Research Article

A Novel Resource Productivity Based on Granular Neural Network in Cloud Computing

Farnaz Mahan ¹, **Seyyed Meysam Rozekhani** ¹ and **Witold Pedrycz**²

¹Computer Science Department, University of Tabriz, Tabriz, Iran

²Electrical and Computer Engineering Department, University of Alberta, Edmonton, Canada

Correspondence should be addressed to Farnaz Mahan; mahan@tabrizu.ac.ir

Received 3 February 2021; Revised 24 March 2021; Accepted 25 April 2021; Published 17 May 2021

Academic Editor: Ning Cai

Copyright © 2021 Farnaz Mahan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, due to the growing demand for computational resources, particularly in cloud computing systems, the data centers' energy consumption is continually increasing, which directly causes price rise and reductions of resources' productivity. Although many energy-aware approaches attempt to minimize the consumption of energy, they cannot minimize the violation of service-level agreements at the same time. In this paper, we propose a method using a granular neural network, which is used to model data processing. This method identifies the physical hosts' workloads before the overflow and can improve energy consumption while also reducing violation of service-level agreements. Unlike the other techniques that use a single criterion, namely, worked on the basis of the history of using the processor, we simultaneously use all the productivity rates criteria, that is, processor productivity rates, main memory, and bandwidth. Extensive real-world simulations using the CloudSim simulator show the high efficiency of the proposed algorithm.

1. Introduction

Cloud computing has become one of the most essential tools in recent years for delivering advanced service demand through the public Internet. National Institute of Standards and Technology describes cloud computing as: it is a model for providing easy access according to user demand through the network to a set of changeable and configurable computing resources like Networks, Servers, Storage space, applications, and services in a way that requires minimum resource management and without the direct involvement of the service provider. These computations are presented as a model beyond a system currently capable of responding to most requests and requirements. The flexible computing infrastructure has provided new opportunities to support business activities [1].

Services provided by cloud providers include the platform as a service (PaaS), the Infrastructure as a service (IaaS), and the software as a service (SaaS).

So, the cloud computing platform requires the installation of infrastructure, software, and platform to remove a

specific application. The cloud computing environment offers many benefits for end-users and customers. Here are some of these features: (1) Optimal and efficient use of computing resources. (2) Low energy costs. (3) Remote access to services and resources.

Since cloud computing as a successful business area is developing, using this area is desirable for the enterprises and organizations too; the volume of presented services is very high so the volume of related data is increasing too. Inattention to the high speed of advance and the high volume of demand for using of resources, also because of high maintenance costs and limited resources of cloud computing, the resources management, and control approach has become an essential and important challenge in this area.

1.1. The Problem. As the number of global Internet users has grown, so too has demand for data center services, giving rise to concerns about growing data center energy use. Between 2010 and 2018, for global IP traffic, the quantity of

data traversing the Internet increased more than tenfold. Over the same period, the number of compute instances running on the world's servers, measured total applications hosted, increased more than sixfold [2].

On average, cloud data centers consume 1.3% of the world's total electricity, which is estimated to be about 8% by 2020. If essential steps are taken, cloud data center energy consumption can be minimized from the predicted worst case of 8000 TWh to 1200 TWh by 2030 [3].

Data gathered from more than 5,000 servers over six months indicates that although servers have been mostly idle, productivity rarely reaches 100% [4]. Servers often use between 10% and 50% of their total capacity, resulting in high overhead costs and higher total costs. Therefore, servers will become very inefficient when it comes to their productivity.

Comprehensive studies on monitoring the availability of cloud resources have shown that there are significant opportunities to address these inefficiencies. One way to address inefficiency is to exploit the capabilities of virtualization technology [5].

Virtualization technology allows cloud providers to build multiple virtual machine instances on a single physical server, thereby improving resource utilization and increasing capital return. The reduction in power consumption can also be achieved by changing the inactive nodes and turning them off. Also, live migration of virtual machines [6] can dynamically respond to current resource requirements with the least number of physically active nodes.

1.2. Objective. Many researchers are exploring and developing solutions for managing and allocating resources efficiently in the cloud environment. Given the scale of the research and the complexity of the issue, no generally accepted standard, such as the Internet standard, has been provided for resource management.

In this study, a method has been proposed to predict the workload of overcrowded physical hosts before the overflow occurs. In the proposed model, all the operations discussed are aggregated in the form of clustering components, neural networks, and network training.

This scheme is based on a statistical analysis of the hosts' CPU, RAM, and BW utilization history. The central contributions of this paper are the following:

- (i) We use a granular neural network to predict high-load hosts; unlike the other methods that use a single criterion, namely, worked the base on the history of using the processor, we simultaneously use all the productivity rates criteria, that is, processor productivity rates, main memory, and bandwidth.
- (ii) An adaptive heuristic algorithm is proposed to predict overloaded hosts. This algorithm aims to minimize total electric energy consumption and avoid Service-Level Agreement (SLA) violations.
- (iii) The performance and effectiveness of our algorithm are evaluated using the CloudSim simulator on real

workload traces and compared with other proposed approaches in the literature.

- (iv) The method forms a general approach that can be applied not only to the problem of allocation of resources but also to solve the challenges and managing the uncertainty. By learning and generating rules from training data, we no longer need to reprocess input data and change existing policies. The algorithm is adaptable and flexible at any time.

1.3. Organization. The paper is organized as follows: Section 2 covers related studies on resource productivity. In Section 3, we describe the basic concepts. In Section 4, we present a thorough analysis of the proposed method. In Section 5, we evaluate our approach. Finally, conclusions are covered in Section 6.

2. Related Works

In this section, related literature on energy efficiency and resource productivity in the cloud data center is briefly reviewed. The review is divided into two categories, including Energy-Aware-based approaches and Forecast-based approaches.

2.1. Forecast-Based Approaches. In 2013, Frankian et al. proposed a method for deciding on the migration of virtual machines to achieve greater cloud resources efficiency [7]. In their proposed method, the CPU load was predicted by linear regression. The authors of this paper used the amount of processor load to detect overload and undercarriage of physical machines. Since the linear regression method is a history-based method, the CPU load rate was recorded over ten-minute intervals during the initial hours.

In the presented algorithm, at first, the value of CPU load is estimated; then, if the usage of CPU is more than the threshold value, this physical machine is sent to the center of virtual machine management as the overloaded machine. Therefore, some of the existing virtual machines are migrated to physical machines with low load.

