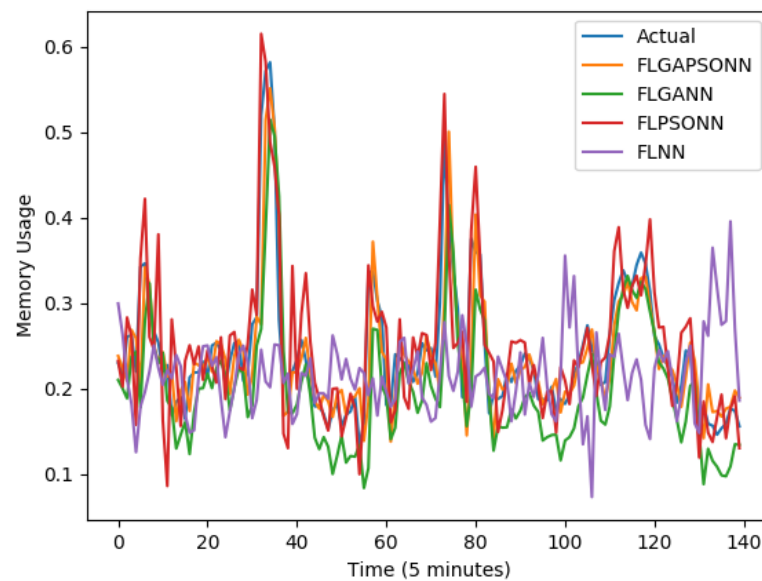


Figure 7. Multiivariate memory utilization prediction for different models.

Percent improvement gain in terms of MAE of the proposed model over other compared models is presented in Figures 8 and 9 for univariate and multivariate input cases, respectively. In case of univariate CPU utilization, the proposed technique achieved percent improvement gain of 21.87, 13.75, and 30.55 over FLPSONN, FLGANN, and FLNN, respectively. In the case of memory utilization prediction, the percentage improvement gain over the compared models is 25.0, 14.28, and 33.3. In the case of multivariate input, the percentage improvement for CPU in terms of MAE over FLPSONN, FLGANN, and FLNN is 23.25, 19.51, and 29.78, respectively. Improvement gain in memory over FLPSONN, FLGANN, and FLNN for multivariate input is 25.71, 21.21, and 33.33, respectively.

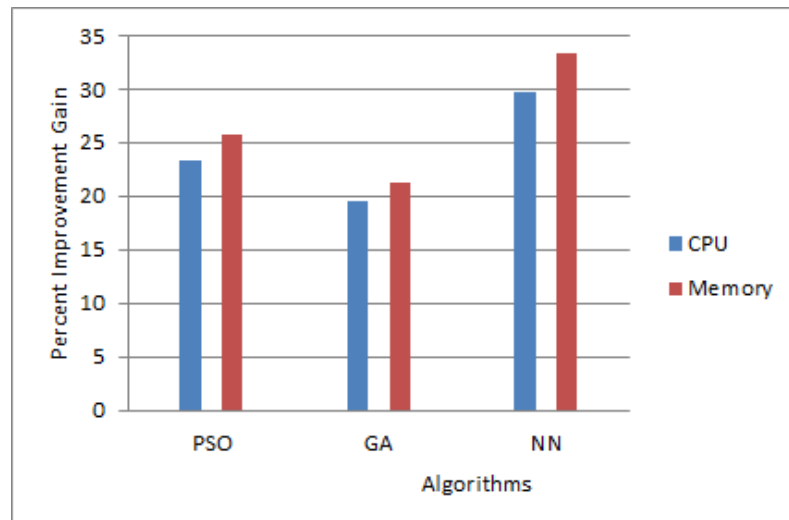


Figure 8. Percentage improvement gain of the proposed model over the other models on univariate input case.

