电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

博士学位论文

DOCTORAL DISSERTATION

论文题目　基于半监督学习的先进聚类算法及其在云计算中的应用

学科专业　软件工程

学　　号　201814090001

作者姓名　Tahseen Khan

指导老师　田文洪　教授

学　　院　信息与软件工程学院

分类号 ______________________ 密级 __________公开__________

UDC 注 1 ______________________

学 位 论 文

基于半监督学习的先进聚类算法及其在云计算中的应用

（题名和副题名）

Tahseen Khan

（作者姓名）

指导老师 **田文洪　教授**

电子科技大学　成都

合作导师 **Rajkumar Buyya　教授**

墨尔本大学　墨尔本

（姓名、职称、单位名称）

申请学位级别 **博士** 学科专业 **软件工程**

提交论文日期 __________ 论文答辩日期 __________

学位授予单位和日期 **电子科技大学　年　月**

答辩委员会主席 ______________________

评阅人 ______________________

注 1：注明《国际十进分类法 UDC》的类号。

Advanced Clustering Algorithms based on Semi-supervised Learning and Applications in Cloud Computing

A Doctoral Dissertation Submitted to
University of Electronic Science and Technology of China

Discipline:	**Software Engineering**
Author:	**Tahseen Khan**
Student ID:	**201814090001**
Supervisor:	**Prof. Wenhong Tian**
Co-Supervisor:	**Prof. Rajkumar Buyya**
School:	**School of Information and Software Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：____________________　　　　日期：　　　年　　月　　日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，同意学校有权保留并向国家有关部门或机构送交论文的复印件和数字文档，允许论文被查阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索及下载，可以采用影印、扫描等复制手段保存、汇编学位论文。

（涉密的学位论文须按照国家及学校相关规定管理，在解密后适用于本授权。）

作者签名：____________________　　导师签名：____________________

日期：　　　年　　月　　日

摘 要

随着云计算的广泛应用，其中的资源管理的需求越来越大。对于复杂场景中资源管理的进行优化的一个重要前提是对于云计算系统运行状态的聚类和预测，包括工作负载、能耗、环境温度等方面。聚类问题受到不同数据集准确性的影响，因此，很难找到一种能够处理所有情况的聚类算法。本文采用基于约束的半监督聚类方法，为了提供很少的监督，本研究采用了诸如成对约束之类的方法。在应用方面，本文提出了半监督聚类算法并应用于云计算领域。由于较大的复杂性和数据中心及其工作负载的非线性特征，传统的启发式或静态的基于资源管理的规则经常无法找到有效的解决方案。本文的创新之处在于，提出了一种适用于不同现实世界数据集的新型聚类集成，并提出了四种不同的聚类算法来估计云数据中心的能耗。因此，本研究侧重于利用回归的基于机器学习的预测方法，并提供半监督聚类算法来预测非线性工作负载和能耗状态。本文的主要工作如下：

首先，对基于机器学习的资源管理进行了全面回顾，确定了现有工作中的挑战问题，并在此基础上提出了未来的研究方向。传统的资源管理依赖于静态规则，这些规则对各种动态设置施加了限制，促使云服务提供商转向数据驱动、基于机器学习的技术。本文评估了基于机器学习的资源管理研究中当前面临的挑战，包括它们的优缺点，以及解决这些问题的当前技术，并基于目前的研究问题和局限性，提出了未来的研究方向。

然后，描述了基于单一聚类算法的半监督聚类集成方法。现有方法在集成生成步骤和聚类集成方法的共识函数中经常使用不同的聚类算法，导致不同聚类算法之间在工作能力方面存在兼容性问题。本文提出了一种基于单一聚类算法（称为 CES）的独特聚类集成技术。由于其产生随机数簇的性质，本研究在该方法的集成生成步骤中循环聚类算法亲和传播（AP）十次，以在每次迭代中创建具有高度多样性的各种基本分区。此外，使用相同的方法（AP）并提出了一种新颖的共识函数，用于将这些基本分区集成到单个分区中，并进行一些调整。AP 仅限于使用这些信息在数据集中产生实际数量的聚类中心，而不是随机数量的聚类，这极大地提高了最终结果的准确性。

接着，介绍了基于机器学习的云数据中心中的工作负载预测和能量状态估计方法。提出了一个基于机器学习的模型来预测负载和能量，从而以这种方式协助资源管理决策。出于建模的目的，本研究提出了基于迁移学习的 GRU 模型，并将其与现有算法进行了比较。在评估指标方面，本文采用均方根误差 (RMSE) 用

于评估预测结果。根据实验数据,GRU 的所有工作负载性能都达到了最低的 RMSE 值，结果优异。同时提出了四种不同的用于能耗状态估计的聚类算法，包括基于迁移学习的半监督亲和传播 (TSSAP)，以根据可能影响能耗的特征而不是为每个虚拟机估计能耗来找到相似的虚拟机组。

最后，应用机器学习技术于云数据中心热能管理中的环境温度预测。当前用于估计温度的解决方案由于其计算复杂性和不准确性而效率低下。这部分的目的是提出一个基于循环神经网络预测环境温度（CPU 和入口温度的组合）的模型。模型可以从单个输入归一化数据中学习，将这些数据与观察到的训练和测试 RMSE 值一起进行预测，以确保所提出混合模型的有效性。本文提出了一个 GRU-RNN 混合模型，并将训练和测试 RMSE 值与已有的先进算法进行了比较。

关键词：机器学习, 聚类, 半监督学习, 云计算, 智能算法

ABSTRACT

With the widespread application of cloud computing, the demand for resource management is increasing. An important prerequisite for optimizing resource management in complex scenarios is clustering and predicting the operational status of cloud computing systems, including workload, energy consumption, environmental temperature, and other aspects. Clustering problems suffer from issues of accuracy on different datasets. So, it is very difficult to find a clustering algorithm which deals with all the situations. This research employs constraints based semi-supervised clustering in this investigation. To give little supervision, this research employs constraints such as pair wise constraints. In terms of application, this research proposed semi-supervised clustering algorithms and used them in the field of cloud computing. Due to the immense complexity and nonlinear characteristics of the data centre and its workloads, conventional heuristics or static rule based resource management rules frequently fail to discover an effective solution. Therefore, this research focuses on machine learning (ML) based predictions utilising regression based approaches, and offers semi-supervised clustering algorithms to predict nonlinear workload and energy consumption state. This research advances the state-of-the-art by making the following key contributions:

The second part of this research represents a comprehensive review that focuses on ML based resource management that identifies challenges in the existing work and proposed future research directions based on it. Traditional resource management has relied on static regulations, which impose restrictions in a variety of dynamic settings, pushing cloud service providers to turn to data driven, ML based techniques. This study assesses current challenges in ML based resource management research, including their benefits and drawbacks, as well as current techniques to tackling them. Finally, this research suggests potential future research directions based on present research problems and limitations.

The third part of the dissertation depicts semi-supervised cluster ensemble based on a single clustering algorithm. Existing methods frequently use distinct clustering algorithms in both the ensemble generation step and the consensus function of the clustering ensemble approach, resulting in a compatibility issue in terms of working capability between different clustering algorithms. This research presents a unique cluster ensemble technique based on a single clustering algorithm called CES. Due to its nature of producing a random number of clusters, this research loops a clustering algorithm affinity propagation (AP) ten times in the ensemble generation step in this method to create various base partitions with a high level of diversity in each iteration. Furthermore, the same method AP is utilised to propose a novel consensus function for integrating these base partitions into a single partition with a few adjustments. AP is confined to producing an actual number of cluster centres in a dataset rather than a random number of clusters by using this information, which greatly improved the accuracy of final results.

The fourth part presents workload forecasting and energy state estimation in cloud data centers approach based on machine learning (ML). This research presents an ML based model to anticipate load and energy to assist resource management decisions in this way. For the purpose of modelling, this research proposed GRU model based on transfer learning and compared with state-of-the-art algorithms. Standard evaluation metrics such as root mean square error (RMSE) are used to assess forecasts. GRU has been discovered to have performed admirably by achieving the lowest RMSE value for all workload performances based on experimental data. This part proposes four different clustering algorithms for energy state estimation, including semi-supervised affinity propagation based on transfer learning (TSSAP), to find similar groups of virtual machines (VMs) based on features that may influence energy consumption rather than estimating it for each VM.

The fifth part represents ambient temperature prediction in the thermal management of Cloud data centers by using ML techniques. Current solutions for estimating temperature are inefficient because of their computational complexity and inaccuracy. The aim of this part is to present a model for predicting ambient temperatures (combination of CPU and inlet temperatures) based on a recurrent neural network. Models can learn from single input normalized data, which must be predicted along with both train and test RMSE values observed in order to ensure the validity of the proposed hyrid model. This study proposed a GRU-RNN hybrid model and compared train and test RMSE values with state-of-the-art algorithms.

Keywords: Machine Learning, Clustering, Semi-supervising Learning, Cloud Computing, Intelligent Algorithms.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Clustering is an unsupervised learning technique that divides items into clusters, with things in similar clusters being similar and those in different clusters being different. This method looks for clusters of comparable objects based on their similarities. For this job, several clustering techniques have been proposed. Clustering algorithms' fundamental purpose is to partition data sets into clusters so that similarities between clusters may be maximised while differences between clusters can be reduced. Generally, dataset features such as noises, overlaps, varied shapes and densities, and so on wreak havoc on clustering problems. As a result, finding a clustering algorithm that works in all cases is quite challenging. Noises affect the partitioning-based clustering technique k-means, increasing the mean value for the cluster centre. DBSCAN, a density-based clustering technique, is ideal for noises, forms, and densities, and automatically determines the number of clusters. However, it contains two parameters that must be set by the user. Affinity Propagation, an affinity-based clustering algorithm, produces a random number of clusters. Semi-supervised learning, which integrates both supervised and unsupervised data, has gotten a lot of attention in recent years. Some prior knowledge, such as pair-wise constraints, is used to improve semi-supervised learning methods. Graph-based clustering, mean-shift clustering, and constrained-spectral clustering are examples of frameworks that combine pairwise constraints with unsupervised learning approaches. Finding labeled data was extremely challenging; but, acquiring a large volume of unlabeled data was rather simple. As a result, semi-supervised learning was developed, in which a tiny quantity of supervised data is provided to increase the algorithm's accuracy. By merging semi-supervised learning and clustering methods, all of these issues were able to be tackled. The rest of the research is based on applications that use advanced semi-supervised clustering algorithms to optimise energy utilisation of cloud-based data centres.

Cloud computing environments include data centres. Due to multitenant users, shifting workload conditions, and increasingly complicated infrastructures, resource management in a data centre is often a tough operation. Workloads in modern data centres are highly non-linear. According to an IBM survey, cloud applications' average CPU and memory use range from 17.76% to 77.99% [1]. According to a Google study, a cluster's CPU and memory utilisation cannot surpass 60%, resulting in significant resource waste in cloud data centres [2]. As a result of the workload's non-linear usage patterns, performance is erratic, energy consumption is excessive, and service quality is impaired (QoS). It also raises operating costs and reduces revenue for service providers. Because data centres are costly to develop and operate, resource utilisation must be maximised. While ensuring the application's Quality of Service, an intelligent energy prediction technique can successfully tackle the issue by increasing resource consumption and lowering operational expenses (QoS). To address the above challenges of clustering algorithms and energy usage of cloud data centres' resource management system, many solutions have been proposed including cluster ensemble and power model-based solutions respectively.

Energy consumption in cloud data centres as an application of advanced semi-supervised clustering is dealt with. In data centre resource management, energy estimation is critical. Energy consumption is a major issue in data centres, and providers are working to reduce overall energy use through better resource management. In today's data centres, hosts feature a variety of sensors that monitor energy at the host level. Recent research has focused on calculating energy usage for each virtual machine (VM) using multiple power models [3,4]. However, calculating the energy consumption of VMs at the software level is difficult. For example, memory energy consumption is calculated based on the events raised by each VM on the last level cache of each core (LLC). To calculate energy consumption, these LLC measurements need to be collected, which makes estimating the energy of each VM a difficult operation [5]. Rather than estimating energy for each VM, patterns of comparable VMs in various energy-consumption situations are looked at. This is accomplished by looking at the available energy usage features and using clustering analysis to find VMs with similar patterns. Finally, this research focuses on advanced semi-supervised clustering algorithms and their application in cloud data centre energy state prediction by proposing a novel semi-supervised cluster ensemble based on a single clustering algorithm and four different semi-supervised clustering algorithms to find similar VMs based on features that affect energy consumption the most respectively.

1.1 Motivations

Massive energy difficulties have arisen as a result of the huge growth of cloud data centres. In the worst-case scenario, data centres might consume up to 8000 terawatts of power by 2030 if essential steps are not taken. This vast energy consumption may be reduced to roughly 1200 terawatts if best practises are implemented across the Cloud computing stack. Adopting energy-efficient strategies into the various levels of data centre resource management platforms is required to accomplish this best-case scenario (such as optimised use of computing and resources such as CPU).