In 2015, Frankian et al. presented a method to improve the consolidation of virtual machines using an ant colony detection algorithm [8]. In this method, after detecting CPU overloading by using linear regression, the ant colony method, which is a method of collecting ant behavior for food have been used, for identifying the best alternative to virtual machines and have found the optimal solution. They tried to find the optimal solution. The proposed technique can significantly reduce energy consumption and SLA violation rates. But, only CPU utilization based on the history of usage in each host is considered.

In 2016, Zuo et al. worked to improve resource productivity [9]. The authors of this study aimed to estimate the number of virtual machines to improve server performance and reliability. In the proposed algorithm, if the estimated number of virtual machines exceeds the required number, most of the server space will be lost, and the so-called pre-activated state will result in additional costs. If this estimate is less than the required number, the customer request will

not be completed in the expected time, and the so-called inactive mode and reliability will decrease. The algorithm seeks to balance the two states by using bioscience to estimate the number of virtual machines needed by adapting the genome model and cloud computing. The basic algorithm used here is the GRASP heuristic algorithm. In the proposed algorithm, the static dimensions are dynamically transformed. This method is composed of a monitoring component and a GRASP-based procedure that estimates the number of VMs based on two criteria: total execution time and financial cost of the workflow execution.

In 2014, Yank proposed a method based on machine learning techniques [10]. Most approaches based on machine learning techniques require training. The authors proposed a linear regression-based approach to predict the number of requests per cloud service. In the proposed algorithm, due to the work fluctuations, the prediction method is adjusted through optimizing regression parameters. In general, the prediction accuracy of these methods is based on the behavioral similarities of the program in training courses and test stages. If the software behavior is not correlated to the test phase and the training phase, the predicted results are not reliable, and the training phase must be repeated so that the model can be adapted to the work dynamics. In this paper, based on this predicted workload, an auto-scaling mechanism is proposed to scale virtual resources at different resource levels in service clouds. The auto-scaling mechanism combines real-time scaling and pre-scaling. Finally, experimental results show that the approach can satisfy the user Service Level Agreement (SLA) while keeping scaling costs low.

In 2017, Amiri et al. proposed a POSITING extraction-based forecasting model to improve the provisioning of cloud resources, which examines the relationship between different sources and identifies behavioral patterns independent of fixed pattern length explicit extracts [11]. The proposed algorithm analyzes program behavior in the past, extracts behavior patterns, and then stores them based on the offline model.

In fact, based on the extracted patterns and recent behavior of the program, future demand predicts different sources. The POSITING algorithm explicitly reduces the shortcomings of earlier predictions by extracting patterns of different lengths. However, software behavior can change due to the dynamic nature of the cloud. Therefore, since offline statistical patterns are derived from previous program behavior, they may not be useful for predicting current behavior. This means that the proposed method is unable to adapt to the changes. On the other hand, the computational resource consumption of the expected model should be reasonable. Therefore, saving all the data and then processing them is impractical.

In 2018, Amiri et al. sought to improve POSITING online learning capability after improving their proposed algorithm [12]. To measure the power of offline patterns to predict the current behavior of the program, they defined a conformance criterion. RELENTING compares the expected results of POSITING with the observed behavior and updates the criterion for matching offline patterns. If there is no

reliable nonlinear pattern for prediction, based on recent and current behavior observed from the program, the online engine will extract new behavior patterns and store them in the online pattern database. Both online and offline pattern databases are used to predict program behavior. Thus, in addition to relying on new behavioral patterns, in-line patterns are recognized and used for prediction.

2.2. Energy-Aware-Based Approaches. In 2010, Blagazov et al. paid particular attention to the response time and quality of service. They therefore attempted to keep the CPU efficiency at a certain level by assuming several threshold levels to guarantee quality. At the same time, efforts were made to optimize energy consumption [13]. The main idea is to set higher and lower deployment thresholds for the host and maintain better CPU performance by all virtual machines dedicated to the host by these thresholds. If a host is lower than the threshold, all virtual machines are moved from that host, and the host is switched to sleep mode to eliminate unused power consumption. If the deployment is too high, some virtual machines need to be moved from the host to reduce performance. The goal is to maintain free resources to prevent SLA violations.

In 2010, Blagazov and Buyya in a follow-up to previous papers attempted to apply some dynamic threshold in another article [14]. They believe that fixed-value thresholds are not suitable for environments with dynamic and unpredictable workloads and with multiple applications occupying a physical resource. The system must be capable of automatically adapting its behavior to patterns for automatic adjustment of values. This paper presents a threshold based on a statistical analysis of historical data collected during the life of a virtual machine.

In 2011, Blagazov and Buyya completed a review of previous work [15]. The difficulty of finding the best time to migrate virtual machines to minimize the cost of energy consumption and prevent offenses imposed by cloud servers due to SLA violations have been examined. They have also compared the behavior of Optimal Online Algorithms and Optimal Offline Algorithms. To find the dynamic threshold, they also suggested using two methods, namely, MAD and IQR. The Median Absolute Deviation is a value for statistical dispersion. This value is a robust estimator of the scale with respect to variance or standard deviation and is more robust to pertinent data in the dataset. For a univariate dataset, the mean absolute deviation is defined as the mean of the absolute deviation from the median. The Inter Quartile Range is one of the scattering measures equal to the distance between the first and third quartiles. In order to eliminate the problem of affecting the range of variations of large and small values, the distance between the largest and the smallest values can be calculated on the basis of quarts. Thus, for data with discarded values, the distance between the first and third quartiles can give a better estimate of the maximum data dispersion calculation.

In 2016, Zhao et al. addressed reducing energy consumption and minimizing service-level violations in cloud data centers [16]. In their view, the issue of high energy

consumption due to the construction of large-scale cloud data centers is becoming more and more severe over time. To reduce the energy consumption and nonviolation of SLA, an optimal placement algorithm for new virtual machines is called ATEA (Three Energy Adaptive Threshold Technique). At ATEA, concerning the shifted load, the data center hosts are divided into four groups: low-load host, light-load host, medium-load host, and heavy-load host. ATEA drives virtual machines from overloaded, low-cost hosts, while virtual machines remain unchanged in light-medium hosts. Based on the proposed algorithm, two types of KAM and KAI, three matching threshold algorithms, and three types of virtual machine selection policies are offered.