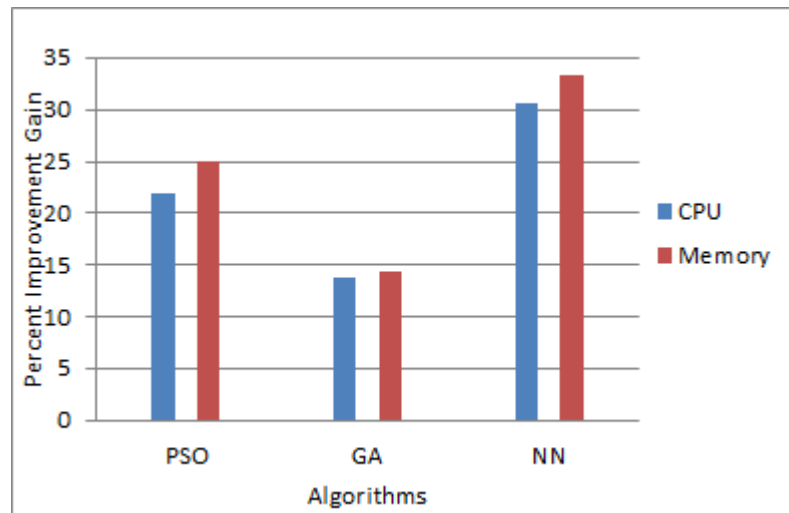


Figure 9. Percentage improvement gain of the proposed models over the other models on multiivariate input case.

5. Conclusions

Predicting cloud resource utilization is an important issue to handle uncertainty in cloud computing environments. In cloud computing, resources are allocated to user applications accessible from anywhere over the Internet. To handle large number of users, the resources need to be scaled dynamically for efficient utilization, reduced energy consumption, and cost with better Quality of Service (QoS). The focus of this research work is to explore the efficiency of neural networks to predict multi-resource utilization. The proposed model uses hybrid GA-PSO to train network weights and uses FLNN for prediction. The hybrid model is capable of training a network for accurate prediction of multivariate resource utilization. The proposed model is validated with comparative experimental results. The results show that the proposed hybrid model yields better accuracy as compared to traditional techniques. It can also be concluded that due to the possibility of rapid and excessive changes in resource utilization, the prediction of multi-variate resource utilization is a challenging task.

The potential directions for future research can be to evaluate neural network predictors further in other areas of cloud computing, such as predicting other resources such as disk utilization, cost-effectiveness, network, and reduction in energy consumption for

green computing. The proposed framework is evaluated with the Google trace dataset for memory and CPU utilization. It would be constructive to endorse the proposed evolutionary neural network approach further by working on other multi-variate resource utilization datasets.

Author Contributions: Conceptualization, S.M. and M.S.; methodology, S.M.; software, S.M.; validation, M.T., M.S. and A.A.; formal analysis, M.S., M.T.; investigation, S.M.; resources, M.S., M.T.; data curation, S.M., M.T., S.M.; writing—original draft preparation, S.M.; writing—review and editing, M.S., M.T., A.A.; visualization, M.T.; supervision, M.S.; project administration, A.A., M.S.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used are available publicly.

Conflicts of Interest: There is no conflicts of interest to report.

