Hybrid Bio-Inspired Optimization-based Cloud Resource Demand Prediction using Improved Support Vector Machine

Nisha Sanjay[1], Dr. Sasikumaran Sreedharan[2]
Research Scholar[1], Research Supervisor[2],
Faculty of Computer Science and Multimedia, Lincoln University College Marian Research Centre,
Marian College Kuttikanam (Autonomous), Kerala, India[1, 2]

***Abstract*—In order to furnish diverse resource requirements in cloud computing, numerous resources are integrated into a data centre. How to deliver resources in a timely and accurate manner to meet user expectations is a significant concern. However, the resource demands of users fluctuate greatly and frequently change regularly. It's possible that the resource provision won't happen on time. Furthermore, because some physical resources are shut down to save energy, there may occasionally not be enough of them to meet user requests. Therefore, it's critical to offer resource provision proactively to ensure positive user involvement using cloud computing. To enable resource provision in advance, it is essential to accurately estimate future resource demands. Using machine learning techniques, we offer a unique approach in this study that tries to identify key features, accelerating the forecast of cloud resource consumption. Finding the classification method with the greatest fit and maximum classification accuracy is crucial when predicting cloud resource consumption. The attribute selection method is used to decrease the dataset. The categorization process is then given the reduced data. The hybrid attribute selection method used in the investigation, which combines the bio-inspired algorithm genetic algorithm, the pulse-coupled neural network, and the particle swarm optimization algorithm, improves classification accuracy. The accuracy of prediction employing this technique is examined using a variety of performance criteria. When it comes to predicting the demand for cloud resources, the experimental results show that the suggested machine learning method performs more effectively than traditional machine learning models.**

Keywords*—Cloud computing; resource demand; machine learning; cloud resource demand prediction; bio-inspired algorithm

I. Introduction

With the benefits of characteristic features like resource pool and a pay-as-you-go paradigm, cloud computing has found widespread use in a variety of industries. Infrastructure as a Service (IaaS) is a brand-new cloud service paradigm which offers clients virtual machines as resources (VMs). Accurate and timely allocation of VMs to client tasks in IaaS service model is a major concern. Some cloud providers continue to offer VMs statically, which results in increased operational cost for customers and reduced resource usage for cloud providers. A better alternative is to create VMs dynamically in response to the current resource demands.

However, resource demands fluctuate significantly at times and change continuously throughout time. Users can quickly apply for numerous VMs, for instance. This causes these VMs to take a very lengthy time to create. Due to some of the physical resources being shut down to save energy, even those that are currently in operation may not be enough to meet user requests. For this reason, developing a proactive resource provision is essential to guaranteeing that customers get a positive cloud computing experience. In response to anticipated resource demands, proactive resource provision might offer resources in advance. However, if overestimated, this can be a resource waster.

It goes without saying that if the resource demand is estimated to be less than the actual requirements, user demand of the resources are not met. Therefore, the main challenge is to effectively forecast future resource demands to reduce overestimation and underestimation. An overview of the challenges and approaches for forecasting usage of resources in cloud computing can be found in the study [1]. To prevent resource over-provisioning, Chen et al. [2] developed a forecast method exclusively for burst workload. To eliminate bursts and sounds, this technique employs the Fast Fourier Transform (FFT) algorithm, which increases prediction accuracy. A cloud workload prediction model is put out by Roy et al. [3]. This model predicts future workload using autoregressive moving average method of the second order (ARMA) and then uses a performance model of the average app response time to forecast resource requirements.

Using ensemble models, two self-adaptive resource demand prediction techniques are proposed [4, 5]. In order to increase prediction accuracy, Xu et al. [6] propose the GFSS-ANFIS/SARIMA prediction model, which integrates Seasonal Autoregressive Integrated Moving Average Model (ARIMA) and Generalized Fuzzy Soft Sets with Adaptive Neuro Fuzzy Inference System. Data mining and statistical techniques are used in Verma et al. [7] resource prediction framework for multi-tenant service clouds to forecast resource demands in order to minimize resources and provisioning time. To obtain precise performance forecasts in hybrid clouds, Imai et al. [8] suggest a model which has workload-tailored elastic compute unit (WECU) as a computing power unit. Brown's quadratic exponential smoothing approach is used by Mi et al. [9] to forecast activities ahead and adjust resources as needed in a data center.

The resource prediction method used by Minarolli et al. [10] incorporates the cross relation of asset utilization among VMs that are part of the same application. Bankole et al. [11] proposed three performance prediction models that employ neural networks (NN), linear regression (LR), and support vector regression (SVR) methods independently. The experimental findings demonstrate that SVR is the favored model, outperforming other models in terms of prediction accuracy. Similar to this, certain machine learning techniques, such as the multi-layer perceptron (MLP) [12], Support vector regression (SVR) [12], deep belief network (DBN) [13], and artificial neural network (ANN) [14], are used to forecast resource utilizations. There are two categories of prediction. The former involves statistical methods and the latter employs machine learning techniques. Even though machine learning methods can produce predictions with a higher degree of accuracy, it requires setting up the training model and extracting the features from a humungous amount of data. Though sampling data might occasionally change greatly and be insufficient as inputs for machine learning techniques. The prediction accuracy of statistical methods is often poor for non-stationary and non-linear data.

One such well-known metaheuristic method based on swarm-intelligence is particle swarm optimization (PSO), which has demonstrated its superiority in resolving a variety of real-world optimization issues from fluid mechanics, wireless sensor networks, engineering, applied sciences and academia [15]. Despite its success in locating competitive solutions, the PSO still has trouble sustaining strong exploration and is susceptible to becoming stuck in local optima, which leads to an insufficient exploration-exploitation tradeoff [16]. These flaws have an impact on the subsequent quality of the scheduling solution. Additionally, the "No Free Lunch Theorem (NFL)" [17] establishes the impossibility of finding a single metaheuristic algorithm capable of efficiently solving all optimization issues. These facts serve as powerful impetuses for the current research project, which proposes a hybrid PSO-based scheduling solution to get over the constraints of regular PSO by combining it with genetic algorithm (GA) and pulse coupled neural networks (PCNN).