In 2017, Yadav et al. [17] have proposed a virtual machine selection algorithm that selects a virtual machine from overloaded or underloaded hosts. Optimization of the energy and minimizing the SLA violation are the two main goals of this paper. To achieve these goals, in this paper, an approach called MuMs has been proposed to select virtual machines. In addition to reducing SLAs violation this method also increases Qos.

In 2017, Yadav and Zhang proposed an algorithm to solve the issues of energy consumption of mobile cloud data centers. In this paper, to detect overload hosts, an algorithm called MeReg has been presented. This algorithm is based on a regression model called M-estimator, which dynamically estimates the high threshold of CPU usage based on the set of CPU usage history. For more details about this algorithm, see [18]. According to the results presented in this paper, it can be concluded that MeReg could decrease electric energy consumption and increase QoS.

In 2018, Yadav et al. [19] proposed three models for minimizing energy consumption and SLA violation. Gradient-descent-based regression (Gdr) and Maximize Correlation Percentage (MCP) are two adaptive models that have been used for detecting overloaded hosts. In this paper, in the stage of virtual machine selection from overloaded hosts, a bandwidth-aware selection algorithm considers the network traffic from the overloaded host under SLAs, and then selects a VM. According to the results presented in this paper, it can be concluded that the Gdr algorithm improves energy consumption better than the MCP algorithm. Network traffic is also an effective factor in selecting virtual machines from the list of overloaded hosts.

In 2020, Yadav et al. [20] proposed an adaptive heuristic algorithm to detect overloaded hosts and virtual machine selection from overloaded hosts. In this paper, the applications running in the VM and the CPU utilization at different times have been considered by the VM selection algorithm. According to the results presented in this paper, it can be concluded that the proposed algorithm has minimized electric energy consumption of cloud data centers (CDCs).

In 2021, Yadav et al. [21] proposed an algorithm called GradCent, which is based on Stochastic Gradient Descent technique. This algorithm creates an upper CPU usage threshold so that it can detect overloaded hosts by the actual CPU workload. This paper has also proposed an algorithm called Minimum Size Utilization (MSU) for selecting virtual

machines dynamically after detection of overloaded hosts. These two algorithms are implemented in the CloudSim simulator. Using of GradCent and MSU together decreases the electric energy consumption of cloud data center. In addition, these algorithms minimize SLA violations.

3. Basic Concepts

3.1. Granular Computing. Granular computing (GrC), as a new concept and computing model of information processing, covers all details of granulation of theories, methods, techniques, tools, and is mainly used to deal with uncertain, fuzzy, and incomplete [22].

The term granular computing first appeared in a 1979 paper [23], which was defined as a subset of computing with words. Specifically, Zadeh proposed a fuzzy-based general framework for granular computation. Granules are constructed and defined according to the concept of general constraints and the relationship between the granules as fuzzy diagrams or fuzzy numerical rules.

Yao believed that granular computing is a robust view of human understanding and problem-solving that may be more important than any specific and fundamental method. [24] He argued that this general view could be achieved by looking at three levels or three attainable perspectives:

- (i) At a philosophical level, granular computing is seen as a way of structured thinking
- (ii) At a practical level, granular computing is seen as a general method of structured problem-solving
- (iii) At a computational level, granular computing is considered an example of information processing

The idea of granular computing in artificial intelligence has been explored through the concepts of aggregation and abstraction. Hobbes proposed his theory in 1985, based on the observation that, “we look at the world under different granular conditions and only consider those that serve our current interests.” Also, “our ability to conceive of the world at different levels and the change between being this seed is fundamental to our intelligence and flexibility. This allows us to map the complexities of the world around us into simpler theories. Calculations are more plausible” [25].

Information awareness is an intrinsic and pervasive activity that humans do intending to understand things better. Specifically, granular information is manageable and accessible to divide complex issues into several parts. As such, significant complexity problems are mapped to a series of more solvable issues [26].

3.2. Granular Computing Feature. What has been said so far relates to the qualitative aspects of the issue. Granular computing has essential features as well [27, 28]:

- (i) Discovering the fundamental similarities between different contexts and issues and creating a framework called Granular World
- (ii) Understanding existing connections between similarities using the granular processing framework

- (iii) Creating sophisticated models for processing granular information using the specified framework
- (iv) Including a wide range of variables, including numeric, abstract, and generic entities
- (v) Creating harmony between different issues despite noticeable differences
- (vi) Reducing the cost and time complexity of solving problems to find the optimal approximate solution

3.3. Granule and Granular Structures. An elementary concept of granular computation is the granule, which represents part of the whole. Granules are our focal points or units used to get a description or a representation. A granule has two distinct roles:

- (1) It may be a member of another seed and be considered part of its constituent
- (2) It may also include a family of granules and constitute a whole

The specific role of each granule is determined by our focus points at different stages of problem-solving. A family of granules of the same type may be collected, and their aggregate characteristics studied. This leads to an understanding of the concept of surfaces. A surface is occupied by granules of similar size or nature. While each granule may provide a particular view of the problem, it provides a level of overview of the problem.

An essential feature of the grains and surfaces is their granularity. The granularity of the grains and surfaces allows us to build a hierarchical structure called a hierarchy. The hierarchical term denotes a family of partially ordered granular surfaces in which each surface comprises a family of related and interacting granules.

While a single hierarchy provides a representation at many granularity levels, multiple orders can have a full understanding of many perspectives. Therefore, granular structures are hierarchical families [29].

3.4. Granular Model. The GRC model is interpreted as a set of theories, methods, tools, and techniques that use grains (classes, groups, and clusters of reference sets) to solve the problem. In this method, the similarity or inseparability of objects is used to form information granules and derive classification rules. In the GRC model, all information in the problem is sorted into a set of objects with a certain number of attributes, such that one of the attributes specifies the class or output values. The GRC model algorithm, using a sorted data set called the data table, actually extracts the natural rules existing between the data.

The information table in the granular model describes several objects or patterns. Each row and column in this table assigns an object and an attribute to the object so that each object is described with these attributes [30].

The information table is displayed as the following relation:

$$S = \{U, At, L, \{Va|a \in At\}, \{f_a|a \in At\}\}. \quad (1)$$

In this relationship, U is a nonempty set of objects, At is a nonempty set of attributes, L is a spoken language of attributes At , Va is a set of values for At and f_a attributes such that for each $a \in At$ and $X \in U$ is the information function $F(x, a) \in V_a$. An information table, as shown in Table 1, is adopted from Quinlan [31].