As a result, this research looks at how it can apply ML techniques, particularly clustering and deep learning approaches, to various data centre resource management challenges to optimise resource usage and energy consumption. This necessitates the use of proper ML algorithms to learn and predict desired outputs, as well as appropriate clustering approaches. Both scenarios, such as workload forecasting and energy status estimate, are crucial for a data center's energy efficiency and must be handled. As a result, monitoring the energy of each VM about the total energy of a host is a good idea [6]. Each component of a host, such as the CPU, RAM, and disc, contributes to the total energy of the host. Thus, awareness of energy consumption at the VM level can assist energy monitoring of hosts, but measuring the energy consumption of VM devices at the software level is exceedingly challenging. Because LLC (last-level-cache) events triggered by each VM on each core must be collected at the VM level, measuring becomes more challenging [5]. As a result, rather than evaluating the energy of each virtual machine, this research opted to look at the patterns of similar virtual machines that are over- or under-utilized. To find VMs with similar patterns, clustering analysis might be performed. The focus of the research is on automation. Thus, this research employs an ML approach such as clustering to teach the machine these states automatically. Clustering automatically discovers similarities between features and classifies data into similar and different categories.

1.1.1 Dissertation Statement

Numerous subsystems in data centres, including computing (application and storage servers), networking equipment, cooling systems, and other facility-related systems, collaborate closely to offer consumers reliable services. Two important subsystems that consume a large amount of energy are cooling and computation. As a result, improving the efficiency of these two subsystems is critical for making cloud data centres more energy-efficient. This research presents novel semi-supervised clustering techniques and shows how they may be applied to cloud data centre workloads to make them more energy-efficient. To attain this goal, this research addresses the following research questions to solve key nonlinear workload challenges.

Q1. How can this research find future research directions in ML-based resource management in cloud data centres?

Q2. How this research can cluster the data by using a more accurate and time-efficient cluster ensemble which can be further used in many resource management tasks?

Q3. How this research can predict highly non-linear workload accurately and the energy consumption state of VMs at the software level?

Q4. How does this research predict the highly non-linear ambient temperature of hosts accurately while ensuring efficient training of the model?

1.1.2 Dissertation Contributions

This research systematically addresses the energy efficiency problem of cloud data centres through various proposed semi-supervised clustering techniques. It presents a detailed survey of the existing resource management techniques where ML has been applied, identifies challenges and proposes potential future research directions. Individual research papers offered a new semi-supervised clustering technique and its application to cloud computing. The important contributions of this research are stated below, based on the research problems mentioned above:

(1) Identifies challenges in machine learning (ML)-based resource management and proposes potential future research directions based on identified challenges. (addresses the Q1).

(2) Proposes a novel semi-supervised cluster ensemble based on a single clustering algorithm. It uses the same clustering algorithm in both stages of a cluster ensemble such as the ensemble generation step and consensus function. (addresses the Q2).

(3) Proposes workload forecasting and energy state estimation in cloud data centers approach based on ML. It first forecasts non-linear workloads by using various ML techniques and then it clusters VMs based on the features that affect energy consumption the most by proposing four different semi-supervised clustering algorithms. (addresses the Q3).

(4) Proposes ambient temperature prediction in thermal management of cloud data centers by using ML techniques. This work proposes a model by using recurrent neural networks and predicts the ambient temperature of hosts. (addresses the Q4).

1.2 Dissertation Organization

This research examines several semi-supervised clustering algorithms and their applicability to various types of data sets, including real-world data sets and data centre workloads. To begin, this research give surveys of existing ML-based cloud computing systems, in which this research identifies issues and limitations of existing work and recommends future research areas based on it. Then, based on semi-supervised learning, this research proposes a novel cluster ensemble technique that clusters real-world datasets. This research proposes four distinct semi-supervised clustering algorithms and groups VM's non-linear workload to predict energy state using this idea. This research also looks into several deep learning techniques for predicting non-linear workloads, which was necessary to recognise the important characteristics of this workload for the research.

Chapter 2 A Review of ML Centric Resource Management

2.1 Outline

As a method for providing utility computing services over the Internet, cloud computing has quickly gained popularity. The model of cloud computing known as “Infrastructure as a Service” (IaaS) is one of the most significant and expanding ones. Some of the key components of cloud computing for IaaS include scalability, quality of service, optimal utility, fewer overheads, better throughput, lower latency, specialised environment, cost-effectiveness, and a simplified interface. Static policies have been used for resource management in the past, but they have significant limits in a variety of dynamic settings, which has led cloud service providers to adopt data-driven, machine-learning-based strategies. Workload estimate, task scheduling, VM consolidation, resource optimisation, and energy optimisation are just a few of the resource management activities that are handled by ML. An in-depth analysis of resource management systems based on ML is provided in this research. Fundamental cloud computing ideas, such as service models, deployment methods, and ML applications are introduced. Then, the problems with resource management in cloud computing are examined, classified according to different aspects of resource management types like workload prediction, VM consolidation, resource provisioning, VM placement, and thermal management, reviewed the available solutions to these problems, and assessed their main advantages and disadvantages. Finally, based on observed resource management issues and deficiencies in existing techniques to address these challenges, potential future research topics are suggested.

2.2 Introduction

Cloud computing has paved the path for the rise of computing as a fifth utility by allowing users to utilise software and IT infrastructure [7]. In cloud computing, resource management in data centres is still a challenge, and it is highly reliant on the application workload. In traditional cloud computing settings, such as data centres, applications were connected to particular physical servers, which were frequently overprovisioned to manage difficulties associated with peak demand [8]. The data centre was costly to run in terms of resource management due to the squandered resources and floor space. On the other side, virtualization technology has demonstrated that it may make data centres easier to manage. Server consolidation and greater server utilisation are only two of the advantages of this technology. Google, Microsoft, and Amazon, for example, have huge data centres with complex resource management. Servers, virtual machines (VMs), and other management roles are all part of the resource management of these enormous data centres [9]. A server host is allocated multiple VMs with varying workload types and amounts in these data centres. Because of the fluctuating and unexpected demand, a server may be overworked or underworked, resulting in an imbalance in resource use allotted to VMs on a particular hosting server. This might lead to difficulties including uneven quality of service (QoS), imbalanced energy usage, and service level agreements (SLA) breaches [10]. According to a survey on imbalanced workload, the average CPU and memory utilisation were 17.76% and 77.93%, respectively, while comparable research at the Google data centre discovered that the CPU and memory utilisation of a Google cluster could not surpass 60% and 50%, respectively [11]. A data center's productivity decreases as a result of the unbalanced workload, resulting in greater energy usage. It is proportional to the operational costs and financial loss of the data centre. Because an ideal machine absorbs more than half of the maximum energy consumption, excessive energy consumption has a direct influence on carbon footprints, which should be avoided [12]. Data centres utilised about 35 Twh (Tera Watt hour) of energy in 2015, according to an EIA (Energy Information Administration) assessment, and this figure is anticipated to grow to 95 Twh by 2040 [13].

The best mapping between VMs and servers must be determined in order to balance resource usage [14]. This is a difficult problem class that is NP-complete. As a result, to fulfill QoS standards while also boosting data centre benefits, an intelligent resource management strategy is required [15]. Future insights will be generated by the intelligent processes, which will help applications in mapping to machines with improved resource use [16]. However, predicting future insights is difficult due to the nonlinear and unpredictable behaviour of workloads for VMs. However, there are two ways to get this future insight: historical workload-based prediction methods, which generate insight by learning trends from historical workload data, and homeostatic-based prediction methods, which provide an upcoming future workload insight by subtracting the previous workload from the current workload [17]. In addition, the mean of the preceding workload may be static or dynamic. Both techniques have benefits and drawbacks, although historical-based predictions are regarded as more straightforward and well-known in this field.

Thus, intelligent resource management will play a crucial role in optimising the data center's SLA, energy use, and operating costs by undertaking effective and intelligent resource provisioning. In data centres, resource management covers a wide range of operations, including resource supply, reporting, workload scheduling, and a number of other tasks [18]. The provisioning of resources is central to many of these tasks. The goal of resource provisioning is to assign cloud resources to VMs based on end-user demands while ensuring that SLAs such as availability, dependability, response time limit, and cost limit are met to the greatest extent possible [19]. It should allocate resources based on end-user demand and avoid over- or under-provisioning, such as allocating more or less resources to VMs. There are two methods to use this resource allocation technique: proactive and reactive. Resource provisioning is centred on workload before prediction in proactive approaches, which is calculated by learning trends from historical workload, whereas reactive measures are implemented after resource demand has arrived. As a result, it can be deduced that the experience of historical-based prediction methods may be effectively included in proactive approaches to provide intelligent dynamic resource scaling, which contributes to intelligent dynamic resource management. Other activities, such as virtual machine consolidation, task scheduling, and thermal management, can also be done based on forecasts to improve resource utilisation, reduce energy consumption, and improve QoS. Computer vision, pattern recognition, and bioinformatics all use machine learning (ML) techniques. The progress of machine learning algorithms has benefitted large-scale computing systems [20]. Google [21] recently produced a paper describing their efforts to optimise electricity, cut expenses, and enhance efficiency. By providing data-driven approaches for future insights, machine learning has brought attention to dynamic resource scaling, which is seen as a promising way to anticipate workload quickly and correctly.

As a result, this article focuses on a review based on issues encountered in state-of-the-art research in resource management using ML algorithms, covering provisioning, VM consolidation, temperature prediction, and other management approaches. Then this research will go over the benefits and drawbacks of different state-of-the-art resource management research projects that employ machine learning methods. This research will also go over the experimental setup, as well as the data sets used and performance improvements. Finally, this research suggests future study directions based on present research problems and limits. Figure 2-1 depicts the cloud computing components while using machine learning.

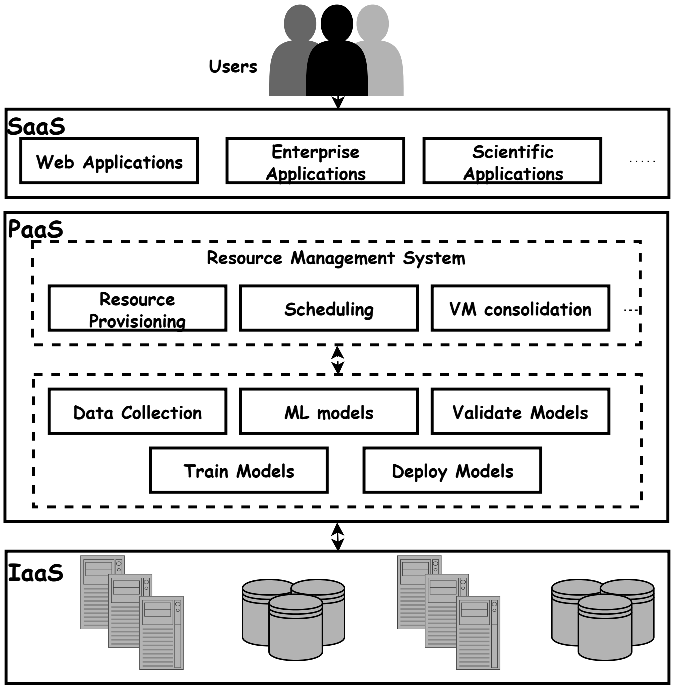


Figure 2-1 Components of Cloud Computing Paradigm Using Machine Learning

2.2.1 Aim and Motivation of Research

Resource management is problematic in cloud operations because multi-tenant end-users demand nonlinear workloads, resulting in many over- and underutilised servers. It has a direct impact on whether electricity is used excessively or inefficiently, resulting in high operational costs. As a result, a prior estimate of workload based on historical data can help intelligent resource management. Static policies are frequently used to manage resources in cloud computing systems, and they have two flows: they are based on a static threshold value that is adjusted in offline mode, and they appear to require reactive behaviour, which may result in excessive overheads and delays in customer responses. These tactics fail in a dynamic situation, such as when the load reaches a static threshold and then swiftly lowers, showing that VM movement isn't required in the case of VM consolidation. They are also unable to grasp the dynamics of technology and workload in complex dynamic contexts (such as Cloud and Edge) and so fail to progress [18]. To overcome these drawbacks, machine learning has replaced static heuristics with dynamic heuristics that adapt to the actual workload in production [22,23]. Machine learning approaches provide predictive management by providing future insight based on historical data. As a result, in an ML-centric RMS, a data-driven Machine Learning (ML) model can estimate future workload demand and govern resource auto-scaling accordingly. Consumers and service providers who wish to improve their QoS and maintain a competitive edge in the market would benefit greatly from such techniques. In the case of cloud resource management, machine learning has been found to provide more accurate forecasts than more traditional methods like time-series analysis [24,25]. For intelligent resource management, several machine learning techniques have been developed to forecast prior workload. Moreover, a number of IT behemoths have started to look into machine learning-based resource management in production [26]. Google uses a neural network [27] to optimise fan speeds and other energy variables. Microsoft Azure [9] uses a framework resource centre to give online forecasts of various workloads using multiple ML Gradient Boosting Trees. Despite these earlier initiatives and prospects, the optimum strategy to incorporate machine learning into cloud resource management remains unknown at the moment. As a result, it's more important than ever to provide research that addresses current difficulties and indicates potential future study routes while simultaneously emphasising the merits and limitations of existing research.

2.2.2 Research Questions

There are several research questions that need to be addressed, of which this research will mention a few here and propose potential future research directions.

(1) How to reduce the time complexity of ML algorithms in ML-based resource management in data centres?

(2) How can the accuracy of workload prediction ML algorithms be improved?

(3) How can training time be reduced while developing an ML model?

(4) How can VMs collaborate in similar groups to estimate the state of energy consumption?

(5) How to reduce energy consumption more effectively?

(6) How do different workloads, such as disc, network, CPU, and memory, affect energy consumption and play an important role in VM consolidation and resource provisioning?