The remainder of the paper is organized as below. Section II includes an overview of prior studies in the same field. The methodology, system workflow, feature extraction, and demand prediction employed in the proposed approach are covered in Section III. Section IV includes the performance analysis of the proposed hybrid solution. Section V provides a conclusion, marking the end of the paper.

II. Related Work

To solve cloud based scheduling (CBS) difficulties, current research uses heuristics and metaheuristics, such as shortest job first (SJF), First come first served (FCFS), Max-min, Min-min, Minimum completion time (MCT), Minimum execution time (MET), and Suffrage [18]. Heuristics offer problem-specific solutions and are apt for solving minor problems. In contrast, ensemble methods (MHs), are simple, iterative, adaptable, highly speculative, algorithms that direct a subordinated heuristic through smart mechanism [19]. For complicated and larger scheduling issues, metaheuristics-based scheduling solutions have outperformed problem-specific heuristics [20]. However, MHs typically have a few flaws, such as premature convergence, getting stuck in neighboring best value, an absence diversity, and imbalance among the examination and development stages of the energy spectrum [21]. If these flaws are applied to work scheduling issues, the results can be undesirable.

The research has also advocated the use of combination heuristics to address the drawbacks of solo metaheuristics [22]. Several whale optimization algorithms (WOA)-based scheduling approaches have already been put out for scheduling bag-of tasks (BoT) applications to get results that are almost optimal and are motivated by the humpback whales' hunting method. These solutions include those that employ conventional, customized, and fusion of WOA techniques [23]. A standard WOA, Gaussian model, and opposition-based learning (OBL) approaches are combined and used in a cloud scheduling solution called GCWOAS2 [24] to provide effective task-resource couples. One more current study paper [23] proposes a combination metaheuristic approach dubbed OWPSO to address the shortcomings of the original WOA by combining OBL and PSO algorithms. The authors of [25] proposed random double adaptive WOA (RDWOA) employing the Bee optimization algorithm techniques for arranging cloud workloads to decrease implementation cost and time.

The increase and decrease operators might be augmented to the typical WOA to enhance search efficiency, according to authors in [26], who also proposed utilizing two advanced optimization algorithms. In study [27], authors proposed an Improved WOA for Cloud task scheduling (IWC) algorithm that makes use of the local weight method to enhance neighboring explore effectiveness and prevent the basic WOA's early convergence. WOA and harmony search algorithm (HS) were hybridized to create WHOA by the authors of [28] in order to reduce execution costs and energy usage. The WOA-based cloud task scheduling solution, which simultaneously optimizes makepan and energy usage, was proposed by Sharma and Garg [29]. To schedule BoT applications over clouds, the whale-Scheduler technique was recommended, yielding the best makespan and execution cost [30]. There have been many different GA-based scheduling approaches proposed in the past, including the basic GA method [31], improved GA strategies employing modified mutation as well as crossover procedures, and fusion of GA results that combine classic GA among additional methodologies [32].

Numerous studies have greatly improved scheduling efficiency over cutting-edge heuristics by using basic variants of SOS and particular improvements to Symbiotic Organism Search (SOS) algorithms utilizing chaotic sequences with dissent learning [33]. To overcome CBS concerns and achieve different QoS objectives, researchers have suggested conventional ant colony optimization (ACO) as well as other altered ACO-based optimization techniques [34]. In numerous researches, classic PSO algorithms, modified PSO variations, and hybrid PSO variants have all been applied to handle CBS situations in order to achieve a variety of objectives, including minimizing computation time and complexity and addressing load balance challenges [35]. An adjustable inertia weighting method is suggested for the CBS problem to tradeoff between exploration and exploitation [36]. For maintaining population variety and enhancing solution quality, Chen and Long [37] proposed a fusion of optimization approach integrating PSO and ACO algorithms. A bi-objective PSO scheduler was used by the developers of [38] to improve system performance and lower execution costs. A multi-objective PSO strategy is used in two deadline-constrained scheduling techniques to enhance QoS metrics values [39].

A non - linear and non PSO was employed by the authors of Ref. [40] to lessen the time for scheduling the workload. Several strategies for workload allocation in a cloud have been suggested, using both a standard Cuckoo Search (CS) implementation [41] as well as modifications to the CS's basic structure and the incorporation of additional metaheuristics. Chhabra et al. proposed a fusion of CS method in [42] to enhance the exploration potential of standard CS and to achieve more appropriate scheduling than current generation heuristics. This metheuristic merged CS and DE algorithms. a solid metaheuristic in the form. GWO has been employed in the past to create almost ideal scheduling solutions to improve various QoS metrics. To optimize both makespan and energy, for instance, a MO-GWO technique was recommended [43]. Modified GWO is formulated in [44] with alterations on the fitness function to consider makespan and cost. In [45], mean GWO is studied to achieve better performance tackling scheduling concerns. It results in lower makespan and reduced energy usage.

Elaziz et al. in [46] integrated the DE algorithm's effective local searching feature and Moth search (MS) method for better scheduling solution. The Bacterial Foraging Optimization (BFO)-based scheduling strategy has been recommended by Milan et al. [47] to optimize extent of imbalance, idle time, and overall execution time. The study in [48] proposes a novel approach to cloud computing scheduling problem solution: Water Pressure Change Optimization (WPCO). The phenomena of water density changing as pressure is increased due to changes in the physical properties of water serves as the inspiration for the novel WPCO technology. WPCO provides the best solution quality in comparison to the standard metaheuristics. According to the authors of [49], a scheduling strategy based on social group optimization algorithm (SGO) is proposed for a diverse cloud environment that can be used to resolve CBS issues with the highest possible throughput and the shortest possible makespan.