In this model, the rules are expressed as if-then in such a way that if an object has a set of values of the information table attributes in the concept form φ and yet belongs to a class Ψ or a certain value of the decision variable, the relation of the two to The form “If an object fits into the concept of φ , then it will have a value of Ψ of the decision variable” is expressed.

For example
ifhair = blond & eyes = blue $\longrightarrow$ class = +.

To derive data rules, several measurements are performed with the GRC model on objects, concepts, and rules. The concepts of generality, absolute support, and coverage are the parameters used in the GRC model discussed below. These parameters are used to extract and prioritize the rules. As such, laws with the highest values of generality, absolute support, and coverage are considered with the least inconsistency for future forecasts [30].

3.4.1. Generality. The generality of a concept such as φ , which represents the relative size of the constituent grain φ , is defined by a (2) relation, which indicates the extent to which the concept is pervasive among the available data.

$$G(\varphi) = \frac{|m(\varphi)|}{|u|}. \quad (2)$$

In this respect, $m(\varphi)$ denotes the constituent φ and $|u|$ The grain size constitutes the reference sum.

3.4.2. Absolute Support. Absolute support denotes the absolute correctness of the relation or law of $\varphi \longrightarrow \Psi$ and is defined as the conditional probability of $\varphi \longrightarrow \Psi$ in the following way.

$$AS(\psi|\varphi) = \frac{|m(\varphi \cap \Psi)|}{|m(\varphi)|}, \quad (3)$$

$m(\varphi \cap \Psi)$ represents the grain size whose constituent objects apply to both φ and Ψ .

3.4.3. Cover. The small amount of coverage φ by Ψ indicates the probability of recalling the law and the accuracy of the law. The probability that if the randomly selected object is set to Ψ also applies to φ . This value is calculated from the relation (4).

TABLE 1: An information table.

Object	Height	Hair	Eyes	Class
o1	Short	Blond	Blue	+
o2	Short	Blond	brown	−
o3	Tall	Red	Blue	+
o4	Tall	Dark	Blue	−
o5	Tall	Dark	Blue	−
o6	Tall	Blond	Blue	+
o7	Tall	Dark	brown	−
o8	Short	Blond	brown	−

$$CV(\varphi|\Psi) = \frac{|m(\varphi \cap \Psi)|}{|m(\Psi)|}. \quad (4)$$

3.4.4. Conditional Incompatibility. The conditional incompatibility $H(\Psi|\varphi)$ is obtained from the relationship (5), and its reasons include a large amount of information, insufficient knowledge about the data, and inconsistency in the views presented by the researchers.

$$H(\Psi|\varphi) = -\sum_{i=1}^n p(\Psi|\varphi) \log p(\Psi_i|\varphi). \quad (5)$$

In (5), $P(\Psi|\varphi)$ denotes the probability of the formula being right for class Ψ_i .

3.5. Granular Neural Network. The first comprehensive study of granular neural networks by Pedrycz was around the year 2000 [26]. This paper first introduces the concept of information granulation, which is the basis of granular computing. Then, the process of granular neural network development is examined. It consists of two primary phases: First, data are summarized in the form of information grains. Second, the information within the seeds is subsequently understood and used by the neural network for all educational purposes.

The neural network is a type of network system that simulates the information-processing mechanism of the human brain based on the development of modern biology research, one of the soft computing techniques. Neural networks are used in various fields such as knowledge discovery, pattern recognition, adaptive learning, and so on. But there are disadvantages to these advantages, such as getting stuck in local optimizations, long training time, difficulty accessing complex information, increasing computational complexity with the increased workload, and so on. These problems mainly limit the development of the neural network. To overcome these shortcomings, many optimization algorithms have been used to learn neural network weight and automated network structure design, such as various evolutionary optimization algorithms. These optimization algorithms are feasible and efficient but will

take a long time to process extensive, complex data. The neural network has good reliability and learning capability, but for large, complex, and high-dimensional data, it cannot work well. This network requires a lot of time and has a high time complexity for large datasets. GrC theories and neural networks both have their own advantages in dealing with problems, both of which are complementary. Combining them can solve complex problems with large dimensions [32].

The granular neural network is a nervous system capable of processing granular data. The granules form the input and output of the neural network. In other words, the neural network that is the input and output of the granules is called the granular neural network. The concept of “granules” here refers to different data sets aggregated by similarity or function criteria. The granular neural network used in this research is a versatile and learning network that is quite similar in function to the fuzzy inference system (ANFIS), and its inputs are clusters of data (input granules). Its outputs are fuzzy rules (output granules). This network has five layers except for inputs and outputs. The first layer performs the fuzzy operation. In the second layer, the output of each node determines the firepower of each rule. The third layer performs the normalization operation. The final part of each fuzzy rule, in the fourth layer, and the total output of the system in the fifth layer, both of them are calculated from the sum of the fourth layer outputs. The simple structure of this network is shown in Figure 1.

Layer 1: in this layer, entries pass the membership function.

$$\begin{aligned} o_{1,i} &= \mu A_i(x), \quad \text{for } i = 1, 2, \\ o_{1,i} &= \mu B_i(x), \quad \text{for } i = 3, 4. \end{aligned} \quad (6)$$

Membership functions can be any suitable function with some parametric representation. In most cases, the Gaussian function is chosen.

Layer 2: The output of this layer is the multiplication of the input signals, which is equivalent to the part of the rules.

$$o_{2,i} = w_i = \mu A_i(x) \mu B_i(y), \quad \text{for } i = 1, 2. \quad (7)$$

Layer 3: the output of this layer is a normalized version of the previous layer.

$$o_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad \text{for } i = 1, 2. \quad (8)$$

Layer 4: this layer is calculated as

$$o_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad (9)$$

where $\bar{w}$ is the output of Layer 3, and p_i , q_i , and r_i are

FIGURE 1: A simple example of the structure with two input variables.

the consequent parameters.

Layer 5: the total output of the system is calculated as

$$o_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \text{for } i = 1, 2. \quad (10)$$

3.6. Energy-Conscious Model. In this model for managing virtual machines, one has to answer the following four crucial questions:

- (1) Which virtual machine should be used for which host?
- (2) Which host is overloaded?
- (3) Which virtual machine from the overflow host is appropriate for selecting and migrating?
- (4) Which host is underweight?