2.2.3 Our Contributions

The following are the main contributions of our work:

(1) This research presents a review of ML-based resource management approaches in cloud computing based on identified challenges in the state-of-art research.

(2) This research identifies the advantages and drawbacks of these methods, as well as their experimental configuration, data sets used, and performance improvements.

(3) This research proposes potential future research directions based on identified challenges and limitations in the state-of-art research to strengthen resource management

2.2.4 Related Surveys

A few research on ML-based resource management in cloud computing have been published. The researchers offered a comprehensive overview of the most relevant research initiatives on data centre resource management, with the goal of optimising resource use [28]. The essay then summarises two important components of the resource management platform and discusses the advantages of precisely anticipating workload in resource management. The researchers focused on resource provisioning, resource allocation, resource mapping, and resource adaptation, among other essential resource management techniques [29]. The researchers surveyed the state of the algorithms, organised them into categories, and addressed closely related topics such as virtual machine migration, forecast methods, stability, and availability [30]. The researchers reported considerable improvements to previous work based on approach optimization, techniques, and objective models [31]. The researchers lay forth a conceptual framework for cloud resource management and use it to organise the state-of-the-art review [32]. The researchers presented a detailed assessment of the most up-to-date VM placement and consolidation techniques utilised in green cloud, with a focus on increasing energy efficiency [33]. The researchers presented a broad overview of IT consolidation at various levels of cloud services, as well as a virtualized data centre and consolidation overview [34].

These articles do not go into greater depth on ML-based resource management or the obstacles and issues that exist in current state-of-the-art and future research paths. As a result, it's critical to give a comprehensive survey that covers the numerous ML algorithms employed in the data centre resource management scenario, as well as their flaws, obstacles, and possible directions, as per the vision. As a result, before moving forward with new ideas in this direction, researchers can use this article to analyse present ML scenarios in cloud resource management and their inadequacies.

2.2.5 Chapter Structure

The remaining sections of this chapter are organised as follows: The background details and definitions for cloud computing components and ML are given in Section 2.

Section 3 discusses the challenges of ML-based resource management in cloud computing systems, as well as the benefits and drawbacks of current research. Section 4 proposes future research directions based on the challenges and limitations pointed out in state-of-the-art research, and Section 5 concludes this chapter.

2.3 Background and Terminologies

2.3.1 Cloud Computing

Cloud computing is the provisioning of resources such as memory, CPU, bandwidth, disc, and applications/services over the Internet. According to the National Institute of Standards and Technology (NIST), [35] Cloud computing is a concept for giving on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, software, and services) that may be supplied and released fast with no administration effort or service provider participation. This cloud model has five key elements, three service models, and four deployment choices. Two more qualities have been added based on the literature.

A client-server architecture is used in this computing model to enable for centralised application deployment and compute offloading. On both the client and server sides, cloud computing is cost-effective in application delivery and maintenance, as well as flexible in resource provisioning and detaching services from related technologies. Many advanced computing systems have been released to the market, including Alibaba Cloud, Microsoft Azure, Adobe Creative Cloud, ServerSpace, Amazon Web Services (AWS), and Oracle Cloud. Cloud computing and its supporting technology have been investigated for years, and many advanced computing systems have been released to the market, including Alibaba Cloud, Microsoft Azure, Adobe Creative Cloud, ServerSpace, Amazon Web Services (AWS), and Oracle Cloud.

2.3.2 Core Features of Cloud Computing

(1) On-demand self-service: A client can use an online control centre to query one or more services as needed and pay utilising a “pay-and-go” mechanism without engaging with living beings.

(2) Broad network access: Resources and services in various cloud provider areas can be accessed from a variety of locations and provisioned using common protocols by incompatible thin and thick clients. This characteristic is also known as “global” reach

capacity and "easy-to-access" standardised methods [36,37].

(3) Resource pooling: It offers a set of resources that act as if they were one blended resource [38]. In other words, the client is not aware of the location of the provided services and is not expected to be. This strategy enables vendors to dynamically include a variety of real or virtual services in the cloud.

(4) Rapid elasticity: Scalability is a synonym for elasticity, which refers to the ability to scale resources up or down as needed. Clients have the ability to request as many services and resources as they desire at any moment. Because of this consistency, Amazon, a well-known cloud service provider, named one of its most popular and widely used services the elastic compute cloud [39].

(5) Measured service: For both vendors and users, various aspects of the cloud should be automatically regulated, monitored, optimised, and recorded at several abstract levels.

(6) Multi-Tenacity: This is the fifth cloud characteristic proposed by the Cloud Security Alliance. Models for policy-driven compliance, segmentation, separation, governance, service levels, and chargeback/billing for distinct client categories are required for multi-tenacity [40].

(7) Auditability and certifiability: It is critical that services plan records and trials in order to analyse how well laws and rules are being followed [36].

2.3.3 Cloud Computing Service Models

(1) Software as a service (SaaS) [41]: A client can access the service provider's cloud-hosted apps using this service model. Applications are accessed using web portals. This strategy has made production and testing easier for providers because they have access to the applications.

(2) Platform as a service (PaaS) [42]: The service provider offers fundamental necessities such as network, servers, and operating system under this service model, allowing the customer to create new applications and maintain their configuration settings.

(3) Infrastructure as a service (IaaS) [43]: The user has already generated all of the required applications and only needs a basic infrastructure. In such circumstances, vendors could include processors, networks, and storage as facilities with customer provisions.

2.3.4 Deployment Models for Cloud Computing

(1) Public cloud [44]: This is the most common cloud computing paradigm, in which the cloud owner delivers public services over the Internet in most circumstances, based on specified rules, restrictions, and a business model. With a large number of regularly used resource bases, suppliers can provide customers with a variety of options for selecting relevant resources while ensuring QoS.

(2) Private cloud [45]: A private cloud is built and configured to give a corporation or institution most of the benefits of a public cloud. Due to the utilisation of corporate firewalls, setting up such a system would result in fewer security issues. The high costs of building a private cloud are a fatal weakness because the company in charge of the system is responsible for all aspects of it.

(3) Community cloud [46]: A group of organisations comes together to share cloud computing with the customers of their community members based on shared criteria, concerns, and standards. A third-party service provider or a group of community members can supply the necessary cloud computing infrastructure. Cost reductions and cost-sharing among community members, as well as excellent security, are the most significant advantages of a community cloud.

(4) Hybrid cloud [47]: By combining two or more independent public, private, or community clouds, a new cloud model known as the hybrid cloud was born, in which constituent services and infrastructure retain their unique characteristics while also requiring standardised or agreed-upon functionalities to enable them to communicate in terms of application and data interoperability and portability.

ML is the study of teaching machines to make predictions or recognise items without being specifically programmed to do so [48]. One of its fundamental assumptions is that by combining training data with statistical approaches, it is feasible to create algorithms that can predict previously unknown values. ML has progressed from a research project to a widely utilised commercial technology in the previous two decades. ML has emerged as the preferred tool for designing functional apps for computer vision [49], speech recognition [50], natural language processing [51], robot control [52], self-driving cars [53], effective web search [54], purchase recommendations [55] and other applications in the field of artificial intelligence (AI). Many AI system developers now understand that, for many applications, training a system by showing it examples of desired input-output actions is much simpler than programming it manually by predicting the desired answer for all possible inputs.

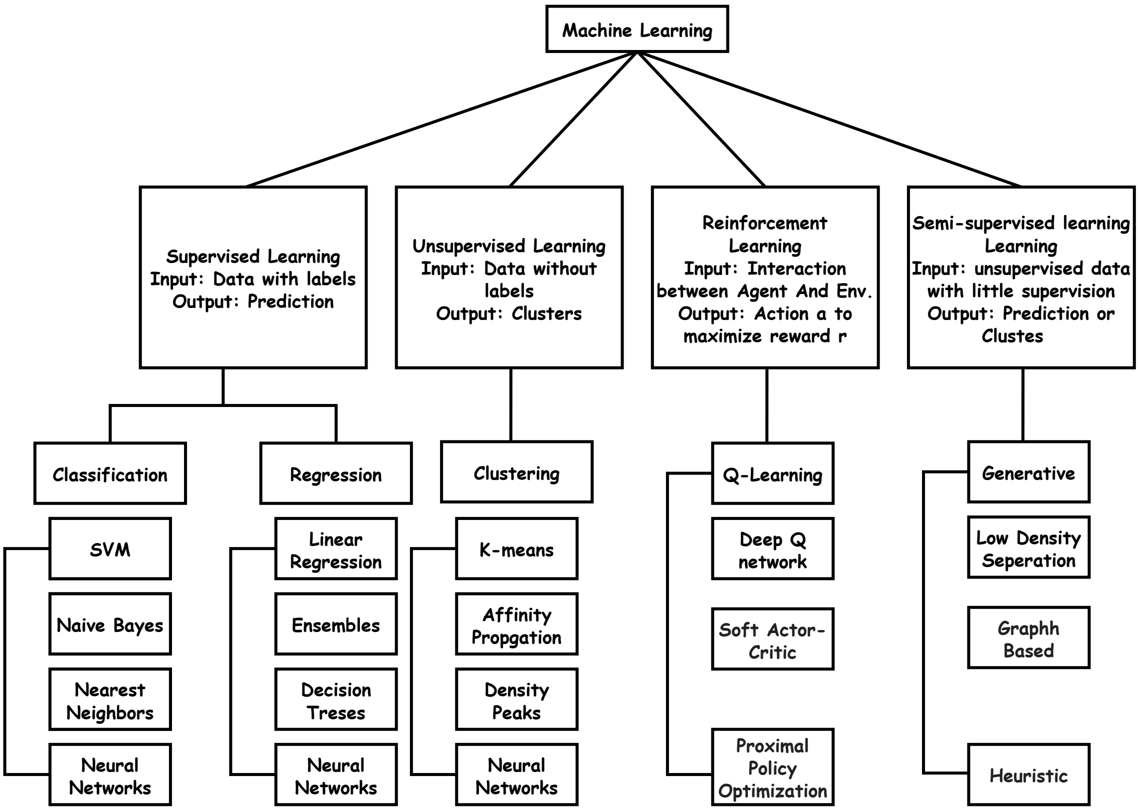


Figure 2-2 Taxonomy of Machine Learning

This success is primarily owing to the accessibility of massive data and increased efficiency in the processing power of servers and GPUs [56]. Based on the modeling objective and the problem at hand, machine-learning algorithms are categorised as supervised learning, semi-supervised learning (SSL), unsupervised learning, and reinforcement learning (RL). Unsupervised learning is categorised as clustering and dimension reduction [57–59], among other things, while supervised learning is categorised as the classification problem (e.g., sentence classification [60,61], image classification [62–64], etc.) and regression problem.

2.3.5 Machine Learning

(1) Supervised learning [65]: In supervised learning, each data sample is made up of numerous input attributes and a name. The learning process is to get as near to a mapping function that connects the features to the label as possible. The mapping function can then be used to produce label predictions for the data based on new input features. This is the most extensively used ML method, and it's been applied to a wide range of applications.

Supervised learning is exemplified by the classification problem, which is identifying an object based on its features, such as classifying a mobile phone by its brand name and specifications. This is a regression task if the supervised learning task is to forecast a continuous variable like stock pricing. As shown in Figure 2-2, this research can further categorise supervised learning based on the model form.

(2) Unsupervised learning [66]: In contrast to supervised learning, unsupervised learning occurs when this research has merely input features but no names to go with them. As a result, the goal of unsupervised learning is to figure out the data distribution and show how the data points differ from one another. Unsupervised learning is shown by the clustering problem, which involves discovering data groupings, such as grouping VMs based on their resource use patterns.

(3) Semi-supervised learning [67]: It's a type of ML that tries to bring these two tasks together. By combining knowledge from the other, SSL algorithms normally strive to improve efficiency in one of these two jobs. Additional data points with unknown labels, for example, may be employed to aid in the classification process when dealing with a classification problem. Knowing that several data points belong to the same class, on the other hand, will aid in the learning process for clustering methods.

(4) Reinforcement learning [68]: RL differs from both supervised and unsupervised learning in various ways. When utilising reinforcement learning to train an agent, it is not required to employ labeled input/output pairings or explicit correction on sub-optimal decisions. Instead, through interacting with the environment, the agent tries to achieve a balance between exploration and exploitation. The agent is rewarded by the translator for making good decisions or acting in a certain way. It would be sanctioned if it wasn't. In robot and computer game agent research, reinforcement learning is widely employed.

2.4 Challenges, State-of-the-art Research and their Limitations

In this section, this research discusses challenges identified in ML-based resource management in state-of-the-art research. In addition, this research explores current approaches to addressing these challenges, as well as their advantages and limitations.

2.4.1 Performance and Online Profiling of Workload

In cloud resource management research, the primary components of large commercial providers' workloads are not thoroughly addressed. They don't look at the lifetime virtual resource use of VMs, for example. The vast majority of research focuses on offline workload profiling, which is impractical because the input workload may not be available until the VMs are not in production. On the other hand, online profiling is tough since it's impossible to tell when a random VM has demonstrated representative behaviour. Resource management can be more effective if diverse task characteristics can be reliably forecasted with little time complexity. As a result, prediction algorithms face another challenge in terms of accuracy and time complexity.