References

1. Kaur, G.; Bala, A.; Chana, I. An intelligent regressive ensemble approach for predicting resource usage in cloud computing. *J. Parallel Distrib. Comput.* **2019**, *123*, 1–12. [[CrossRef](#)]
2. Muteeh, A.; Sardaraz, M.; Tahir, M. MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization. *Clust. Comput.* **2021**, *24*, 3135–3145. [[CrossRef](#)]
3. Malik, N.; Sardaraz, M.; Tahir, M.; Shah, B.; Ali, G.; Moreira, F. Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Thresholds. *Appl. Sci.* **2021**, *11*, 5849. [[CrossRef](#)]
4. Rahmanian, A.A.; Ghobaei-Arani, M.; Tofighy, S. A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Future Gener. Comput. Syst.* **2018**, *79*, 54–71. [[CrossRef](#)]
5. Mason, K.; Duggan, M.; Barrett, E.; Duggan, J.; Howley, E. Predicting host CPU utilization in the cloud using evolutionary neural networks. *Future Gener. Comput. Syst.* **2018**, *86*, 162–173. [[CrossRef](#)]
6. Liang, Z.; Ouyang, J.; Yang, F. A hybrid GA-PSO optimization algorithm for conformal antenna array pattern synthesis. *J. Electromagn. Waves Appl.* **2018**, *32*, 1601–1615. [[CrossRef](#)]
7. Moslehi, F.; Haeri, A.; Martinez-Alvarez, F. A novel hybrid GA-PSO framework for mining quantitative association rules. *Soft Comput.* **2020**, *24*, 4645–4666. [[CrossRef](#)]
8. Anand, A.; Suganthi, L. Hybrid GA-PSO optimization of artificial neural network for forecasting electricity demand. *Energies* **2018**, *11*, 728. [[CrossRef](#)]
9. Manasrah, A.M.; Ba Ali, H. Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1934784. [[CrossRef](#)]
10. Zhang, X.; Guo, P.; Zhang, H.; Yao, J. Hybrid Particle Swarm Optimization Algorithm for Process Planning. *Mathematics* **2020**, *8*, 1745. [[CrossRef](#)]
11. Ru, N.; Jianhua, Y. A GA and particle swarm optimization based hybrid algorithm. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 1047–1050.
12. Nguyen, T.; Tran, N.; Nguyen, B.M.; Nguyen, G. A resource usage prediction system using functional-link and genetic algorithm neural network for multivariate cloud metrics. In Proceedings of the 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), Paris, France, 20–22 November 2018; pp. 49–56.
13. Liu, C.; Liu, C.; Shang, Y.; Chen, S.; Cheng, B.; Chen, J. An adaptive prediction approach based on workload pattern discrimination in the cloud. *J. Netw. Comput. Appl.* **2017**, *80*, 35–44. [[CrossRef](#)]
14. Moreno-Vozmediano, R.; Montero, R.S.; Huedo, E.; Llorente, I.M. Efficient resource provisioning for elastic Cloud services based on machine learning techniques. *J. Cloud Comput.* **2019**, *8*, 5. [[CrossRef](#)]
15. Song, B.; Yu, Y.; Zhou, Y.; Wang, Z.; Du, S. Host load prediction with long short-term memory in cloud computing. *J. Supercomput.* **2018**, *74*, 6554–6568. [[CrossRef](#)]
16. Sniezynski, B.; Nawrocki, P.; Wilk, M.; Jarzab, M.; Zielinski, K. VM reservation plan adaptation using machine learning in cloud computing. *J. Grid Comput.* **2019**, *17*, 797–812. [[CrossRef](#)]
17. Kumar, J.; Singh, A.K. Workload prediction in cloud using artificial neural network and adaptive differential evolution. *Future Gener. Comput. Syst.* **2018**, *81*, 41–52. [[CrossRef](#)]
18. Kumar, J.; Goomer, R.; Singh, A.K. Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters. *Procedia Comput. Sci.* **2018**, *125*, 676–682. [[CrossRef](#)]
19. Tran, N.; Nguyen, T.; Nguyen, B.M.; Nguyen, G. A multivariate fuzzy time series resource forecast model for clouds using LSTM and data correlation analysis. *Procedia Comput. Sci.* **2018**, *126*, 636–645. [[CrossRef](#)]
20. Babu, G.P.; Tiwari, A. Energy Efficient Scheduling Algorithm for Cloud Computing Systems Based on Prediction Model. *Int. J. Adv. Netw. Appl.* **2019**, *10*, 4013–4018. [[CrossRef](#)]

21. Ramanathan, R.; Latha, B. Towards optimal resource provisioning for Hadoop-MapReduce jobs using scale-out strategy and its performance analysis in private cloud environment. *Clust. Comput.* **2019**, *22*, 14061–14071. [[CrossRef](#)]
22. Gupta, S.; Dileep, A.D.; Gonsalves, T.A. Online sparse blstm models for resource usage prediction in cloud datacentres. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2335–2349. [[CrossRef](#)]
23. Saxena, D.; Singh, A.K.; Buyya, R. OP-MLB: An online VM prediction based multi-objective load balancing framework for resource management at cloud datacenter. *IEEE Trans. Cloud Comput.* **2021**. [[CrossRef](#)]
24. Ouhamme, S.; Hadi, Y.; Ullah, A. An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. *Neural Comput. Appl.* **2021**, *33*, 10043–10055. [[CrossRef](#)]
25. Abdullah, L.; Li, H.; Al-Jamali, S.; Al-Badwi, A.; Ruan, C. Predicting multi-attribute host resource utilization using support vector regression technique. *IEEE Access* **2020**, *8*, 66048–66067. [[CrossRef](#)]
26. Hassim, Y.M.M.; Ghazali, R. Functional link neural network–artificial bee colony for time series temperature prediction. In Proceedings of the International Conference on Computational Science and Its Applications, Ho Chi Minh City, Vietnam, 24–27 June 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 427–437.
27. Sardaraz, M.; Tahir, M. A hybrid algorithm for scheduling scientific workflows in cloud computing. *IEEE Access* **2019**, *7*, 186137–186146. [[CrossRef](#)]
28. Reiss, C.; Tumanov, A.; Ganger, G.R.; Katz, R.H.; Kozuch, M.A. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In Proceedings of the Third ACM Symposium on Cloud Computing, San Jose, CA, USA, 14–17 October 2012; pp. 1–13.