There are four phases to answer the above questions.

- (1) The following is the first issue of virtual machine deployment: In this sub-issue, it is considered to select a suitable host for each virtual machine in such a way that the sum of resources required by the virtual machines allocated to a host exceeds that of the host and also the minimum amount of Use hosts.
- (2) The second subproblem of detecting physical hosts that have overflowed: The critical aspect at this point is to make this determination before the overflow occurs, and it is not a welcome issue. Hence, it is essential to know when a host overflows. If this is announced too early, it will prevent the optimal use of host resources, and if too late, it will cause an overflow and error in the computation process.
- (3) The third subproblem of choosing a virtual machine is overflow from the host: It is also essential to reduce the volume of the virtual machine from the host. Selecting a virtual machine that uses many host resources will result in lower productivity. It will result in related host performance decrease extremely, and on the other hand, a high migration cost

will be forced to the data center. Choosing to let the host fall into a risky state requires making sequential choices and having many migrations that will be both time-consuming and costly. Therefore, keeping this balance is essential.

The following can be considered as problems with the complexity of class P: Generally speaking, the P class is a problem with fast algorithms to find the solution. That is why most of the methods presented for this subproblem are detailed guidelines. The only difference is the solutions offered by their policies for removing the physical host from overflow.

- (4) The fourth subproblem of choosing low-load physical hosts: This step is also essential for the second step, only with less sensitivity when a low-host is introduced. It is essential that if the threshold is set, we may fail to respond to clear servers, and we have to turn off the servers. In this case, we have to turn on another server to shut down another server. The critical point is, has the power consumption been reduced to such a degree that this decision is cost-effective?

If this threshold is low, the servers may be turned on, and the available data will be processed by turning them off. So, this balance also needs to be observed.

The Power-Aware Best Fit Decreasing algorithm, or PABFD for short, uses two nested loops which the outer loop scrolls all virtual machines and the inner loop scrolls all physical hosts. Each host is checked, and it is estimated that the energy consumption of that host will increase if the virtual machine is targeted to each host. Therefore, each virtual machine is considered to have the lowest energy consumption.

4. The Methodology

The most important goal in improving the efficiency of cloud resources is to achieve the best possible utilization of resources concerning service-level agreements and improving the expected performance on energy consumption. The

necessity of presenting a new approach, including the use of neural network algorithms and its integration with granular computing, and in general the use of data mining to predict the process of management and resource allocation in cloud environments, as an efficient solution in this environment, is more than ever felt and is also considered as the basis of this research.

The model used in this research is based on the energy-conscious model presented by Anton Beloglazov [15].

In this study, we have tried to identify the productive hosts before overflow using the granular neural network. For this purpose, by the granular neural networks described in the previous section, the rate of processor productivity, main memory, and bandwidth is given to the granular model in the form of granular data and the rate of these productivities shortly is estimated by the model. It is estimated. In this study, training data are first run using a simulator without a method to detect overloaded hosts, and the CPU rate, main memory, and bandwidth of the hosts are recorded. The simulator is run several times to collect a complete set of modifications. After collecting the training data in the first step, the number of granules should be determined. In terms of bandwidth efficiency, the CPU and physical memory of the physical hosts are categorized into three categories of high-load, low-load, and regular hosts. The k-means algorithm is used to perform the classification. Figure 2 shows the general outline of the steps performed in the proposed method.

Initially, all the virtual machines delivered to the mediator are listed in the VM migrate list, which is assigned to physical hosts in the first phase, i.e., the replacement of machines using the PABFD Algorithms 1 and 2. The GNN is applied to identify hosts at the second stage, and it classifies the productivity rate of CPU, RAM, and BW of hosts using determined thresholds. The neural network is trained to be able to adapt to changes. Finally, the rules are extracted, and if the host is predicted to be productive, some VM, with the use of the selection method of the virtual machine, are added to the migrate list to be allocated to a suitable host in the first phase. Also, if the host is considered to have a small workload, all the virtual machines on that host are transferred to the list, and the host is switched off. In the following sections, GNN operation to distinguish hosts is discussed.

In this paper, we aim to solve some issues mentioned earlier. Our proposed method has some advantages that have been shown in Figure 3.

4.1. Formation of Granular Inputs. The efficiency rates of the physical hosts collected are used as training data to train the granular neural network. In the granular grid, if we have N inputs and M is the membership function for each input, the number of fuzzy rules produced is equal to M^N , which is the number of fuzzy rules generated by increasing the number of network inputs' membership functions. It also increases exponentially. This can slow down the speed of network training. Therefore, it is best to input the data selected for network training first into granules (clusters). Since the data inside each granule exhibit the same behavior under different conditions, the number of fuzzy rules created (output

granules) and the number of membership functions of each input depend on the number of primary granules. As a result, the initial granules will significantly influence the overall performance of the model. There are many methods, including FCM, sub-clustering, grid partitioning for aggregating input data. In the proposed work, the k-means clustering process is used. The k-means method may be the most straightforward clustering algorithm, which in many cases is also the best and most practical algorithm. This algorithm is one of those algorithms that must have specified the number of clusters in advance. Since we are dealing with three groups of physical hosts, the number of clusters is already known.

4.2. Granular Neural Network Training. After the granules are formed, determine the number of membership functions and the number of rules that will be created. The granular network must be trained so that the fuzzy laws produced can predict the new data. The type of network input and output membership functions must also be selected before network training. Depending on the kind of problem, you can choose from triangular, trapezoidal, Gaussian, or any other member function. In this study, a trapezoidal type was used for network input. The reason for choosing the trapezoidal membership function lies in its nature. In this type of function, the intermediate function can have a more excellent range, which in our case is equal to the regular host; therefore, being in a normal situation means the best state for hosts. Empirically and by trial and error, hosts below 30% productivity are considered as low-cost hosts. The hosts with a productivity rate more than 80% are high-risk and productive hosts, and accordingly, the range of 50% productivity rate range is related to for regular hosts.