The researchers presented an ML-based prediction system on Microsoft Azure compute fabric [9]. This system may learn behaviour from previous data and provide predictions to various resource managers, such as Server health manager, migration manager, container scheduler, and energy capping manager, via a rest API. They also disclosed extensive Microsoft Azure real-world workload traces from this system, which demonstrate that numerous VMs exhibit peak CPU utilisation in varied ranges on a constant basis. In the case of overloaded servers, they changed Azure's VM scheduling to use resource-central benefit forecasts. This forecast-based timetable helps to avoid physical resource misuse and tiredness. However, despite the fact that memory consumption plays a key influence in physical resource exhaustion, (1) they did not consider memory utilisation in released traces or the predictive system RC. (2) They analysed CPU utilisation time series to establish whether a VM is interactive or delay-insensitive, classified the workload into these two groups, and performed supervised classification of these VM workloads using an Extreme Gradient Boosting Tree (EGBT). However, they did not consider the issue of a distributed data centre, where data is dispersed and incomplete labels for these two classes may exist; in this instance, there will be insufficient labels to train their algorithm.

2.4.2 Multiple Resource Usage in VM Consolidation

In order to power off the remaining hosts and save energy, VM consolidation technologies seek to consolidate more VMs on a smaller number of hosts. In this procedure, most researchers used current CPU use to determine if a host was overwhelmed or not. This could result in unnecessary VM migration and host power mode shift, reducing the efficiency of the consolidation operation. The host with the highest CPU utilisation is the destination host for migrating VMs, but due to a lack of future estimates, this may result in overutilization. As a result, future resource utilisation estimation can help to solve this problem. Other resource use, such as memory and disc, can overburden the host, making the consolidation process complex and challenging.

To improve energy usage, the researchers proposed an intelligent VM consolidation technique [69]. This technique anticipated resource utilisation in the past based on previous data and used that prediction to select a host with higher utilisation in advance for VM migration. This problem was solved using a dynamic consolidation approach. An ML method called Linear Regression (LR) was utilised to forecast the future utilisation of all VMs. Real workload traces from PlanetLab VMs [70] were used for this exercise. To model a data centre and apply their VM movement approach to save energy, they used the CloudSim toolkit. On a bigger simulated benchmark with 7600 hosts, their method provided the key benefit of taking into account time overheads while lowering energy use. However, if this strategy is employed in the creation of real-world workloads, the time overhead is a significant element that is also influenced by the data training time of the ML algorithm. They did, however, explore the LR technique, which relies on a variety of factors to predict the target variable, making it time-intensive and potentially hurting the data center's reaction time.

2.4.3 Cloud Network Traffic

The present VM allocation research includes a variety of strategies for allocating a single VM to a host and allocating various VM resources by guaranteeing that each host has the capacity to handle the task. Because the application demand varies from time to time, with a combination of high and low resource utilisation, this method results in inefficient resource utilisation. Various applications have different resource demands, which are assigned to appropriate VMs in data centres, resulting in varying resource demand patterns. Furthermore, many VM placement methods only evaluate current resource utilisation, such as CPU demands; nevertheless, a constantly changing workload presents a problem with such solutions. Future resources, such as CPU demand, may be more useful in VM placement techniques. Cloud network bandwidth is becoming another hard aspect in data centre resource management, in addition to CPU resource requirements [71] [72]. Research reported that there will be 51,774 GB/sec amount of internet traffic would be produced because of computing as a service via cloud computing and this would affect the cloud network as well [73]. This key factor affects the VM migration time in case of dynamic VM placement and violates SLAs [74].

The researchers developed a network-aware predictive VM placement heuristic that considers CPU demand as well as network bandwidth to reduce energy consumption and SLA breaches [75]. The main benefit of their work was that they were able to design a dynamic VM placement strategy that was based on the prediction of both CPU utilisation and network bandwidth. Estimating network bandwidth in the case of large VM migrations aids in making better scheduling decisions and makes VM placement more efficient and reliable. As a result, VM placement techniques should take into account future resource insights in order to balance restricted resource availability and manage energy efficiently. They did not, however, evaluate another factor, disc throughput, which could affect VM migration time [76].

2.4.4 Host Temperature

Minimizing host temperature in modern cloud data centres is a difficult task. This is caused by the heat that is emitted during the host's energy consumption process. To keep the temperature of the host below the threshold, cooling systems are used to remove this dissipated heat. This higher temperature has a direct impact on cooling system costs and has become a difficult problem to tackle in resource management systems. It also causes various system failures by creating host spots in the system. As a result of the dynamic behaviour of the host's temperature, thermal management is both required and difficult.

The researchers proposed a thermally aware predictive scheduling method for lowering a host's peak temperature and energy usage [77]. Because most data centres contain monitoring devices that record a variety of factors such as resource utilisation, energy consumption, thermal readings, and fan speed measurements, this type of information was gathered from the University of Melbourne's private cloud data centre. They used different ML methods to predict host temperature and created a thermal aware scheduling strategy to reduce energy consumption by minimising host peak temperatures while transferring VMs to the fewest hosts possible. The prediction model is used in this approach to anticipate the host temperature, and further scheduling is guided. In compared to existing algorithms, their work reduces peak temperature by up to 6.5° and 34% energy usage, that reducing temperature by even one degree can save up to millions of dollars in a large-scale data centre [27]. They use the host's ambient temperature for prediction rather than the CPU temperature, which is a combination of inlet and CPU temperature; nevertheless, this may increase algorithm overhead.

2.4.5 False Host Overloaded Detection

Overloaded host identification is unreliable due to the current resource utilisation prediction, especially when the current resource utilisation exceeds a threshold value. The difficulty occurs in choosing whether or not the VMs assigned to this host should be relocated because the load drops rapidly after a short period of time, resulting in a false hot detection point, i.e., false overloaded host detection. However, VMs must be relocated if the length of load decrease is long enough to minimise over-utilization. The resource management system faces a unique difficulty in avoiding unnecessary VM migration overhead with such a VM consolidation approach.

The researchers suggested a VM consolidation technique based on multiple-use prediction and multi-step prediction to restrict unwanted VM migrations and reduce data centre overheads and energy consumption [78]. As a result, based on historical data for a specific PM, this mechanism was generated to estimate the long-term utilisation of several resources such as CPU and memory. The basic goal of VM consolidation is to identify overcrowded and underloaded hosts. To identify the overloaded and underloaded hosts, they took into account both current and predicted resource usage. Based on local historical data, an efficient multiple usage prediction technique was presented to compute the long-term utilisation of various resource kinds. A VM consolidation based on multiple usage forecasts was also presented to save energy by minimising unwanted VM migrations from overcrowded hosts. As a result, the combination of present and expected resource use is critical for detecting overloaded and underloaded hosts. According to this definition, a host is overloaded if it meets two criteria: (1) it is overloaded in both current and expected resource use, and (2) it is in normal condition but will be overloaded in the future. Following these two limits, VM consolidation was conducted based on the discovered overloaded hosts. They did not, however, explore the issue where a host is overloaded in the now but will not be overloaded in the future. This is something that should be taken into account while putting together a VM consolidation plan.

2.4.6 Energy Metering at Software-level

Modern servers contain several energy metres to track energy usage, but they can't track the energy of a single virtual machine, which is difficult to achieve because energy is difficult to measure at the software level. Energy consumption has become a tough aspect to consider for a successful VM consolidation phase, according to data centre energy budgets. The previous study focused solely on server resource use for VM consolidation, which contradicted the energy capping method by growing across the levels of particular servers during the process, violating energy limits. The term “energy capping” is used to describe a hardware-based procedure. As a result, lowering the CPU frequency lowers the combined server’s energy usage, which is in violation of the energy constraints. As a result, lowering the server’s CPU clock owing to the load of one VM has a knock-on effect on all other running VMs. As a result, the efficiency of workloads operating in VMs suffers, breaching SLAs and the virtualization’s isolation characteristic. In data centres, the two most frequent solutions are VM consolidation and energy capping, however, neither allows for reliable monitoring of energy usage for individual VMs.

The iMeter energy consumption prediction methodology, which is based on the Support Vector Regressor ML algorithm, was proposed by the researchers (SVR) [79]. They used principal component analysis (PCA) to find the most closely related components that drove VM energy consumption, as well as to forecast individual VM and multiple consolidated VM energy consumption for a variety of workloads. Due to the numerous types of cloud resources stored in the VM, such as CPU, memory, and IO, and the fact that different cloud end users can require varying quantities of the same resources at the same time, estimating the energy consumption of a single VM is difficult. In addition, the resource manager must make individual decisions for each VM, slowing end-user response time and violating QoS.

2.4.7 SLA-based VM Management

Data centres have traditionally employed over-provisioning to avoid the worst-case situation of high load utilisation while still maintaining SLA commitments. The hosts, on the other hand, utilise very little energy during regular hours, resulting in resource waste. The researchers looked at actual workload traces of VMs’ resource utilisation from the Google data centre and discovered that the average CPU and memory utilisation was less than 60% and 50%, respectively [2]. As a result of overprovisioning services, more maintenance costs in host cooling and administrative activities are incurred [28]. The aim of research has been to solve this difficult problem by using dynamic resource provisioning of resources in virtualization technology, but it primarily focuses on a particular form of SLA or application, such as transactional workload. However, computationally intensive applications are increasingly becoming a part of enterprise data centres, which run multiple types of applications on multiple VMs without taking into account SLA criteria, such as the deadline that results in an under-utilized host. In the case of resource estimation, this factor presents a unique challenge.

The researchers proposed a novel resource management approach for multiple apps running on various VMs that took into consideration various types of SLA criteria [80]. Non-interactive compute-intensive jobs and transactional applications are both addressed by this strategy. Both types of applications had a wide range of SLA criteria and specifications. The main benefit of their work was that they were able to forecast potential insight using historical CPU utilisation data combined with SLA penalties, allowing them to make complex placement decisions in response to shifts in transactional workload and scheduled jobs, taking into account CPU cycles in the event of under-utilisation during normal or off-peak periods. The data was used to train an artificial neural network (ANN) to forecast VM CPU utilisation for the next two hours, with the results shown versus actual usage. The X-axis was supplied every 5 minutes at a regular interval. At this time, this research noticed some flaws in their work: (1) The ANN forecast deviates from the real value in some situations when there is a large variance in preparation; (2) In a few cases, it also predicts low CPU utilisation from the actual value; (3) They did not take into account extremely non-linear data. Non-linearity in workload is a serious issue currently, as data centres have extremely high non-linearity in workload, which results in a range of concerns such as excessive energy consumption, inconsistent QoS, and SLA violations [11].

2.4.8 QoS-aware Resource Provisioning

The pattern of evaluating apps installed on running VMs in modern data centres changes over time, with multiple users attempting to access the application at the same time. As a result, static resource allocation to SaaS apps in the cloud has been found to be wasteful due to non-linear resource utilisation during periods of low demand and high utilisation. When demand is low, available resources are squandered, resulting in high overhead and costs for the cloud service provider; when demand is high, available resources may be insufficient, resulting in poor QoS. This issue can be overcome using dynamic resource provisioning, however, the challenge in this scenario is calculating the optimum number of resources to deploy in a particular period of time to meet QoS requirements when a fluctuating workload is available. This problem is being approached from two perspectives: reactively and proactively. Because it is dependent on future load variations prior to their occurrence, the latter has been extensively adjusted, i.e., estimating the QoS parameters in advance.

A workload prediction model based on ARIMA was proposed by researchers [81]. The expected requests were utilised to dynamically create VMs in an elastic cloud environment while taking into consideration QoS metrics such as response time and rejection rate, which was the main benefit of their work. The accuracy of predicted user requests was also evaluated to examine how it impacted resource usage and QoS factors. This research would like to draw your attention to the following limitation in this work, though. They took historical web request data from the Wikimedia Foundation [82] and fed it into a *Workload Analyzer* component of their proposed model. In this component, the ARIMA model was employed to provide a future estimation for a specific time interval that could be changed for a particular application. For optimal system utilisation, the time interval should be long enough to allow for the introduction of a new VM. If a VM deployment time is smaller than this static time interval, the extra leftover time may affect QoS parameters like response time, which could cause problems.

2.4.9 Varying Patterns of a Service Tenant in Resource Allocation

In a multi-tenant service cloud context, resource demand prediction requires historical data to learn the past profiles of service tenants, which is difficult owing to the necessity to update the prediction model on a frequent basis as service tenants' profiles or trends change. Another challenge is maintaining the amount of resources required by a service tenant to conduct its operations, which is dependent on a number of factors, including (1) the operation type, (2) the specific period during which the operation is conducted, and (3) the service tenant's load at any given time. As a result, it poses a problem because the resource requirements of a service tenant can fluctuate. When dealing with resource provisioning using proactive approaches for a single service tenant as well as multiple service tenants, this is an important problem to handle.

The researchers suggested a dynamic resource demand prediction and provisioning strategy to assign resources in advance in multi-tenant service clouds [83]. They separated the service tenants into categories based on whether their resource usage was likely to increase in the future. As a result, the suggested system prioritised resource demand forecasting for only those service tenants whose resource demand was predicted to increase, lowering the time necessary for prediction, which may affect the overall duration of all operations, consequently affecting QoS. Furthermore, by merging service tenants with matching VMs and distributing them to physical machines, the proposed technique used the Best-fit decreasing heuristic method to assess the efficiency of maximum physical machines (PMs) use. The most important part of this research is that it classifies service renters based on whether resource demand will rise or not, and then forecasts resource demand for tenants whose resource demand will increase, reducing computing time and cost of prediction. However, despite the fact that labeling data is required in order to categorise it using supervised learning approaches, (1) This research is unable to identify on what basis they associate binary (0,1) with the service tenants' attributes. (2) Assuming that the service tenants' features were labeled with binary based on some condition, labeling the data in a large-scale multi-tenant cloud would take time and increase the prediction cost. (3) In a large-scale distributed multi-tenant cloud, some data may be accessible without labels, in which case supervised classification would be useless.