Inadequate exploration and utilization process balancing, slow convergence, a failure to focus on schedule order optimization, a lack of products developed using standard workloads, a lack of numerical solutions to tune metaheuristic variables, and concurrent performance and energy consumption optimization are common problems or limitations of the current research studies based on metaheuristic approach for scheduling BoT applications over cloud systems. These flaws leave a lot of room for developing new metaheuristics or refining already-existing ones to increase the CBS problem's efficiency.

III. Proposed Approach

Fig. 1 depicts the planned work's entire organizational structure. It consists of two phases namely training and using the constructed model for prediction. The major goal of the training phase is to learn about the cloud resource requirements and how to predict resource demand from provided data sets. As a result, it is referred to as the training phase. At this point, a method called attribute selection is used to cut down on the amount of data that was collected. The correlating attributes are identified when the data set's dimensions are reduced. These characteristics are crucial for forecasting cloud resource consumption.

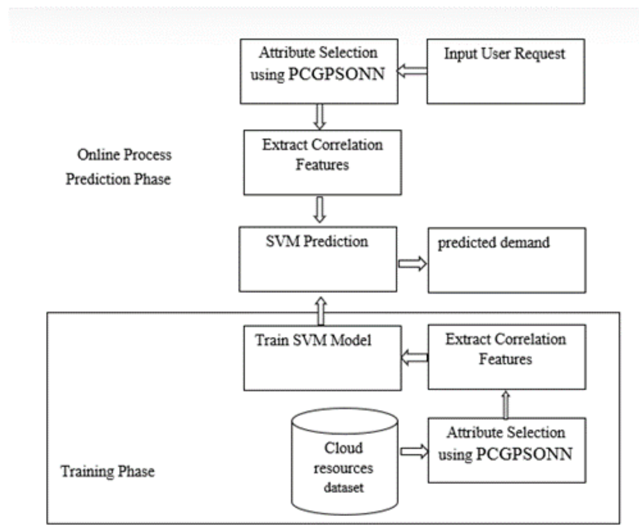

Fig. 1. Organizational structure of the proposed approach.

The steps of training and prediction are the same. The similar procedure was performed throughout prediction for the test data. Finally, the anticipated outcomes will be attained. In the following section, each procedure is thoroughly explained. The following list of modules contains the working stages of the planned work.

1) Attribute Selection using PCGPSONN.
2) Correlation feature extraction.
3) Cloud Resource Demand Prediction using SVM.

A. Attribute Selection using PCGPSONN

Only the unique Pulse Coupled Genetic Particle Swarm Optimization Neural Network (PCGPSONN) is used to determine the cloud resource database's most crucial properties. The attributes must be carefully chosen for an accurate demand prediction of cloud resources. Low accuracy, prediction inaccuracy, or failure might result from the incorrect selection of these attributes. If the feature selection initial selection was incorrect, the approach will never reach the global minimum and more runs will only take the algorithm to a local minimum. The best position for each particle, Pg, as determined by the Genetic Particle Swarm Optimization (GPSO) algorithm, is searched for in this work using the Pulse Coupled Neural Network (PCNN) algorithm.

In GPSO method, the GA operators and the PSO update mechanism typically operate with the same population throughout initialization step. Uniformly distributed random numbers should be used to create the initial population of all its members. Therefore, the attribute selection process must go through more iterations because of this random distribution's sluggish convergence. However, in our suggested method, GPCNN solutions are used to allocate the PSO's initial population. GPCNN and PSO split the entire number of iterations evenly. GPCNN runs the first half of the iterations, and the answers are provided as the PSO's initial population. PSO is in charge of the final iterations. Thus, the issue of delayed convergence is resolved, and the attribute selection process requires fewer iterations.

Additionally, the hybrid approach we suggest should include local and worldwide search. Then, for each particle, generate the best position Pg, and the attribute selection method is then given these best positions Pg to further refine the search process. Consequently, our hybrid strategy performs better than this approach. The PCGPSONN algorithm is displayed below.

Algorithm of PCGPSONN:

Input:

- Attributes (X) and their values.
- Population Size (pop_size): Number of individuals in the population.
- Max Generations (max_gen): Maximum number of generations.
- Crossover Probability (crossover_prob): Probability of crossover occurring.
- Mutation Probability (mutation_prob): Probability of mutation occurring.
- Particle Swarm Size (swarm_size): Number of particles in the swarm.
- Inertia Weight (inertia_wt): Weight for the inertia term in PSO.
- C1 and C2 (c1, c2): Constants for the cognitive and social components in PSO.

Output:

- Best Attribute (S): The best solution found.
- Fitness Value (W): The fitness value corresponding to the best solution.

1. Initialize Population:
    - Generate an initial population of solutions (pop_size) randomly.
2. Loop through Generations:
    - Repeat for a specified number of generations (max_gen) or until a convergence criterion is met.

    2.1 Evaluate Fitness:

    - Evaluate the fitness W of each solution X in the population.

    2.2 Genetic Algorithm Steps:

    - Select solutions S1 for crossover based on their fitness.
    - Perform crossover with a certain probability (crossover_prob).
    - Mutate selected solutions with a certain probability (mutation_prob).

    2.3 Particle Swarm Optimization (PSO) Steps:

    - Initialize a particle swarm (swarm_size) with positions and velocities.
    - Evaluate the fitness W of each particle X.
    - Update particle positions and velocities based on PSO equations.
    - Track the global best position S2 found by the swarm.