The following relationships illustrate the trapezoidal function used in this study.

$$\begin{aligned} \text{underload}(x) &= \begin{cases} 1 & x \leq 0.20 \\ \frac{0.30 - x}{0.10} & 0.20 < x < 0.30 \end{cases}, \\ \text{normal}(x) &= \begin{cases} \frac{x - 0.20}{0.10} & 0.20 < x \leq 0.30 \\ 1 & 0.30 < x < 0.80 \\ \frac{0.90 - x}{0.10} & 0.80 \leq x < 0.90 \end{cases}, \quad (11) \\ \text{overload}(x) &= \begin{cases} 1 & x \geq 0.90 \\ \frac{x - 0.8}{0.10} & 0.80 < x < 0.90 \end{cases} \end{aligned}$$

The gradient decreasing algorithm is used to the network training process, which is an iterative optimization algorithm, in this algorithm to find the local minimum of the error function, and the steps are taken according to the negativity of the function gradient at the current location.

FIGURE 2: Proposed flowchart.

Result: Host Detection

Let *VMMigrateableList* be the list of Vms;

Call PABFD Algorithm to allocate each Vm to physical hosts.;

Suppose Host be the physical host;

Host = Call Algorithm 4 to construct GNN structure.;

if Host == overloaded then

Apply VM selection policy;

Transfer selected VM to *VMMigrateableList*;

else

if Host == underloaded then

while There exists Vm in the Host do

transfer the Vm to *VMMigrateableList*;

Turn off the Host;

ALGORITHM 1: The proposed algorithm.

```

Result: Host
Suppose  $D$  is the list of physical hosts.;
while there exists a physical host in the  $D$  do
  Extract Productivity rates for each physical host;
  Run k-means algorithm for  $D$ ;
  Find the best clustering based on the rate of CPU, RAM, and BW;
  Set GNN parameter;
  Gradient descent from data;
  Fuzzification of  $D$  using equation (11);
  Discovering fuzzy knowledge (Granules);
return Host;

```

ALGORITHM 2: GNN structure.

The input and output data are as follows:

$$\begin{aligned}
 (X_0^P, Y_0^P) \quad P = 1, 2, 3, \dots, N, \\
 Y_0^P \in V = [\alpha_Y, \beta_Y] \subset R, \\
 X_0^P \in U = [\alpha_l, \beta_l] \times \dots \times [\alpha_n, \beta_n] \subset R^n.
 \end{aligned} \tag{12}$$

The goal here is to design a fuzzy system that can minimize the matching error.

$$e^P = \frac{1}{2} [f(x_0^P - y_0^P)]^2. \tag{13}$$

That is, the goal is to determine parameters y_i^{-1} , x_i^{-1} , and σ_i^{-1} in such a way that e^P has the lowest value.

So the system first starts with defects and errors and then adapts to the network by entering new data and continuing the training process to become a model with low matching errors. The number of network training steps for 400 performances and the target or error tolerance value (the amount of error acceptable to us) for all performances is set to zero. After the training process, the output granules are produced.

4.3. Starting the Migration Process. After completing the load forecasting process, the physical hosts have the following states:

- (1) If the physical host is detected to be low, all virtual machines are removed from that host and migrated to the new host using the PABFD method. And, the host is turned off.
- (2) If the host is detected frequently, by one of the virtual machine selection methods, one or more virtual machines are selected and migrated to the other host, to reduce the hosting burden.

4.4. Migration Methods. Another issue affecting host performance and the energy consumption is the number of migrations performed that should be minimized. If hosting is found to be a low load that does not require a specific policy, all virtual machines need to be ported. But, if a host is found to be overloaded, then according to the policy used, it

FIGURE 3: Main advantages of the proposed method.

will transfer one of the virtual machines and continue this process until it gets out of the situation.

There are three most commonly used methods for migrating virtual machines:

- (i) Minimum Migration Time policy (MMT) means the shortest migration time; it selects a virtual machine for migration that takes less time to transfer than other machines. Estimation of the migration time is obtained by dividing the amount of RAM memory occupied by that virtual machine over the bandwidth available on the network. In Formula V_j virtual machines set on host j , $RAM_u(a)$ is the amount of memory used by virtual machines a , NET_j bandwidth is available to this host.

$$v \in V_j | \forall a \in V_j \cdot \frac{RAM_u(v)}{NET_j} \leq \frac{RAM_u(a)}{NET_j}. \tag{14}$$
- (ii) Random Selection or RS based on the normal distribution of discrete variable X randomly selects and migrates virtual machines. And, it repeats this operation until the productivity drops below the threshold.
- (iii) The Maximum Correlation Policy (MC) means the highest correlation policy; it operates on the correlation of applications on the CPU with the CPU productivity, and makes its choice on the highest correlation.

5. Experimental Studies

In this section, we investigate the performance of the proposed algorithm.

Since the target system is an infrastructure system as a service (IaaS), cloud computing environments must provide users with a view of unlimited computing resources. Evaluation and monitoring of virtual machine scheduling algorithms and methods are required in cloud-scale environments. Testing these algorithms in real environments requires a great deal of time and money. Therefore, the simulation method was used to test the environment to ensure the integrity and repeatability of the experiment.

To implement the algorithm, the Cloud simulator, a popular Toolkit, is used to simulate Claude in Java. To run the Toolkit, we need a platform that we use here as Net Beans.

To accomplish the project goals and evaluate the proposed algorithm, three widely used MAD, IQR, and ATEA methods for the sub-issue “When to do migration” and three commonly used methods RS, MMT, and MC for the sub-issue “Which virtual machine be selected to migration” have been simulated.

5.1. Specifications of Simulation Scenarios. To provide a simulation-based evaluation, considering all tracks of an entire system is essential. In this paper, all experiments used the data provided by a part of the monitoring infrastructure of a project of PlanetLab [33].

We have used the data on CPU utilization by more than a thousand VMs from servers located at more than 500 places around the world. The interval of utilization measurements is 5 minutes. We have randomly chosen ten days from the workload traces collected during March and April 2011. The characteristics of the data for each day are shown in Table 2.

The simulation environment is a data center containing 800 heterogeneous computing nodes or hosts. The physical nodes used are the HP ProLiant G4 and HP ProLiant G5 server models. The hardware specifications of these hosts are described in Table 3.

Physical host power consumption ranges are shown in Table 4, from about 175 watts with zero percent CPU usage to 250 watts with 100 percent CPU usage.

The specification of virtual machines built is a function of processor power. In this experiment, our virtual machines are as shown in Table 5.