2.4.10 Single ML Model in Energy Consumption Prediction

In offline mode, the bulk of cloud service providers' tools calculate and estimate a host's or a group of hosts' energy usage, but doing so in real-time running applications is difficult. Furthermore, due to the non-linear workload across different hosts, a single ML method cannot be considered capable of doing this task well. A Google cluster or node, does not employ more than 60% and 50% of its CPU and RAM, respectively [2]. As a result, ensemble learning can be an important part of delivering correct predictions in a cloud architecture.

In a cloud computing context, the researchers proposed an ensemble learning approach for forecasting future energy efficiency in virtual machine resources such as CPU utilisation, infrastructure, and service levels [84]. The major benefit of their study is ensemble learning, which uses four different prediction methodologies such as moving average, exponential smoothing, linear regression, and double exponential smoothing. In each time iteration, they forecast the next use of VM resources, such as CPU consumption, and calculate the mean absolute error (MAE) of all iterations to select the best-performing model predictions for measuring and forecasting energy efficiency and ecological efficiency in an IaaS setting in real-time. They do not, however, anticipate measures such as Last-level-cache (LLC) and disc throughput, which have an impact on a host's energy usage at the

VM level [85]. Furthermore, rather than being generalised for all data, the accuracy of the chosen model is workload-dependent, i.e., interactive and batch workloads.

2.4.11 Prediction Accuracy in Auto-scaling of Web Applications

When and how resources are distributed for cloud-based apps is determined by auto-scaling. There are two types of auto-scaling: reactive and proactive. The reactive strategy allocates resources when system events such as CPU utilisation, number of requests, and queue length surpass a preset threshold. The proactive method entails forecasting the quantity of resources needed ahead of time to avoid needless events. Furthermore, proactive approaches include forecasts based on classical statistical time-series analysis, which does not fit all circumstances in terms of accuracy, making it a difficult assignment. Statistical learning also has the following disadvantages:

(1) Statistical learning is based on rule-based programming, which is formalised as a variable relationship.

(2) Statistical learning is based on a dataset with a limited number of attributes.

(3) Statistical learning is based on assumptions like as normality, non-multicollinearity, and homoscedasticity.

(4) In statistical learning, the sample, population, and hypothesis generate the majority of the ideas.

(5) Statistical learning is a math-intensive subject that uses the coefficient estimator and requires a deep understanding of a dataset.

In order to acquire the highest-performing prediction results for web application auto-scaling, a genetic algorithm is employed to fit a proper weight to each time-series prediction model in the system.

(1) Auto-scaling can adapt to any new workload as its characteristics vary over time, which is one of the main benefits of their job.

(2) The sort of prediction models used has no bearing on this method.

(3) Adapting to a range of more advanced prediction models is simple.

However, because of the high time complexity of this strategy, it may impair the response time of any web application housed in cloud infrastructure, resulting in a breach of SLAs.

2.4.12 Time-series Prediction Data

In modern data centres, workload follows a time series trend. As a result, time series prediction models should be trained on historical data, as future trends are assumed to be identical to those observed previously. However, data centres have very non-linear workload fluctuations, which is why new patterns emerge frequently, making the model difficult to understand correctly. Because there is no single model that is adequate for all sorts of time series prediction data, an ensemble technique is being utilised to solve this problem dynamically. Furthermore, most ensemble models for time series prediction are built on a set of fixed predictors, which might be homogeneous or heterogeneous, making it difficult for the models to learn pattern change.

The researchers proposed a new ensemble method for swiftly responding to trend shifts in time-series prediction by dynamically updating the predictors in the ensemble approach [86]. The primary feature of this work is that the ensemble method dynamically modifies the models. It's adjustable, as additional models can be added and removed as needed, based on how well it handles non-linear workload. To evaluate which predictor is working well and which is not, they set a threshold value of 5 and a floor limit of 0. A score is assigned to each predictor, which rises and decreases in proportion to the predictor's outcomes. If this predictor's score reaches the threshold value, it is chosen as a representative predictor; otherwise, it is rejected if it falls below the floor limit. On the other hand, these fixed parameters produce excellent results for their chosen dataset, resulting in a non-generalized strategy.

2.4.13 Data Training

Virtual resources such as virtual CPUs (vCPUs) and memory (vRAMs) exhibit non-linear resource demand in modern cloud systems, resulting in complex resource utilisation behaviour. As a result, with such a high volume of work on a daily basis, virtual resource performance improvement is essential. Large firms like Amazon, Alibaba, and others have failed on occasion due to a lack of resource management strategy. As a result, forecasting virtual resources (such as vCPU and vRAM) is difficult. Furthermore, resource forecasting poses some difficulties:

(1) resource prediction should be dynamic in order to respond to changing workload patterns over time;

(2) training data should be chosen in such a way that it has the greatest impact on

the target variable so that the model can learn to predict it effectively.

The researchers developed a model that takes a range of criteria into consideration in a virtualized platform to anticipate virtual resources with the fewest SLA breaches [87]. This strategy used a Bayesian approach to identify numerous factors and take the best training data into account. The main benefit of their work is that it finds changeable dependencies in a systematic manner using non-linear workloads from multiple data centres such as Amazon, EC2, and Google. However,

(1) they do not account for the combination of multiple application types,

(2) this approach lacks generalisation because it relies on the dependencies of a specific problem,

(3) this method ignores high-level metrics such as transaction throughput and latency of underlying resources, such as vCPU cores, for prediction.

2.4.14 VM Multi Resources

In cloud data centres, flexible resource provisioning frameworks are required to manage host load based on diverse requirements. As a result, data centres use prediction models to anticipate the number of resources required in advance for variable workloads throughout time. Its goal is to use historical usage trends to forecast future VM request workloads. However, because VM requests comprise a range of virtual resources such as CPU, memory, disc space, and network throughput, forecasting demand for each type of resource independently is highly difficult and complex. The multi-resource presence of a VM poses a unique issue when picking an ML prediction model. Furthermore, various cloud users can make distinct cloud resource demands. As a result, it's impossible and unrealistic to anticipate demand for each type of resource.

The researchers provided a method for categorising VM clusters and building prediction models for each cluster [88]. The main advantage of their research is that (1) they employ Extreme Learning Machines (ELMs), which can discover the best weight for the predictor in a single step. (2) Gradient-based learning methods such as NN and ANFIS use ELMs to circumvent issues such as halting conditions, learning rate selection, learning period size, and local minimums. (3) Because it works with nonlinear processes, this study can handle the LR method's linear behaviour. (4) It uses a single network to predict VM demands in each cluster. (5) Each cluster has the ability to have its own prediction network. They set the number of clusters to 3 in K-means clustering, resulting in a model with a fixed number of VM clusters.

2.5 Future Research Directions

2.5.1 Performance and Online Profiling of Workload

Many elements influence the efficiency of an intelligent resource management system, including the prediction model's accuracy and time complexity. Large firms like Google, Microsoft, Amazon, and others are in charge of incredibly complex data centres that handle a diverse set of workloads. As a result, in the context of such a highly changing or nonlinear workload for VMs, a more accurate assessment of prior workload through the use of more advanced ML and DL modes is a future research topic. In addition, an algorithm's time complexity is a measure of its performance in terms of the time it takes to run the input code. As a result, the method should be constructed to have as little temporal complexity as possible. Furthermore, online profiling is required to avoid VM blackouts till they are running in development, as well as other resource utilisations such as CPU and memory, which are important contributors to physical resource exhaustion and should be taken into account when predicting. The researchers performed online workload profiling and analysis to identify whether a virtual machine is interactive or delay-insensitive [9,26]. They employed supervised classification to divide VMs into these two groups. Semi-supervised learning [89] may play a critical role in this case, and it may be a viable research direction for training data with these partial labels and performing classification with high accuracy in large-scale distant data centres.

2.5.2 Multiple Resource Usage in VM Consolidation

During the VM consolidation phase, a host is considered overcrowded if CPU utilisation reaches a throughput threshold, such as 80% [78]. Other resource usage, such as memory and bandwidth usage [90], however, results in host overloading. As a result, at the VM consolidation phase, detecting overloaded hosts using a combination of CPU, memory, and bandwidth usage is a possible study direction. The estimation of current and future CPU, memory, and bandwidth usage should be addressed for an efficient VM consolidation operation. The present research uses a range of ML algorithms, such as linear regression and multiple regression, in which the model is trained using numerous features to simulate a target variable, such as CPU use [69,90]. The VM migration time in the VM consolidation process is affected by the training time of numerous features, which has an impact on QoS and SLAs in large-scale distributed data centres with millions of VMs in production. As a result, dealing with the training duration of ML models could be a promising study topic in the future. Different deep learning (DL) approaches, such as Long Short-Term Memory (LSTM) networks [91] and Gated Recurrent Unit (GRU) [92], can reduce training time by avoiding multiple feature overheads by using a single feature, such as a vector of CPU utilisation, as an input for training to predict its next state in the future.

2.5.3 Cloud Network Traffic

When considering current resource utilisation in VM allocation on a host, the problem of varying patterns of distinct types of workloads is a hurdle. As a result, forecasting future resource consumption, such as CPU and network bandwidth, has shown to be a viable alternative [75]. However, disc throughput is an important consideration in addition to these resources. Taking disc throughput into account in VM placement heuristics is a new research direction. It works out how much data can be saved, read, and written each second. According to researchers, disc tail latency, particularly reads, is a critical element for delivering online services where a user is waiting for a response [76]. As a result, disc throughput might affect VM migration time, causing tail latency to increase and SLA violations. As a result, a prior maximum estimate of disc throughput will be crucial in preventing delays.

2.5.4 Host Temperature

The researchers developed a scheduling technique to minimise host temperature, which was based on a host temperature prediction generated using numerous ML algorithms [93]. As a result, anticipating host temperature can aid thermal management decisions such as VM migration to lower host temperature (i.e., CPU temperature). The ambient temperature, which is a combination of CPU and inlet temperature, was taken into consideration for prediction by [77]. It's possible that this will result in an increase in algorithm overhead. Furthermore, they discovered that CPU load and power consumption have the greatest impact on the host's CPU temperature. As a result, the host is waiting for the CPU to get overloaded, causing the temperature to rise and incurring additional cooling costs. Prior CPU estimation-based resource provisioning, as a prospective future research topic, can prevent the CPU from getting overloaded and conserve energy. Furthermore, due to the training of many features, several ML methods require a large amount of training time, which might slow down VM migration. It will postpone VM migration, which will slow down host temperature decline and increase costs. As a result, adopting an ML or deep learning method like GRU, where the inlet temperature is utilised as an input to train a model that can predict its future state using single-feature training, could be an option. This will avoid an overhead algorithm, a VM migration delay, and a delay in lowering the host temperature.

2.5.5 False Host Overloaded Detection

The static threshold for overloaded host detection can lead to unreliable VM migration. There is no need to move a VM if its resource use degrades in a short period of time. In this situation, the method should have a dynamic resource utilisation threshold that stops VM migration when it hits a set threshold, taking into consideration data from the near future. This is the next research direction for efficient VM migration in VM consolidation. Furthermore, VMs should be transferred if there is a long period of load decrease in the near future.

2.5.6 Energy Metering at Software-level

Visibility of energy usage at the host and VM levels will help with many power management decisions, such as power capping. Energy consumption is easy to predict or calculate at the host level because modern data centres have several built-in sensors that track it, but it is difficult to measure at the VM level because to measure memory-induced energy consumption, this research must collect LLC (last-level-cache) events raised by each VM on each core, which is difficult to do [5,85]. Clustering analysis can be used to assess the status of VMs in terms of energy consumption, such as low, moderate, or critical, rather than calculating or estimating energy consumption at the VM level. As a result, partitioning VMs by doing clustering analysis based on highly co-related variables with energy usage at the VM level is a possible research topic, and no host-level information would be required. To determine the link with energy use, ML approaches such as ChiSquare Score, Fisher Score, Gini Index, and Correlation-based Feature Selection (CFS) can be utilised [94]. A clustering analysis can then be performed using a clustering algorithm or a clustering ensemble [95] to discover which VMs are in low and essential energy consumption phases. A collection of VMs can be handled together in the resource management system of a data centre, potentially lowering response time and increasing QoS.

2.5.7 SLA-based VM Management

Dynamic resource provisioning and dynamic VM consolidation, which take into account various types of VM resources such as CPU, memory, and bandwidth, current and future resource needs, and SLAs such as compute-intensive non-interactive jobs and transactional applications, are two future research directions for avoiding non-linear resource utilisation in modern data centres. Both of these approaches rely largely on resource prediction accuracy. For example, the researchers gave long-term CPU utilisation projections that deviated significantly from actual test phase data due to a considerable shift in CPU utilisation during the training phase, which is crucial for dealing with non-linear utilisation in modern data centres [80]. Hyperparameters utilised in Artificial Neural Network (ANN) learning, such as mini-batch size, epochs, and amount of neurons, will be optimised in a future study. If the model is trained on the data in an optimised manner, it is said to work better. When both plots begin moving closely and consistently, it may signal that the model has learned a lot, and learning should be stopped at these optimised hyperparameters.