    2.4 Combine Genetic and PSO Steps:

    - Replace the worst solutions S1 in the population with the best particles from the swarm S2.

    2.5 Update Population:

    - Create a new population for the next generation by combining the modified population and the PSO swarm.
3. Return Best Solution:
    - Return the best solution S found in the final population.
4. Execute PCNN:

- Execute PCNN with W and S

PCNN algorithm

Alpha_F = 0.1 Decay term for feeding

Alpha_L = 1.0 Decay term for linking

Alpha_T = 1.0 Decay term for threshold

V_F = 0.5 Magnitude scaling term for feeding

V_L = 0.2 Magnitude scaling term for linking

V_T = 20.0 Magnitude scaling term for linking

Beta = 0.1 Linking strength

Num = 100 The number of iterations

W= [0.5 1 0.5;1 0 1;0.5 1 0.5] Initial values for W

M = [0.5 1 0.5;1 0 1;0.5 1 0.5] Initial values for M

F = zeros(size(S)) Initial values for F

L = F Initial values for L

Y = F Initial values for Y

U = F Initial values for U

T = Ones(size(S)) Initial values for T

S = im2double(S) Normalizing to lie within [0,1]

for n = 1:Num

F = exp(_Alpha_F) *F + V_F*conv2(Y,W) + S Update the feeding input

L = exp(_Alpha_L) * L + V_L*conv2(Y,M) Update the linking input

U = F_ *(1 + Beta*L) Compute the internal activation

Y = double(UiT) Update the output

T = exp(_Alpha_T) *T + V_T*Y Update the threshold input

End

The genetic algorithm's fitness value is W in this case and S is the best quality of genetic algorithms. The proposed approach chooses seven of the 11 attributes that are present in the cloud resource dataset. Table I provides a description of the complete attributes. Table II provides descriptions of the chosen features.

Based on the values of the attribute weights, PCGPSONN chooses the attribute. Instead of using binary presentation, a population of 200 clients used real-valued representation since the parameter coefficients were expressed using real-valued numbers rather than just 0 and 1. Seven separate attribute weight sets totaling 28 qualities made up each individual. The beginning population's members were chosen using machine learning techniques and expert-set weights. More precisely specified is the initial population. The PCGPSONN used a uniform crossover with discrete recombination for offspring creation and a roulette-wheel selection for parent selection. 80.0% of the time was spent on the crossover, and each gene's crossover points were selected separately and arbitrarily. The gene underwent mutation with a likelihood of 1.0% and was uniformly carried out by selecting a random value at random from the range and setting it as the new value at the present place. Additionally, elitism was employed during runs to preserve the greatest person among the population. The weight set that performed the best during evolution was something we did not want to lose. If the total population was higher than 21 at the culmination of the generation, a strategy for survivor screening was used. Those with the lowest categorization accuracy were removed from the population after the individuals were graded according to their accuracy. After 20 generations, the PCGPSONN algorithm concluded, or sooner if the best classification accuracy remained constant during a period of 10 iterations. Additionally, the examination came to an end if every member of the population were the same. The proposed PCGPSONN approach is contrasted with the GPSO and GPCNN approaches to demonstrate its efficacy. Tables III and IV display the GPSO and GPCNN results, respectively.

The selected attribute list of GPSO is shown in Table III. In GPSO approach first genetic algorithm is completed to get the fitness values of all attributes and then it is given to the PSO approach to complete the attribute selection process.

TABLE I. FEATURES OF CLOUD RESOURCE DATASET

S.No	Feature name
1	Timestamp
2	Disk read throughput
3	Disk write throughput
4	Network transmitted throughput
5	Provisioned capacity for CPU
6	Use of CPU [MHZ]
7	Use of CPU [%]
8	Provisioned Memory capacity [KB]
9	Memory consumed [KB]
10	Network received throughput [KB/s]
11	CPU cores

TABLE II. SELECTED FEATURES BY PCGPSONN

S.No	Feature name
1	Provisioned capacity for CPU
2	Use of CPU [MHZ]
3	Use of CPU [%]
4	Provisioned Memory capacity [KB]
5	Memory consumed [KB]
6	Network received throughput [KB/s]
7	CPU cores

TABLE III. SELECTED ATTRIBUTES BY GPSO

S.No	Feature name
1	Disk read throughput
2	Disk write throughput
3	Network transmitted throughput
4	Provisioned capacity for CPU
5	Use of CPU [MHZ]
6	Use of CPU [%]
7	Memory capacity provisioned [KB]
8	Memory usage [KB]
9	Network received throughput [KB/s]
10	CPU cores

Table IV displays the GPCNN's chosen attribute list. When using the GPCNN strategy, all attributes' fitness values are first obtained using a genetic algorithm, and they are then given to the PCNN approach for double threshold operation during the attribute selection phase.

TABLE IV. SELECTED ATTRIBUTES BY GPCNN

S.No	Feature name
1	Disk read throughput
2	Disk write throughput
3	CPU capacity provisioned
4	Use of CPU [MHZ]
5	Use of CPU [%]
6	Memory capacity provisioned [KB]
7	Memory usage [KB]
8	Network received throughput [KB/s]
9	CPU cores

B. Correlation Feature Extraction

The degree to which two or more variables vary together is shown by a statistical metric known as correlation. It describes, in layman's terms, how much one variable changes in reaction to a slight variation in another. In accordance with the direction of the change, it might have positive, negative, or zero values. An independent attribute's strong significance in affecting the output is shown by a high correlation value between it and a dependent attribute. Finding the connection between the dependent and all of the independent variables in a multiple regression setup with numerous parameters is required to produce a more accurate and useful model. It is important to always keep in mind that additional characteristics do not necessarily translate into greater accuracy. Irrelevant characteristics would add unnecessary noise to our model and the accuracy could fall.