The user logs request to supply 1175 heterogeneous virtual machines that simulate the total capacity of the data center. Each test is run ten times. The current results are based on average values.

5.2. Evaluation Criteria. The methods discussed are from four aspects of energy consumption, namely, SLA, ESV, and makespan.

5.2.1. Energy Consumption. Increased energy consumption in large-scale computer systems, such as clouds, has many

economic and environmental consequences. Hence, reducing the energy consumption of data centers is a complex and challenging issue in the design of modern systems. The CPU utilization rate in the physical host

$$\mu_j = \frac{P_j^{w_{cpu}}}{P_j^{cpu}}. \quad (15)$$

Thus, the energy consumption of the physical host can be calculated using relation (15).

$$E(p_j) = k_j \cdot e_j^{\max} + (1 - k_j) \cdot e_j^{\max} \cdot \mu_j. \quad (16)$$

5.2.2. Service-Level Agreement (SLA). Service-Level Agreement is a document that includes a description of the service agreed upon, service-level parameters, warranties and measures, and remedies for all violations. SLA is essential as a contract between the consumer and the provider.

5.2.3. Energy SLA Violation (ESV). The ESV function measures the amount of energy consumption, which decreases as the SLA increases. It is necessary to examine this function in the sense that the reduction of power consumption in this method is achieved by minimizing the number of physical hosts, which in some cases violates the SLAs; so, a metric is needed to determine whether the reduced energy was justified by the increased breach of SLA contracts. ESV is calculated from relation (16).

$$ESV = \text{energy} \left(\frac{kw}{h} \right) * SLA (\%). \quad (17)$$

5.2.4. Makespan Time. Makespan time refers to the time within which all tasks are completed. In some sources, it is also called baking time.

Each method is evaluated in ten different execution rounds, and the average results are visible in the following charts and are the result of the testing data. According to the graphs, it is clear that the proposed method, along with the MC algorithm for virtual machine migration, performs best in all cases compared to other algorithms.

The proposed method was evaluated with regard to four essential and practical aspects, namely, energy consumption, validation of SLA, ESV, and makespan time. Figure 4 shows the results. Figure 4(a) shows the average amount of energy consumed in ten different execution rounds. As can be observed, the proposed method by using the MC algorithm to select the Virtual Machine by 383 watts per hour has the lowest consumption. Figure 4(b) shows that the proposed method using the MC algorithm with a value of 4.42 has the minimum SLA violation. In Figure 4(c), the criterion is to measure the energy consumption rate as the SLA increases. As can be observed, the proposed method was successful. Finally, Figure 4(d) shows the finish time of all tasks. This criterion has been evaluated with ten different execution rounds with three standard migration methods. The

TABLE 2: Workload data characteristics.

Date	Number of VMs	Mean (%)	St. dev. (%)	Quartile 1 (%)	Median (%)	Quartile 3 (%)
03/03/2011	1052	12.31	17.09	2	6	15
06/03/2011	898	11.44	16.83	2	5	13
09/03/2011	1061	10.70	15.57	2	4	13
22/03/2011	1516	9.26	12.78	2	5	12
25/03/2011	1078	10.56	14.14	2	6	14
03/04/2011	1463	12.39	16.55	2	6	17
09/04/2011	1358	11.12	15.09	2	6	15
11/04/2011	1233	11.56	15.07	2	6	16
12/04/2011	1054	11.54	15.15	2	6	16
20/04/2011	1033	10.43	15.21	2	4	12

TABLE 3: Specifications of the servers evaluated.

Server model	Mips	Ram	Bandwidth (GB)	Storage (TB)
Hp proliant G4	1860 each core	4 GB	1	1
Hp proliant G5	2660 each core	4 GB	1	1

TABLE 4: Servers power consumption at different levels of workload per watt-hour.

Server	0%	10%	20%	30%	40%	5%	60%	70%	80%	90%	100%
HP PROLIANT G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP PROLIANT G5	93	97	101	105	110	116	121	125	129	133	135

TABLE 5: Specifications of virtual machines.

VM type	MIPS	RAM
High-CPU medium	2500	0.85
Extra-large	2000	3.75
Small	1000	1.7
Micro	500	613

FIGURE 4: Continued.

FIGURE 4: Analysis charts of four assessment criteria.

TABLE 6: Proposed method compared to other algorithms.

Situation	Compared to IQR	Compared to MAD	Compared to ATEA	Metrics
MMT	7.8389	-1.3986	-2.3529	Energy consumption
MC	11.1368	10.0938	11.9540	
RS	19.4560	12.6984	13.8702	
MMT	21.8312	32.3923	15.0817	SLA
MC	13.2591	30.7054	22.9316	
RS	0.4256	13.6078	-7.2919	
MMT	26.6675	33.4805	20.7185	Esv
MC	17.3187	29.4088	29.9547	
RS	20.4556	28.7686	7.9827	
MMT	0.4455	0.2342	0.2194	Makespan
MC	1.2556	0.8935	1.0294	
RS	0.3101	0.3763	0.2967	

proposed method using the MC migration algorithm with a value of 139.98 has the lowest amount of makespan.

To calculate the percentage improvement of the proposed method over other methods, we use the relation (17) in which the result equals the algorithmic criterion we want to compare the proposed work with. The proposed approach uses the following index:

$$\text{improvement} = \frac{(\text{result} - \text{proposedapproach})}{\text{result}} * 100. \quad (18)$$

As mentioned earlier, for virtual machine migration, the proposed algorithm, besides the MC algorithm, achieved the best results. Table 6 shows the improvement of the proposed algorithm in four different aspects compared to the other algorithms. The energy consumption has about 10% improvement in the worst case and has about 12% improvement in the best case. Also, SLV has about 13% improvement in the worst case and has about 30% improvement in the best case. The ESV, which is an adaptation of the last two criteria, has about 17% in the worst case and 30% in the best case improvement. Finally, the makespan time is 1% better than the others. Values are obtained based on a 17 relationship.

6. Conclusions and Future Directions

In this study, a strategy based on the energy-conscious model and granular neural network for predicting the load of physical hosts and improving cloud resources was proposed and investigated. The purpose of this approach is to find a way to improve the required performance. To maximize their return on investment, Cloud providers have to apply energy-efficient resource management strategies, such as dynamic consolidation of VMs and switching idle servers to power-saving modes. However, such a consolidation is not trivial, as it can result in violations of the SLA negotiated with customers. We have concluded that an adaptive heuristic algorithm has to predict overloaded hosts to improve upon the performance of the optimal deterministic algorithms.