2.5.8 QoS-aware Resource Provisioning

The goal of this research is to improve QoS metrics such as response time and rejection rate by adopting constructive dynamic resource provisioning based on workload estimation using historical data. Future studies could focus on reacting to it, with resource supply taking place after resource demand, such as the number of requests, has arrived. Furthermore, according to the current study, adhoc decisions in dynamic resource provisioning can help to decrease request prediction error, which can help to improve low QoS efficiency [81]. Furthermore, there is a potential research direction to forecast peak CPU use using more sophisticated ML models such as XGBoost [96], LSTM [91], and GRU [92] in a correct manner that cannot be equipped with the ARIMA model. Furthermore, no single ML algorithm can handle any non-linear workload with time-series data, necessitating an ensemble learning strategy in the future, where different ML and DL approaches might be employed. The best-performing model can then be chosen for possible application. As explained in Section 2.4.8, The researchers predict web requests based on a static time period that can affect response time [81]. As a result, it can be handled by anticipating future web requests with a dynamic time interval that varies dynamically dependent on the time it takes for the VM to deploy. If the VM deployment time is considerably lower than this static time interval that influences the QoS parameter like the response time, the estimation time interval can be similar to the VM deployment time, and the remaining time can be saved. To satisfy the criterion of equivalence with the estimated time of the request prediction, a prior calculation of VM deployment time based on historical data should be generated and applied in the above-mentioned situation.

2.5.9 Varying Patterns of a Service Tenant in Resource Allocation

As a future research direction, clustering analysis, which does not require any data labelling, could be utilised to classify service tenants. Similar patterns of service tenants can be automatically obtained based on previous resource demands. Service tenants with high and low resource demand can be differentiated using clustering approaches, and predictions for those with high resource demand can be offered using ML and DL regression techniques. In the case of a distributed data centre where data is dispersed and partial labels are available, a concept known as semi-supervised clustering [97] can be used, in which unsupervised data is given a little supervision using partial labels and techniques such as instance-level constraints [98] and relative distance constraints [92].

2.5.10 Single ML Model in Energy Consumption Prediction

A system power model contains memory, disc, and network components in addition to the CPU, therefore these components could be evaluated as well. The current research focuses on the linear link between these measurements and energy usage; however, nonlinear relationships, such as polynomial or exponential relationships, could be investigated in the future. Furthermore, the best individual model is picked in an ensemble learning approach, which may or may not be the optimal solution. Another alternative is to aggregate and analyse the information offered by each separate model. This can be done by estimating the average using weights depending on the mean average error of each individual prediction. In addition, each workload type necessitates a unique set of configuration parameters. The goal of future research is to maintain track of the model parameters that have raised the maximum resource utilisation in the past and apply them in real-time scenarios to adapt the models to the workload type of each unique VM. Furthermore, a quick shift in resource usage has an impact on forecast accuracy. As a result, another potential research topic is to provide the ML model with average workload performance, such as CPU usage.

2.5.11 Prediction Accuracy in Auto-scaling of Web Applications

ML models, rather than statistical methods, can be used to forecast future workload, which has a number of benefits: (1) Without the requirement for explicit programming, ML learns from data. (2) ML can learn from billions of observations and features, and (3) ML is less reliant on assumptions and, in most situations, ignores them. (4) Predictions, supervised learning, unsupervised learning, and semi-supervised learning are all highlighted in ML. (5) ML identifies patterns in a dataset through iterations, requiring significantly less human work. To forecast the target variable, numerous features must be trained, which raises the time complexity of ML approaches like regression. When processing multiple features, ML approaches suffer from latency and computational complexity issues due to the presence of redundant information. The number of functions, feature dependencies, number of records, feature types, and nested feature categories all contribute significantly to the processing time of ML approaches in such datasets. As a result, future research should focus on applying appropriate feature selection approaches, such as wrappers, filters, embedded methods, and upgraded versions [99], to effectively overcome the computing speed versus accuracy trade-off when processing big and complicated datasets.

2.5.12 Time-series Prediction Data

A future research goal is the development of a generic ensemble framework for any form of dataset in cloud time series workload data. In general, deep learning (DL) is a rapidly growing and extensive study topic involving unique architectures. Researchers, on the other hand, never know when they'll have to adapt which methodologies to which conditions. Global neural network models, which are prone to outlier errors in some time series, were employed by researchers [100]. As a result, unique hierarchical models containing both global and local characteristics for specific time series must be devised. Ensembling, which involves training numerous models with the same dataset in different methods, can be integrated with these models. Furthermore, while CNNs have long been used to analyse images, they are also being utilised to forecast time series data. Traditional RNN models are inefficient at predicting seasonality in time series forecasting [101,102]. As a result, they use a proprietary attention score mechanism for long-term dependencies and CNN filters for local dependencies. The researchers have also used recurrent skip connections to capture seasonality patterns [101]. Dilated Causal Convolutions were developed by researchers [103] to effectively capture long-range dependencies throughout the temporal dimension. They've lately been utilised in combination with CNNs to handle time series forecasting difficulties. As more advanced CNNs, Temporal Convolution Networks (TCN) have been introduced, which integrate dilated convolutions and residual skip connections [104]. TCNs are prospective NN architectures for sequence modeling tasks, according to [105] TCNs are promising NN architectures for sequence modeling tasks, in addition to being efficient in training. As a result, forecasting practitioners may gain a competitive edge by using CNNs rather than RNNs. As a result, these advanced neural networks could be utilised to forecast workload time series in cloud infrastructure in the future.

2.5.13 Data Training

The goal of optimising ML hyperparameters is to determine the hyperparameters for a specific ML algorithm that results in the best validation data results. In contrast to the model parameters, the engineer sets the hyperparameters before the training. A hyperparameter is, for example, the number of trees in a random forest, whereas the weights in a neural network are model parameters gained during training. Support vector machine hyperparameters (SVM) and k in k-nearest neighbours (KNN) are size and decay, respectively. Furthermore, by discovering a combination of hyperparameters, hyperparameter optimization yields an optimal model that decreases a preset loss function and, as a result, increases the accuracy of given independent data. Hyperparameters can thus have a direct impact on the training of ML algorithms. Understanding how to optimise them in order to get optimal performance is consequently crucial. This suggests that optimising the hyperparameters of ML algorithms for effective dataset training is a potential research focus. Grid Search, Random Search, Bayesian Optimization, Gradient-based Optimization, and Evolutionary Optimization are some of the common heuristics that can be used to accomplish this [106].

2.5.14 VM Multi Resources

There is a future research direction to categorise the VMs and construct a prediction model for each cluster to meet the multi-resource demand concerns, as described in the above sections. However, utilising a clustering technique like K-means can limit the number of clusters accessible, resulting in a VM being placed in the wrong one. Because it seeks to combine numerous clustering techniques to achieve a final consensus solution that is more robust and accurate than a single clustering algorithm, a clustering ensemble may be a superior approach to clustering [107]. A number of clustering ensemble approaches are mentioned in the literature [108]. In addition to clustering accuracy, two other evaluation criteria such as time complexity and resource use (CPU and memory utilisation) were examined in a recent work [95] to evaluate the novel clustering ensemble. As a result, improved clustering algorithms like clustering ensembles will be employed in the future to produce the best clusters with the highest precision, the shortest time complexity, and the smallest resource usage.

2.6 Summary

The challenges of machine-learning-based resource management in a cloud computing environment are discussed in this chapter, as well as the numerous ways that have been utilised to address these challenges in recent years, as well as their benefits and downsides. The amount of studies looking at ways to apply ML techniques to undertake workload prediction, energy consumption prediction, and other tasks has increased dramatically in recent years. These strategies employ a variety of ML methods to address a variety of issues. Finally, new prospective future research topics are given to strengthen the current ML approaches for resource management in cloud-based systems, based on the problems and disadvantages revealed in the state-of-the-art study. This chapter's overall expertise helps cloud researchers understand cloud resource management and the importance of ML approaches.

ML models can be employed in cloud computing systems to achieve various optimization goals and deal with challenging jobs, according to the research. The adoption of ML technologies also brings up new possibilities for resource and application management. The progress of ML methodologies in current research is illustrated in this article, which helps readers comprehend the research gap in this topic. One possible strategy to increase system efficiency is to undertake intelligent resource management using advanced ML techniques such as reinforcement learning and deep learning.

Chapter 3 A Cluster Ensemble based on Single Clustering

3.1 Outline

Clustering is a common way of classifying system states in cloud computing. Numerous cluster ensemble approaches have been created recently, however they still have certain drawbacks. The ensemble generation step and the consensus function of the clustering ensemble technique frequently employ distinct clustering algorithms, which creates a compatibility problem in terms of how well the various clustering algorithms function. The end results' accuracy in a clustering ensemble method is also a crucial consideration. This study suggests a unique cluster ensemble method based on a single clustering algorithm (CES) to address it. Due to the affinity propagation (AP) clustering algorithm's inherent property of producing a random number of clusters, this study iterates AP ten times in this method's ensemble generation step to create different base partitions with a high level of diversity in each iteration. To overcome it, this study proposes a special cluster ensemble method based on a single clustering algorithm (CES). This study iterates affinity propagation (AP) 10 times in the ensemble generation step of this method to produce various base partitions with a high level of variety in each iteration due to the affinity propagation (AP) clustering algorithm's inherent attribute of producing a random number of clusters. The same technique AP is also used to suggest a novel consensus function for fusing these base partitions into a single partition with a few adjustments. Using pairwise constraints with AP and the number of clusters in a dataset, the proposed consensus function makes use of sparse side information in the form of partial labels. By using this data, AP is forced to create an actual number of cluster centres rather than a haphazard number of clusters, greatly improving the accuracy of the results. In order to produce the appropriate number of clusters in the final partition of a dataset, CES leverages the same clustering functionality in both stages of the proposed cluster ensemble approach, which considerably improves accuracy when compared to state-of-the-art cluster ensemble methods. The CES performs better than AP in terms of accuracy and execution time as a result of these improvements. Studies using actual datasets from a variety of sources reveal that CES improves accuracy while using 44.60% less execution time than AP and modern cluster ensemble approaches, respectively, by an average of 5% and 55.54%.

3.2 Introduction

Clustering is an unsupervised learning technique for dividing a set of data items into related classes [109–111]. It is a crucial and challenging subject in data mining and ML, and it has been successfully applied in a wide range of fields, including image processing [112], recommender systems [113], text mining [114], and pattern recognition [115]. In recent years, a variety of methodologies have been employed to produce a huge number of clustering algorithms [116]. For a given dataset, different methods may produce drastically varied clustering results. Each clustering approach comes with its own set of benefits and drawbacks. No single algorithm, on the other hand, is adequate for all datasets or applications. Even with a specific algorithm, selecting the appropriate parameters for the clustering process might be challenging.

A single clustering algorithm has traditionally been employed to produce a single clustering result, which has a high rate of error. Cluster ensemble is a new technique for integrating several clustering results (from different clustering methods or the same approach with different iterations) into a potentially superior, more resilient, and single partition [117]. In detail, a cluster ensemble has two stages: the ensemble generation phase gets several base partitions, and the consensus function merges these base partitions [118]. When compared to discrete clustering techniques, a functional clustering ensemble should give reconcilable and well-grounded clustering results. However, while constructing an ensemble for clustering, there were some different and hard issues to cope with, and it was not as simple as this interpretation suggests. Cluster ensemble is gaining popularity, and several algorithms have been proposed in recent years [107, 119]. In terms of resilience, innovation, stability, and confidence estimation, as well as parallelization and scalability, cluster ensembles outperform single clustering algorithms [120]. Despite its enormous progress, the current research continues to encounter significant obstacles. They all have the same flaw: to get base partitions and a final partition, the existing cluster ensemble approaches use different clustering algorithms in both stages. Furthermore, the usage of various clustering methods in both stages of the existing cluster ensemble design may cause issues with working functionality compatibility. This prompted us to employ a single clustering method in both stages of the new cluster ensemble design, which increased the accuracy of the final results dramatically. As a result, this study offers a new cluster ensemble approach that uses the same clustering in both stages of the process. As a result, in the first stage of the ensemble generation process, multiple base partitions are obtained by running an unsupervised clustering algorithm affinity propagation (AP) ten times, which provides a high level of diversity among base partitions in each iteration because it generates a random number of clusters [121]. Furthermore, it collects all available diverse information about a data set, which may aid clustering efficiency. Then, known as cluster-based similarity, a similarity matrix is calculated between these basis divisions [107]. The generated similarity matrix is then supplied as a parameter in the novel consensus function proposed in the cluster ensemble method's second step, which employs the same clustering algorithm AP but with some alterations. Furthermore, this research uses pairwise constraints [98] that employs the concept of must-link (two objects must be in the same cluster) and cannot link (two objects cannot be in the same cluster) with the same clustering algorithm AP to provide a little supervision to the computed similarity matrix in the proposed consensus function. This supervised little information is then added to the computed similarity matrix, which aids in boosting clustering efficiency. The similarity matrix is updated with the Gram matrix at this point, which improves clustering efficiency. Furthermore, as previously mentioned, AP has a problem in that it generates random number clusters. As a result, the number of clusters produced by AP is restricted to the number of classes in a dataset. When this proposed consensus function was applied in the proposed cluster ensemble approach, this novel innovation in AP served to greatly raise the accuracy of the final outcomes. As a result, the cluster ensemble method's suggested innovative consensus function merges the basis partitions into a single partition. Because this research applies the same functionality in each stage of the proposed technique, this research calls it “A Novel Cluster Ensemble based on a Single Clustering Algorithm (CES)”, as shown in the left of Figure 3-1. The main advantage of CES is that it reduces the burden of employing two different clustering paradigms in both stages, making it compatible and enhancing clustering outcomes such as accuracy over current cluster ensemble approaches. Furthermore, when compared to AP, the unique change improves accuracy and execution time greatly.