The Pearson r correlation method used in this study can be used to identify correlation between two attributes. The Pearson r correlation is the most often used correlation statistic to assess the degree of relationship between two linearly related characteristics. It is possible to determine the Pearson correlation between any two qualities, x, and y using:

$$r_{xy} = \frac{n\Sigma x_i y_i - \Sigma x_i y_i}{\sqrt{n\sum x_i^2 - (\Sigma x_i)^2}\sqrt{n\sum y_i^2 - (\Sigma y_i)^2}} \quad (1)$$

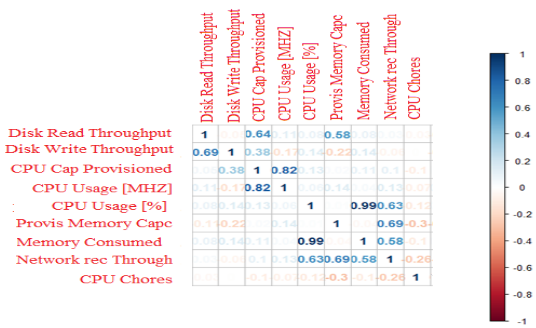

Fig. 2. Correlation Feature extracted values using Pearson correlation method.

Pearson correlation coefficient, often denoted as r, that calculate the Pearson correlation coefficients between each selected attributes which are derived from section 3.1 and the target variable (cloud resource demand). Using Pearson correlation coefficient, the extracted correlation feature table is shown in Fig. 2. In the Fig. 2, the Pearson correlation ranges from -1 to 1, where:

- r=1 indicates a perfect positive linear relationship.
- r=−1 indicates a perfect negative linear relationship.
- r=0 indicates no linear relationship.

These extracted correlation feature values are given into the SVM to predict the cloud resource demand.

C. Cloud Resource Demand Prediction using SVM

The equations are an exception to the prescribed specifications of this template. You will need to determine w Regression analysis and classification are two applications of support vector machines, commonly referred to as support vector networks or SVMs, which are supervised learning models in machine learning. They examine data and identify patterns. Provided a set of training examples that have been labelled as falling into one of two categories, an SVM training process builds a model that places new examples into either group. The model is now a non-probabilistic binary linear classifier as a result of this. The objective of an SVM model is to generate as big of a gap as possible between the instances of the different categories by mapping the examples as points in space. Next, by mapping them into the same region and identifying which side of the gap they fall into, new instances are projected to fit into a particular category. SVMs can perform non-linear classification as well as linear classification by implicitly translating their inputs into feature spaces with many dimensions, a method known as the "kernel trick."

The primary objective of SVM in cloud resource demand prediction is to find a hyperplane that best separates data points belonging to different classes. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. The hyperplane serves as the decision boundary that separates data points into different classes. The margin in SVM is the distance between the hyperplane and the nearest data points from each class (support vectors). SVM aims to maximize this margin. Support vectors are the data points that lie closest to the decision boundary. They play a crucial role in defining the optimal hyperplane. Train an SVM model in this work, using correlation features as input and cloud resource prediction value as the target variable. Use the trained SVM model to predict cloud resource prediction value for new, unseen data.

The work flow of SVM for cloud resource prediction is shown in Fig. 3.

Input:
- *D*, a data set consisting of correlation features and their associated target values;
- *l*, the learning rate;

Output: Cloud resource demand predicted value

Methods:

Initialize all weights and biases in *network*;
while terminating condition is not satisfied {
for each training tuple *X* in *D* {
// Propagate the inputs :
for each input unit *j* {
$Oj = Ij$; // output of an input unit is its actual input value
for each hidden or output unit *j* {
$I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit *j* with respect to the previous , *i*
$O_j = \frac{1}{1+e^{-I}j}$;} // compute the output of each unit *j*
// Calculate the errors:
for each unit *j* in the output
$Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error
for each unit *j* in the hidden unit , from the last to the first hidden unit
$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher unit , *k*
for each weight *wi j* in *network* {
$\Delta w_{ij} = (l) Err_j O_i$; // weight increment
$w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
for each bias θ_j in *network* {
$\Delta \theta_j = (l) Err_j$; // bias increment
$\theta_j = \theta_j + \Delta \theta_j$; } // bias update
} }

Fig. 3. SVM algorithm.

IV. RESULTS

The dataset used for designing and developing the cloud resource demand prediction using SVM comprises information on resource utilization over a period of one month. In total, there were 123 million observed instances involving 1250 machines. The dataset which is used in this work is Real-time dataset named GoCJ. The GoCJ datasets used in our work is the publicly available dataset collected from https://data.mendeley.com/datasets/b7bp6xhrcd/1. The new proposed approach is implemented in Matlab.

A. Examination Parameters

A wide range of assessment criteria are available to assess the prediction algorithms' performance. This work considers the following elements.

Detection Accuracy is:

$$\frac{True\ Positive(TP)+True\ Negative(TN)}{TP+False\ Positive(FP)+TN+False\ Negative(FN)} \tag{2}$$

Error Rate is:

$$\frac{No\ of\ Datas\ of\ falsely\ labeled\ texts}{Total\ No\ of\ texts} \tag{3}$$

Precision rate:

$$\frac{TP}{TP+FP} \tag{4}$$

Recall rate:

$$\frac{TP}{TP+FP} \tag{5}$$

B. Performance Analysis

Using the performance measures indicated above, the classifier system's performance is analyzed and contrasted with that of other techniques. The tables and graphs below demonstrate this.

1) Experiment No #1: Newly Developed Attribute Selection Approach Evaluation with Accuracy.

We will evaluate the impact of every attribute selection strategy used in the work in this research. Eq. (2) to Eq. (5) are used to evaluate the effectiveness of this cloud resource demand prediction technique. A great attribute selection strategy is preferred to have a high value of Eq. (2). The PCGPSONN accuracy analysis is listed in Table V.