We used three commonly used algorithms to evaluate the proposed method. The performance of the proposed

method has been evaluated concerning four aspects: energy consumption, service-level violations, ESV, and makespan rates. The proposed method, along with the MC algorithm for the migration of virtual machines, in addition to achieving the lowest energy consumption, has the least amount in violation of SLA contracts. Carefully, the charts presented in this article show an inverse relationship between energy consumption and breach of SLA. When the energy consumption is optimized, the rate of breach of SLA contracts increases, or the energy consumption increases as the SLA contracts are reduced. We should try to make this relationship to its minimum. Referring to the results of the ESV function, it is seen that the proposed method along with the MC algorithm for the migration of virtual machines has worked the best. Given that we have not focused on the scheduling approach in the proposed approach, we have still achieved the lowest makespan.

According to the contents mentioned in the research, topics for further study and research are proposed:

- (1) Provide innovative clustering algorithms (combining clustering with evolutionary algorithms) to improve the clustering of granular neural network input data.
- (2) Use evolutionary algorithms to optimize the output of the granular neural network.
- (3) Using the proposed approach to estimate the future works of virtual machines and assigning them to physical hosts in the suitable numbers.
- (4) Using the granular neural network approach in various cloud computing types, including allocation, management, scheduling, migration, etc.

Data Availability

Data supporting this study are previously reported by PlanetLab, which have been cited. The processed data are available at <https://doi.org/10.1145/1113361.1113374>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] P. Corcoran and A. Andrae, *Emerging Trends in Electricity Consumption for Consumer ICT*, National University of Ireland, Galway, Ireland, 2013.
- [2] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.
- [3] A. Andrae and T. Edler, "On global electricity usage of communication technology: trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015.
- [4] L. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer Society*, vol. 40, no. 10, pp. 33–37, 2007.
- [5] P. Barham, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, pp. 164–177, 2003.
- [6] C. Clark, K. Fraser, S. Hand, et al., "Live migration of virtual machines," in *Proceedings of the Networked Systems Design Implementation*, pp. 273–276, Boston, MA, USA, 2005.
- [7] F. Farahnakian, "Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," in *Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 357–364, Santander, Spain, September 2013.
- [8] F. Farahnakian, A. Ashraf, T. Pahikkala et al., "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, 2015.
- [9] L. Zuo, "Dynamically weighted load evaluation method based on self-adaptive threshold in cloud computing," *Mobile Networks and Applications*, vol. 22, 2017.
- [10] J. Yang, "A cost-aware auto-scaling approach using the workload prediction in service clouds," *Information Systems Frontiers*, vol. 16, pp. 7–18, 2014.
- [11] M. Amiri, "A sequential pattern mining model for application workload prediction in cloud environment," *Journal of Network and Computer Applications*, vol. 105, pp. 21–62, 2017.
- [12] M. Amiri, L. Mohammad-Khanli, and R. Mirandola, "An online learning model based on episode mining for workload prediction in cloud," *Future Generation Computer Systems*, vol. 87, pp. 83–101, 2018.
- [13] A. Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," in *Proceedings of the 10th IEEE/ACM International Conference*, pp. 577–578, Melbourne, Australia, May 2010.
- [14] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proceedings of the 8th International Workshop on Middleware for Grids*, Bangalore, India, November 2010.
- [15] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation*, vol. 24, 2012.
- [16] Z. Zhou, Z. Hu, K. Li et al., "Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers," *Scientific Programming*, vol. 2016, Article ID 5612039, 11 pages, 2016.
- [17] R. Yadav, W. Zhang, H. Chen, and T. Guo, "Mums: energy-aware vm selection scheme for cloud data center," in *Proceedings of the 28th International Workshop on Database and Expert Systems Applications (DEXA)*, pp. 132–136, Lyon, France, August 2017.
- [18] R. Yadav and W. Zhang, "MeReg: managing energy-SLA tradeoff for green mobile cloud computing," *Wireless Communications and Mobile Computing* 2017, vol. 2017, Article ID 6741972, 11 pages, 2017.
- [19] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y.-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing," *IEEE Access*, vol. 6, pp. 55923–55936, 2018.
- [20] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Networks*, vol. 26, no. 3, pp. 1905–1919, 2020.
- [21] R. Yadav, W. Zhang, K. Li, C. Liu, and A. A. Laghari, "Managing overloaded hosts for energy-efficiency in cloud data centers," *Cluster Computing*, pp. 1–15, 2021.
- [22] W. Pedrycz, "Granular computing as an emerging paradigm of information processing," *The Springer International Series in Engineering and Computer Science*, pp. 212–224, 2001.
- [23] L. A. Zadeh, "Fuzzy sets and information granularity," *Advances in Fuzzy Set Theory and Applications*, pp. 3–18, 1979.
- [24] Y. Yao, "Three perspectives of granular computing," *Journal of Nanchang Institute of Technology*, pp. 16–21, 2006.
- [25] J. R. Hobbs, "Granularity," in *Readings in Qualitative Reasoning about Physical Systems*, pp. 432–435, Morgan Kaufmann, Burlington, MA, USA, 1990.
- [26] W. Pedrycz, *Granular Computing Analysis and Design of Intelligent Systems*, CRC Press, Boca Raton, FL, USA, 2013.
- [27] W. Pedrycz, "Granular computing as a framework of system modeling," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 1-2, pp. 81–86, 2013.
- [28] A. Bargiela and W. Pedrycz, "Toward a theory of granular computing for human-centered information processing," *IEEE Transactions on Fuzzy Systems*, vol. 16, 2008.
- [29] Y. Yao, *Granular Computing: Past, Present, and Future*, pp. 27–28, Springer, Berlin, Germany, 2008.
- [30] Y. Zhao, "An interactive classification system," in *Proceedings of the 20th Canadian Conference on Artificial Intelligence*, Montreal, Canada, May 2007.
- [31] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," in *Machine Learning: An Artificial Intelligence Approach*, J. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Morgan Kaufmann, Palo Alto, CA, USA.
- [32] T. Chunming and G. Xieping, "The researching development of evolutionary neural networks," *J Syst Eng Electron*, vol. 92–97, 2001.
- [33] K. Park and V. Pai, "A mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.