The contributions of this chapter are given below:

(1) This research proposes a new cluster ensemble approach based on a single clustering algorithm, whereas traditional cluster ensemble methods use distinct clustering algorithms in both stages, resulting in ensemble creation and consensus function compatibility issues.

(2) This research proposes a novel AP-based consensus function that combines pair-

wise constraints, the Gram matrix, and AP limits to produce the actual number of clusters in the dataset.

(3) In terms of accuracy and execution time, the proposed cluster ensemble technique outperforms AP.

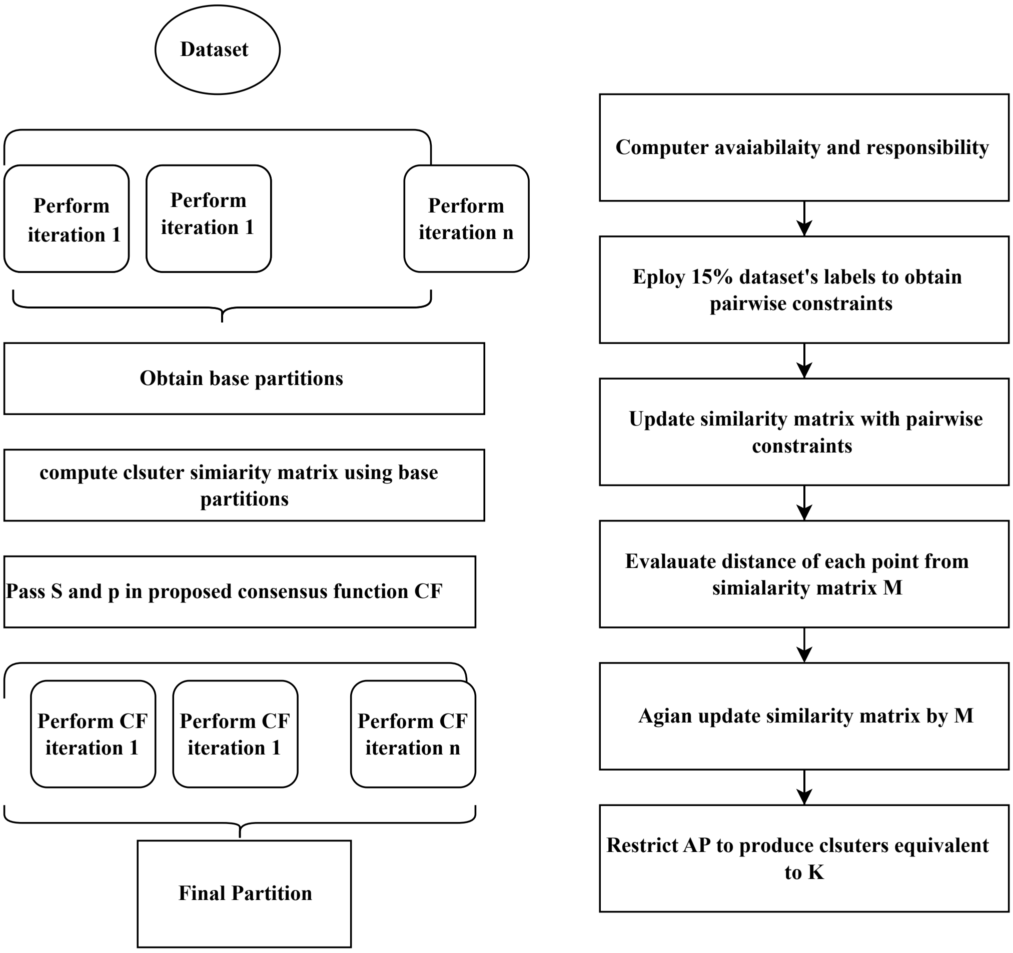

Dataset
Perform iteration 1
Perform iteration 1
Perform iteration n
Obtain base partitions
compute clsuter simiarity matrix using base partitions
Pass S and p in proposed consensus function CF
Perform CF iteration 1
Perform CF iteration 1
Perform CF iteration n
Final Partition
Computer avaiabilaity and responsibility
Eploy 15% dataset's labels to obtain pairwise constraints
Update similarity matrix with pairwise constraints
Evalauate distance of each point from simialarity matrix M
Agian update similarity matrix by M
Restrict AP to produce clsuters equivalent to K

Figure 3-1 Proposed Methodology: Left is Proposed Cluster Ensemble (CES) and Right is Proposed Consensus Function (CF)

3.3 Related Work

Using a consensus function, a clustering ensemble merges numerous base partitions obtained in the ensemble generation step into a robust, accurate, and single partition [119]. Cluster ensemble has the advantage of increasing the accuracy of the results by accounting for individual solution biases. The researchers proposed the first three cluster ensembles in 2002 [122]. The first was the cluster-based similarity partitioning algorithm (CSPA), which was based on data point similarity *S*, with *S* varying depending on whether the data points were similar or different. The second approach was the hypergraph partitioning algorithm (HGPA), which was based on re-partitioning data using provided clusters. The meta-clustering algorithm (MCLA) was the last one, and it was based on clustering clusters and rendering each cluster with a hyperedge. The Adaptive Clustering Ensemble (ACE) was proposed by the researchers [107] and consisted of three stages: the first was to convert the basis clusters into binary representations. The second stage involved finding similar clusters based on cluster-based similarity, and the third involved dealing with uncertain objects to produce consensus function outcomes in order to create superior final consensus clustering partitions of data. In addition, numerous new cluster ensembles have recently been suggested, such as the quad mutual information consensus function (QMI) and the mixture model (EM) [120]. QMI is a quadratic mutual information-based consensus function that has been suggested and reduced to k-means clustering in the space of carefully adjusted cluster labels. EM is an unsupervised decision-making fusion method that uses a probability model of the consensus partition in the space of contributing clusters to make decisions. The weighted spectral cluster ensemble (WSCE) was proposed by researchers in 2015 as a new cluster ensemble focusing on group identification arena and graph-based clustering principles [119]. A proposed consensus function is used to integrate many base partitions into a single robust partition using a new version of spectral clustering. The researchers developed a cluster ensemble method based on distribution cluster structure, with final results generated using a distribution-based normalised hypergraph cut methodology [123]. The researchers presented two new cluster ensemble methods: ensemble clustering using a hierarchical consensus function to propagate cluster-wise similarities (ECPCS HC) and ensemble clustering using a meta-cluster-based consensus function to propagate cluster-wise similarities (ECPCS MC) (ECPCS MC) [124]. Some research has focused on the applications of cluster ensembles in many fields. For example, in the field of pattern recognition, time series analysis has become a hot research topic, particularly for detecting manufacturing faults. As a result, researchers suggested an automated alternative based on consensus clustering dubbed control chart pattern recognition (CCPR) [125]. The researchers also developed a cluster ensemble approach for unsupervised pattern identification that focused on the evolution of damages in composites under solicitations [126]. The researchers offered a novel approach with the best pure results and a quick implementation time. It also enhanced accuracy. Rapid Clustering with Semi-supervised Ensemble Density Centres is the name of this model [127]. The researchers developed a clustering ensemble approach via structured hypergraph learning, i.e., the hypergraph is dynamically learned from base results rather than being generated directly, which will be more reliable. Additionally, this study enforces the hypergraph's unambiguous clustering structure during dynamic learning, making it more suitable for clustering tasks and eliminating the need for any uncertain postprocessing, such as hypergraph partitioning [128]. In an effort to offer a new approach to the cluster ensemble problem and apply knowledge granulation to ensemble learning, a hierarchical cluster ensemble model based on knowledge granulation is proposed by the researchers [129].

Table 3-1 The Important Notations Used in this Chapter

Definition	Symbol/Notation
Dataset	D
Data object	$x_i \in D, 1 \leq i \leq n$
Number of objects	n
Number of ensemble members	m
Ensemble member	$\beta_i, 1 \leq j \leq m$
Similarities between objects	$S_{ij}, 1 \leq i \leq n, 1 \leq j \leq n$
Distance from similarity matrix	$P_{ij}, 1 \leq i \leq n, 1 \leq j \leq n$
Euclidean distance	d_{euc}
Similarities between ensemble members	S_m
Preference parameter for ensemble members	p_m

The following notations will be used consistently in this chapter. Table 3-1 also contains several important notations with their definitions that were used in this article. This research calls a set of objects $D = \{x_1, x_2, ..., x_n\}$, where each object $x_i \in D$ is represented by a vector of N attribute values $x_i = (x_{i,1}, ..., x_{i,N})$. Let $\Gamma = \{\beta_1, \beta_2, ..., \beta_m\}$ be a cluster ensemble with m base partitions, where each base partition is an "ensemble member", and returns a set of clusters $\beta_h = \{\beta_1^h, \beta_2^h, ...\beta_n^h\}$, such that $\bigcup_{p=1}^{k_h} \beta_p^h = D$,where k_h is the number of h^{th} clustering. For each data point $x_i \in D, \beta^h(x_i)$ indicates cluster label in the g^{th} base partition to which data point x_i belongs to, i.e. $\beta^h(x_i) = \beta_h^p$, if $x_i \in \beta_h^p$. As a result, the problem is to find a new partition $\Gamma^* = \beta_1^*, \beta_2^*, ...\beta_K^*$, where K is the number of clusters in the final clustering result of the dataset D, which summarises the details from the cluster ensemble Γ [108].

The proposed cluster ensemble method's operation is described in more detail below. Algorithm 3-1 presents the pseudo-code of CES.

Algorithm 3-1 The pseudo code of the proposed cluster ensemble method CES

```
Require: data, No. of clusters $K$
Ensure: the clustering Outcomes $\Gamma^*$
 1: $no\_classes \leftarrow K$, $random \leftarrow [\,]$, $temp \leftarrow [\,]$, $O \leftarrow [\,]$, $s \leftarrow [\,]$ $Z \leftarrow [\,]$, $idx \leftarrow [\,]$,
    $status \leftarrow [\,]$, $availability \leftarrow a_{ik}$, $responsibility \leftarrow r_{ik}$
 2: Calculate $m$ base partitions $\beta_i$ by executing AP ten times
 3: $S_m \leftarrow Euclidean(\beta_i, \beta_i)$ where $S_m$ is similarity matrix
 4: $p_m \leftarrow min(S_m)$ where $p_m$ is preference parameter
 5: Pass $S_m$ and $p_m$ in proposed consensus function (modified AP, execute consensus
    function ten times)
 6: Compute $a_{ik}$ and $r_{ik}$
 7: $s \leftarrow .15(labels)$
 8: for $i$ = 1 to length(s) do
        for $j$ = $i$ + 1 to length(s) do
            if $(x_i, x_j) \in C$ then
                $status \leftarrow 0$
            else
                $status \leftarrow 1$
            where $C$ denotes cannot-link constraints
 9: return $status$
10: $S_{ij}$ & $S_{ji}$ = $status$ where $i \in (1, ..., n), j \in (1, ..., n)$
11: $P_{ij} \leftarrow \frac{S_{1j}^2+S_{i1}^2+S_{ij}^2}{2}$ where $i \in (1, ..., n), j \in (1, ..., n)$
12: $S_{ij} \leftarrow P_{ij}$ where $i \in (1, ..., n), j \in (1, ..., n)$
13: $Z \leftarrow$ set of exemplars
14: $Z \leftarrow Sort(Z, descending)$
15: if $length(Z) < no\_classes$ then
        $no\_classes \leftarrow length(Z)$
16: $random \leftarrow Random(length(Z), no\_classes)$
17: $O \leftarrow Z[random]$
18: for $i$ = 1 to $no\_classes$ do
        for $j$ =1 to length(Z) do
            $temp \leftarrow Z[j]$
        if $temp$ = O(i) then
            $idx \leftarrow temp$
19: return $idx$
20: $\Gamma^* \leftarrow idx$
```

3.4 First Stage: Ensemble Generation Step

Ensemble creation is the first phase, and the main aim is to produce *m* base clustering members. The ensemble generation stage is represented by steps 2 to 5 in the method 3-1 [107]. To create ensemble members, any clustering algorithm can be utilised as long as it creates as many different members as possible. Using independent runs of various clustering methods or the same clustering algorithm, multiple partitions of the same dataset can be produced at this step [117, 124, 130]. Then, in the following stage, a consensus function is used to obtain a final partition from the base partitions generated in the previous stage. Accordingly, this research uses unsupervised AP, as described in Section 3.5.1, and runs it $(iter = 10)$ times to create multiple m ensemble members, such that $\beta_i \in \Gamma$, where $i \in (1, ..., n)$ and n are the number of data objects. The reason for AP's popularity is that it generates a random set of exemplars (clusters) in β_h, where β_h is an ensemble member, which provides a high level of diversity among ensemble members in each iteration and acquires all possible distinct information about a data set, potentially improving clustering performance. In other words, AP delivers unique clusters in each iteration, assuring the basis of ensemble clustering, which is that ensemble members should have a high level of variety to capture all of the data in a dataset [107].