As is evident from Table V, the PCGPSONN has an accuracy of 96.29. The accuracy of attribute selection is shown in Fig. 4.

TABLE V. NEWLY DEVELOPED ATTRIBUTE SELECTION APPROACH EVALUATION WITH ACCURACY

GPSO	GPCNN	PCGPSONN
93.14	94.23	96.29

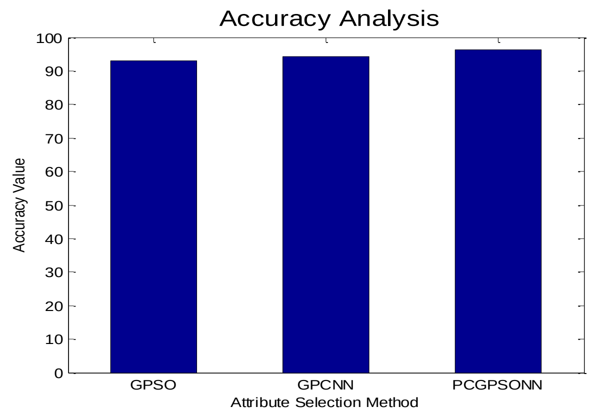

Fig. 4. Accuracy of feature selection strategies.

2) Experiment No #2: Newly Developed Attribute Selection Approach Evaluation with Precision Rate.

Precision tells the proportion of instances that the models predicted as positive and were actually positive out of all the instances it predicted as positive. A high precision indicates that when the model predicts a positive outcome, it is likely to be correct. The PCGPSONN Precision rate analysis is listed in Table VI.

TABLE VI. NEWLY DEVELOPED ATTRIBUTE SELECTION APPROACH EVALUATION WITH PRECISION RATE

GPSO	GPCNN	PCGPSONN
90.21	91.35	93.10

As is evident from Table VI, the PCGPSONN has Precision rate of 93.1, which is better than other approaches. As a result, the PCGPSONN classifier is thought to be the best for choosing attributes. The Precision rate of attribute selection is shown in Fig. 5.

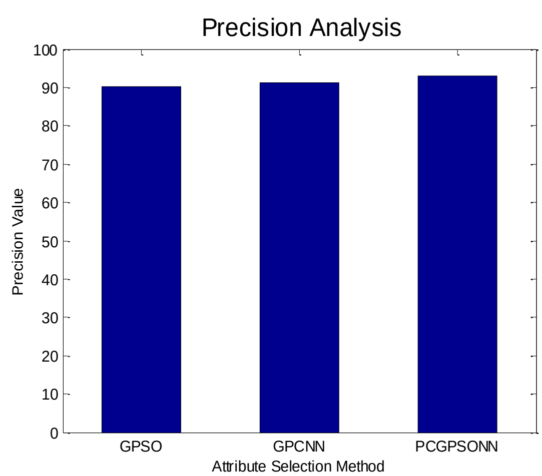

Fig. 5. Precision rate of attribute selection strategies.

3) *Experiment No #3:* Newly Developed Attribute Selection Approach Evaluation with Recall Rate.

Recall depicts the proportion of actual positive instances that were correctly predicted by the model out of all the actual positive instances. A good attribute selection strategy is preferred to have a high value of Eq. (4). The PCGPSONN Recall rate analysis is listed in Table VII.

TABLE VII. NEWLY DEVELOPED ATTRIBUTE SELECTION APPROACH EVALUATION WITH RECALL RATE

GPSO	GPCNN	PCGPSONN
90.11	91.35	92.85

As is evident from Table VII, the PCGPSONN has Recall rate of 93, which is better than other approaches. The Recall rate of attribute selection is shown in Fig. 6.

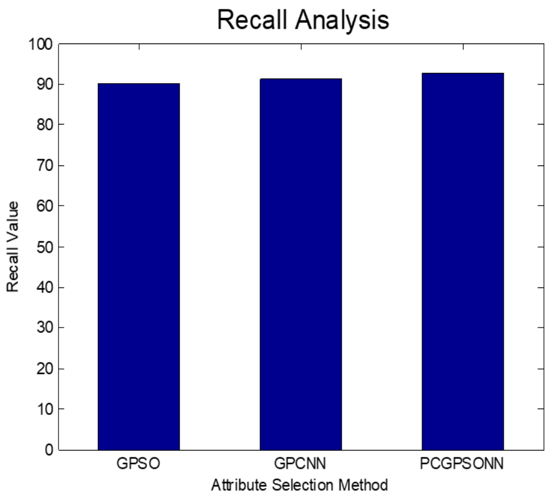

Fig. 6. Recall rate of attribute selection strategies.

As can be seen from the accompanying figure, the PCGPSONN's Recall Rate is better than other methods. Therefore, the PCGPSONN is the best for choosing attributes.

4) *Experiment No #4:* Newly Developed Attribute Selection Approach Evaluation with Error Rate.

Error rate, in the context of machine learning, is a metric that represents the proportion of incorrectly classified instances in a model's predictions. It is the complement of accuracy. The PCGPSONN error rate analysis is listed in Table VIII.

TABLE VIII. NEWLY DEVELOPED ATTRIBUTE SELECTION APPROACH EVALUATION WITH ERROR RATE

GPSO	GPCNN	PCGPSONN
6.86	5.77	0.03703704

As is evident from Table VIII, the PCGPSONN has an error rate of 0.037. The error rate of attribute selection is shown in Fig. 7.

As can be seen from the associated figure, the PCGPSONN's error rate is lower than other methods. Therefore, the PCGPSONN is the best for choosing attributes.

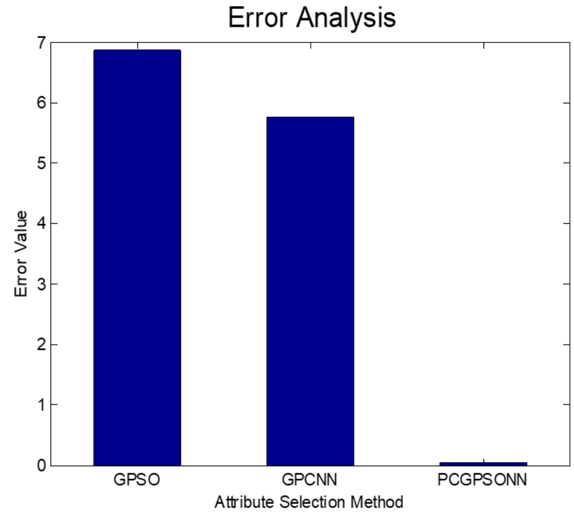

Fig. 7. Error rate of feature selection strategies.

V. CONCLUSION

Demands on cloud resources can be sudden, intense, and volatile. The reactive resource providing approach may result in sluggish or insufficient resource delivery. Thus, to guarantee resource availability, it is imperative to predict resource demands. To process the raw data and produce a fresh and original prediction for Cloud resource demands, machine learning techniques were applied in this study. We were able to develop a more accurate model for cloud resource demand prediction in this study by effectively using a feature selection method based on data mining methodology. PCGPSONN showed to be quite accurate at predicting the demand for Cloud resource. The future direction of this research can be carried out with several machine learning approaches to enhance prediction techniques. Additionally, novel feature selection techniques can be used to develop a deeper comprehension of critical traits and enhance predictions of cloud resource consumption.

REFERENCES

[1] M. Ullrich and J. Lässig. 2013. Current Challenges and Approaches for Resource Demand Estimation in the Cloud. 2013 International Conference on Cloud Computing and Big Data, 2013:387-394.

[2] L. Chen and H. Shen. 2016. Towards Resource-Efficient Cloud Systems: Avoiding Over-Provisioning in Demand-Prediction Based Resource Provisioning. 2016 IEEE International Conference on Big Data, 2016:184-193.

[3] N. Roy, A. Dubey and A. Gokhale. 2011. Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting. 2011 IEEE 4th International Conference on Cloud Computing, 2011:500- 507.

[4] Z. Chen, Y. Zhu, Y. Di and S. Feng. 2015. Self-Adaptive Prediction of Cloud Resource Demands Using Ensemble Model and Subtractive Fuzzy Clustering Based Fuzzy Neural Network. Computational Intelligence and Neuroscience, 2015:1-14.

[5] Y. Jiang, C. Perng, T. Li, et al. 2011. ASAP: A Self-Adaptive Prediction System for Instant Cloud Resource Demand Provisioning. 2011 IEEE 11th International Conference on Data Mining, 2011:1104-1109.

[6] D. Xu, S. Yang and H. Luo. 2015. Research on Generalized Fuzzy Soft Sets Theory based Combined Model for Demanded Cloud Computing Resource Prediction. Chinese Journal of Management Science, 2015, 23(5):56-64.

[7] M. Verma, G. R. Gangadharan, V. Ravi, et al. 2013. Resource Demand Prediction in Multi-tenant Service Clouds. 2013 IEEE International Conference on Cloud Computing in Emerging Marks, 2013, 28(17):1-8.

[8] S. Imai, T. Chestna and C. A. Varela. 2013. Accurate Resource Prediction for Hybrid IaaS Clouds Using Workload-Tailored Elastic Compute Units. 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013:171-178.

[9] H. Mi, H. Wang, G. Yin, D. Shi, Y. Zhou and L. Yuan. 2011. Resource On-Demand Reconfiguration Method for Virtualized Data Centers. Journal of Software, 2011, 22(9):2193-2205.

[10] D. Minarolli and B. Freisleben. 2014. Cross-Correlation Prediction of Resource Demand for Virtual Machine Resource Allocation in Clouds. 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, 2014:119-124.

[11] A. A. Bankole and S. A. Ajila. 2013. Predicting Cloud Resource Provisioning Using Machine Learning Techniques. 2013 26th IEEE Canadian Conference of Electrical and Computer Engineering, 2013:1-4.

[12] K. Rajaram and M. P. Malarvizhi. 2017. Utilization based Prediction Model for Resource Provisioning. 2017 IEEE International Conference on Computer, Communication and Signal Processing, 2017:1-6.

[13] W. Zhang, P. Duan, L. T. Yang, et al. 2017. Resource Requests Prediction in the Cloud Computing Environment with a Deep Belief Network. Software Practice and Experience, 2017, 47(3):473-488.

[14] M. Borkowski, S. Schulte and C. Hochreiner. 2016. Predicting Cloud Resource Utilization. 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing, 2016:37-42.

[15] Mostafa Bozorgi, S., & Yazdani, S. (2019). IWOA: An improved whale optimization algorithm for optimization problems. Journal of Computational Design and Engineering, 6(3), 243-259.

[16] Manikandan, N., Gobalakrishnan, N., & Pradeep, K. (2022). Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment. Computer Communications, 187, 35-44.

[17] Chen, X.; Cheng, L.; Liu, C.; Liu, Q.; Liu, J.; Mao, Y.; Murphy, J. A WOA-based optimization approach for task scheduling in cloud computing systems. IEEE Syst. J. 2020, 14, 3117–3128.

[18] Madni, S.H.H.; Abd Latiff, M.S.; Abdullahi, M.; Abdulhamid, S.M.; Usman, M.J. Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PLoS ONE 2017, 12, e0176321.

[19] Bezdan, T.; Zivkovic, M.; Bacanin, N.; Strumberger, I.; Tuba, E.; Tuba, M. Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. IFS 2021, 42, 411–423.

[20] Sukhoroslov, O., Nazarenko, A., & Aleksandrov, R. (2019). An experimental study of scheduling algorithms for many-task applications. The Journal of Supercomputing, 75, 7857-7871.

[21] Buang, N.; Hanawi, S.A.; Mohamed, H.; Jenal, R. B-Spline Curve Modelling Based on Nature Inspired Algorithms. APJITM 2016.

[22] Kumar, M., Sharma, S. C., Goel, A., & Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. Journal of Network and Computer Applications, 143, 1-33.

[23] Chhabra, A., Huang, K. C., Bacanin, N., & Rashid, T. A. (2022). Optimizing bag-of-tasks scheduling on cloud data centers using hybrid swarm-intelligence meta-heuristic. The Journal of Supercomputing, 1-63.

[24] Ni, L., Sun, X., Li, X., & Zhang, J. (2021). GCWOAS2: multi objective task scheduling strategy based on Gaussian cloud-whale optimization in cloud computing. Computational Intelligence and Neuroscience, 2021, 1-17.

[25] Manikandan, N., Gobalakrishnan, N., & Pradeep, K. (2022). Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment. Computer Communications, 187, 35-44.

[26] Chen, X.; Cheng, L.; Liu, C.; Liu, Q.; Liu, J.; Mao, Y.; Murphy, J. A WOA-based optimization approach for task scheduling in cloud computing systems. IEEE Syst. J. 2020, 14, 3117–3128.

[27] Jia, L., Li, K., & Shi, X. (2021). Cloud computing task scheduling model based on improved whale optimization algorithm. Wireless Communications and Mobile Computing, 2021, 1-13.

[28] Albert, P.; Nanjappan, M. WHOA: Hybrid based task scheduling in cloud computing environment. Wireless. Pers. Communication. 2021, 121, 2327–2345.

[29] Sharma, M.; Garg, R. Energy-aware whale-optimized task scheduler in cloud computing. In Proceedings of the 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 7–8 December 2017; pp. 121–126.

[30] Sreenu, K.; Sreelatha, M. W-Scheduler: Whale optimization for task scheduling in cloud computing. Cluster Computing. 2019, 22, 1087–1098.

[31] Rekha, P.M.; Dakshayini, M. Efficient task allocation approach using genetic algorithm for cloud environment. Cluster Computing. 2019, 22, 1241–1251.

[32] Sun, Y., Li, J., Fu, X., Wang, H., & Li, H. (2020). Application research based on improved genetic algorithm in cloud task scheduling. Journal of Intelligent & Fuzzy Systems, 38(1), 239-246.

[33] Abdullahi, M., & Ngadi, M. A. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Generation Computer Systems, 56, 640-650.

[34] Li, G.; Wu, Z. Ant Colony Optimization Task Scheduling Algorithm for SWIM Based on Load Balancing. Future Internet 2019, 11, 90.

[35] Huang, X.; Li, C.; Chen, H.; An, D. Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies. Cluster. Computing. 2020, 23, 1137–1147.

[36] Nabi, S.; Ahmad, M.; Ibrahim, M.; Hamam, H. AdPSO: Adaptive pso-based task scheduling approach for cloud computing. Sensors 2022, 22, 920.

[37] Chen, X.; Long, D. Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. Cluster. Computing. 2019, 22, 2761–2769.

[38] Kumar, M., & Sharma, S. C. (2018). PSO-COGENT: Cost and energy efficient scheduling in cloud environment with deadline constraint. Sustainable Computing: Informatics and Systems, 19, 147-164.

[39] Zhou, Z.; Li, F.; Abawajy, J.H.; Gao, C. Improved PSO Algorithm Integrated with Opposition-Based Learning and Tentative Perception in Networked Data Centers. IEEE Access 2020, 8, 55872–55880.

[40] Madni, S. H. H., Latiff, M. S. A., Ali, J., & Abdulhamid, S. I. M. (2019). Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. Arabian Journal for Science and Engineering, 44, 3585-3602.

[41] Madni, S.H.H.; Abd Latiff, M.S.; Abdulhamid, S.M.; Ali, J. Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Cluster Computing. 2019, 22, 301–334.

[42] Chhabra, A.; Singh, G.; Kahlon, K.S. Multi-criteria HPC task scheduling on IaaS cloud infrastructures using meta-heuristics. Cluster. Computing 2021, 24, 885–918.

[43] Natesha, B.V.; Kumar Sharma, N.; Domanal, S.; Reddy Guddeti, R.M. GWOTS: Grey Wolf Optimization Based Task Scheduling at the Green Cloud Data Center. In Proceedings of the 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 12–14 September 2018; pp. 181–187.

[44] Alzaqebah, A.; Al-Sayyed, R.; Masadeh, R. Task Scheduling based on Modified Grey Wolf Optimizer in Cloud Computing Environment. In Proceedings of the 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6.

[45] Natesan, G.; Chokkalingam, A. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. ICT Express 2019, 5, 110–114.

[46] Elaziz, M.A.; Xiong, S.; Jayasena, K.P.N.; Li, L. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl.-Based Syst. 2019, 169, 39–52.

[47] Srichandan, S., Kumar, T. A., & Bibhudatta, S. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. Future Computing and Informatics Journal, 3(2), 210-230.

[48] Nasr, A.A.; Chronopoulos, A.T.; El-Bahnasawy, N.A.; Attiya, G.; El-Sayed, A. A novel water pressure change optimization technique for solving scheduling problem in cloud computing. Cluster. Computing. 2019, 22, 601–617.

[49] Praveen, S.P.; Rao, K.T.; Janakiramaiah, B. Effective Allocation of Resources and Task Scheduling in Cloud Environment using Social Group Optimization. Arab. J. Sci. Eng. 2018, 43, 4265–4272.