As a consequence, this research merges the m base partitions found in Section 3.4 using this approach. To compute similarities between pairs of ensemble members, this research utilises the Euclidean distance, as explained previously in Equation (5-2). Cluster-based similarity refers to the commonalities between ensemble members. As a result, the basis partitions are determined as similarities between m ensemble members, and these base partitions are then clustered together using the proposed consensus function in Section 3.5.2. For this, the proposed consensus function uses parameters S_m and $pm = min(S_m)$, which is recommended using AP.

3.5 Second Stage: Consensus Function

Another essential component of the cluster ensemble technique is the consensus function, which is responsible for obtaining the final partition of the data utilising base partitions created during the ensemble creation stage. Because the consensus function has a direct impact on the performance of the cluster ensemble technique, this research proposes a very effective and efficient consensus function, as detailed in the sections below. The consensus function step is represented by steps 6 to 20 in the method 3-1. Rather than computing similarities between data objects, the fundamental concept behind presenting a novel consensus function is to calculate cluster-based similarities between pairs of ensemble members or clusters [107]. The operation of the proposed consensus function is explained further below. This research gives basic information regarding the classic clustering method AP in Section 3.5.1, and then explains how it is enhanced and utilised in proposing the consensus function in Section 3.5.2.

3.5.1 Affinity Propagation (AP)

Affinity Propagation (AP) [121] is a clustering algorithm that works on the principle of message passing between data objects. Unlike other clustering algorithms such as k-medoids or k-means, AP does not seek to determine the number of clusters before running the algorithm. AP, like k-medoids, seeks “exemplars”, or members of the input set that are representative of clusters. In other words, rather than taking the number of clusters K as input, AP takes the collection of real-valued similarities S_{ik}, which indicates how well the data object at index k is suited to be an exemplar for the data object i for two data objects $(x_i, x_k) \in D$. In addition, AP accepts real numbers S_{kk} as input, with the possibility of selecting high-similarity data objects as exemplars (number of clusters), referred to as preference p. The exemplars are influenced not only by p but also by message passing. This value can be changed to generate a different number of clusters. Moreover, this value can be a median of the input collection of real-valued similarities that yields a moderate number of clusters or a minimum of these that yields the fewest clusters. Additionally, two real-valued messages which are the “responsibility” r_{ik} from data object x_i to x_k that depicts how well deserved the data object x_k is to serve as the exemplar of data object xi and the “availability” a_{ik} from data object x_k to x_i that depicts how suitable it would be for data object x_i to select x_k as its exemplar, are computed. r_{ik} and a_{ik} can be considered as log-probability ratios. Initially, availabilities a_{ik} were set to zero: $a_{ik} = 0$. The responsibilities r_{ik} are then computed using Equation (3-1).

$$r_{ik} \leftarrow S_{ik} - \max_{k' \; s.t. \; k' \neq k} \{a_{ik'} + S_{ik'}\} \tag{3-1}$$

Because a_{ik} is set to 0 in the first iteration, r_{ik} has been assigned the difference of s_{ik} and the largest of the similarities between the data object at index i and the other candidates. As a result, if some data objects are assigned to exemplars in subsequent iterations, their availabilities a_{ik} fall below zero, as shown by the Equation (3-2). These negative availabilities will have an effect on the similarities $S_{ik'}$ in Equation (3-1), and the corresponding exemplar will be removed from the competition. And in the Equation (3-1), for $i = k$, the responsibilities become r_{kk}, which is equivalent to input preference and the point at indexed k or i is chosen as an exemplar. This condition allows other candidate exemplars to compete to be an exemplar for a data object and updates availabilities using Equation

(3-2) below.

$$a_{ik} \leftarrow \min\left\{0, r_{kk} + \sum_{i' \ s.t. \ i' \notin \{i,k\}} \max\left\{0, r_{i'k}\right\}\right\} \tag{3-2}$$

Thus, in Equation (3-2), availabilities a_{ik} are assigned to the sum of self-responsibility r_{kk} and positive responsibilities received by the candidate exemplar at index k from other data objects. Only positive responsibilities are added here because it is required for a good exemplar. If self-responsibility becomes negative, the availability of data objects at index k can be increased, and self-availability a_{kk} is updated using Equation (3-3).

$$a_{kk} \leftarrow \sum_{i' \ s.t. \ i' \notin k} \max\{0, r_{i'k}\} \tag{3-3}$$

As a result, these messages are exchanged between two data objects with pre-computed similarities. At any point, availabilities and responsibilities can be combined to identify a potential exemplar. As a result, $(a_{ik} + r_{ik})$ should be the maximum to determine which data object at index i should be chosen as an exemplar. Knowing $i = k$ leads to knowing the data object that is an exemplar for the data object at index i.

3.5.2 Proposed Consensus Function

This research use limited side-information in the suggested consensus function, such as pairwise constraints [98], which are made up of two constraints: must-link and cannot-link. It has aided in improving accuracy and precision. This research assumes that partial class information is provided in the form of pairwise constraints showing whether two objects are members of the same ($must - link$ constraint) or different ($cannot - link$ constraint) clusters. The cluster information is expressed via a set $\Psi \subset D \times D, m_l = \{x_i, x_j\}$ where $\Psi = M \cup C$, is a set and

$$\begin{aligned} M &= \{(x_i, x_j) \in D \times D : x_i \text{ and } x_j \in \text{same cluster}\} \\ C &= \{(x_i, x_j) \in D \times D : x_i \text{ and } x_j \in \text{different clusters}\} \\ &\qquad\qquad\qquad\qquad \text{where } i\ ,j \in (1, 2, ..., n) \end{aligned} \tag{3-4}$$

Consider the following scenario: this research has pairwise restrictions for certain data items and wishes to include this side information in the proposed model. The first question is how this research can make use of this additional information. One method is to use a function that applies the constraints to connect the hidden variables corresponding to data points that must be in the same cluster, and an appropriate function [131] to connect the hidden variables corresponding to cannot-link data items. Another option is to play with the similarity between the data items. This research can maximise similarities between two data objects if they are in the same cluster and minimise them if they are in separate clusters. As a consequence, this research may deduce that clustering efficiency is inversely proportional to data object similarity.

Definition 2: Let us suppose there two data objects such that $(x_i, x_j) \in D$ where $i \in (1, 2, ..., n), j \in (1, 2, ..., n)$, the similarities between these objects S_{ij} or S_{ji} will be adjusted according to Equation (3-5) below.

$$\begin{aligned} (x_i, x_j) \in M \Rightarrow S_{ij} = 1 \,\&\, S_{ji} = 1 \\ \text{and } (x_i, x_j) \in C \Rightarrow S_{ij} = 0 \,\&\, S_{ji} = 0 \end{aligned} \tag{3-5}$$

As a consequence, by increasing the probability of similar constraints being in the same cluster as much as feasible, this adjustment in similarity can increase supervision to enhance clustering performance. AP, like other algorithms like k-means and k-medoids, accepts as input a collection of data object similarities and a preference that might be the median or minimum of the input similarities; unlike other algorithms like k-means and k-medoids, it does not take as input the number of exemplars K. It also produces a random number of exemplars to compute a_{ik} and r_{ik} after exchanging real-valued messages, which may impact its clustering performance. As a result, this research employs the number of exemplars K as an input parameter in AP to solve this problem. The real-valued messages a_{ik} and r_{ik} are calculated after that. This research now incorporates the idea of pairwise constraints, and 15% of the real labels were enforced to know restrictions for each pair of data objects, resulting in similarities being updated. This research already has S_m and p_m in the AP's parameter from Section 3.4. Therefore, S_m is iteratively updated with 1 (if they are in the same cluster) or 0 (if they are not) (if they are in different clusters), for two data objects $(x_i, x_j) \in D$, where $i \in (1, 2, ..., n)$ and $j \in (1, 2, ..., n)$.

After adjusting similarities with constraints, new similarities are again updated with Gram Matrix as shown in Equation (3-6).

$$S_m \Leftarrow P_{ij} \tag{3-6}$$

When this consensus function was used in the suggested cluster ensemble technique CES, it resulted in an increase in clustering accuracy. Finally, utilising the updated similarities, as indicated in Equation (3-6), a good collection of exemplars is obtained. At this stage, this research solves the unsupervised AP problem, which creates a random number of exemplars, as previously stated. By iterating the acquired fine set of exemplars, this research leverages side information such as the number of exemplars K passed as input to AP and constrains it to create exemplars corresponding to K. As a consequence, the accuracy of AP clustering and the execution speed have greatly improved. Thus, this research presents a novel consensus function that is used in the proposed cluster ensemble method CES. Finally, a single robust dataset partition is produced in Γ^* equivalent to a number of clusters in the dataset.

3.6 Performance Evaluation

3.6.1 Experimental Design

The proposed clustering ensemble method CES is compared to several representative clustering ensemble methods on a variety of real-world data sets using representative assessment criteria to assess its performance. The proposed method is tested in ten separate runs. This research chooses a standard evaluation criterion, such as micro-precision, to assess its performance, which compares real labels to predicted labels to assess clustering approaches' accuracy [132]. The researchers have evaluated the consensus cluster's accuracy in terms of true labels using micro-precision [129]. This assessment criterion is also taken into account by the researchers [133]. As a result, this research has used the only considered evaluation criterion to compare the CES approach to other clustering approaches in order to further evaluate its performance. The following are the remaining paragraphs in this section: The datasets used for comparisons will be addressed first. Then this research will go over the assessment criteria and the steps of the experiment in detail.

This research chooses a variety of real-world data sets to implement the experimental study of the proposed CES approach, which are described in Table 3-2. The twelve real-world data sets, which include different samples, features, and classes, were gathered from various sources, including the UCI repository and the Microsoft Research Asia Multimedia (MSRA-MM) image dataset obtained from Microsoft [134]. These data sets are also used in classification due to the availability of class labels, but class labels are not used in clustering for the evolutionary process of clustering [135]. This research uses micro-precision to assess the accuracy of the consensus cluster with respect to the true labels. Matlab R2019a was used to design the experiment. The experiment is divided into two phases: generating ensemble members for these real-world datasets using the cluster-

Table 3-2 The Real-world Data Sets Taken from Different Sources

No.	Dataset	number of objects	Features	Classes
1.	aerosol	905	892	3
2.	alphabet	814	892	3
3.	aquarium	922	892	3
4.	banana	840	892	3
5.	basket	892	892	3
6.	blog	943	892	3
7.	book	896	892	3
8.	heartdisseaseh	294	13	5
9.	glass	214	10	6
10.	heap	155	19	2
11.	wing	856	899	3
12.	water	922	899	3

ing algorithm AP and obtaining consensus function results using the proposed consensus function described in Section 3.5.2. To begin, a similarity matrix is computed using pairwise Euclidean distance and the number of objects n and features f in a dataset, yielding a $n \times n$ similarity matrix S. The preference parameter p is then set to $p = \min(S) / iter \times 0.3$, where $iter$ denotes the iteration number for this step, which is set to 10 to produce m ensemble members. The value $iter \times 0.3$ is used to generate various base partitions and has an impact on clustering performance. The similarity matrix Sm is computed using these acquired base partitions and the preference parameter is set to $p_m = \min(S_m)) / iter \times .09$ after receiving m base partitions after 10 execution of unsupervised AP. These parameters, as well as the number of classes K, are passed as input parameters into the proposed consensus function for further calculations to determine the final partitions of a dataset in K clusters. The introduced consensus function is also executed with $iter = 10$. The primary goal of this experiment is to evaluate the performance of CES and to see how effective the proposed algorithm is when compared to other traditional clustering ensemble methods. CES also outperforms AP in terms of accuracy and execution time due to innovative changes.

3.6.2 Results and Discussions

Table 3-3 shows how the accuracy of CES and other classic cluster ensemble methods is tested on real-world data sets collected from various sources and assessed by micro-precision. The accuracy and execution time of AP and CES are compared in Tables 3-4 and 3-5. The experimental results are divided into two sections: (1) accuracy compar-

Table 3-3 The Comparison of Accuracy Evaluated Using Micro-precision between CES and other Cluster Ensemble Methods

Dataset	**CES**	CSPA	HGPA	MCLA	WSCE	EM	QMI	ECPCS MC	ECPCS HC	CESH	RCSSEDC
aerosol	**54.03**	50.28	50.28	50.28	51.27	39.67	50.61	53.26	51.05	52.26	51.36
alphabet	**51.97**	47.30	47.30	47.30	47.30	37.59	48.40	47.91	48.16	48.36	50.12
aquarium	**70.17**	**70.17**	**70.17**	**70.17**	69.63	36.23	70.07	65.73	69.96	65.16	64.12
banana	**47.98**	42.74	42.74	42.74	44.29	39.40	44.17	43.57	43.21	43.12	44/26
basket	**56.28**	56.05	56.05	56.05	**56.28**	37.89	55.83	52.58	**56.28**	55.12	55.56
blog	**73.59**	73.49	73.49	73.49	72.64	35.42	73.49	66.49	73.49	72.71	71.12
book	**57.70**	57.48	57.48	57.48	57.59	36.27	57.48	56.70	57.37	55.12	54.12
heartdisseaseh	**66.33**	63.95	63.95	63.95	50.00	30.27	54.08	55.10	57.82	55.17	56.12
glass	**65.42**	35.51	35.51	35.51	58.88	45.79	45.79	52.34	52.80	51.12	50.26
heap	**79.35**	54.84	54.84	54.84	77.42	59.35	59.35	59.35	58.71	59.12	59.86
wing	**62.03**	61.92	61.92	61.92	61.68	37.38	6168	57.59	61.68	61.12	62.01
water	**57.16**	56.94	56.94	56.94	56.29	36.66	56.62	55.86	57.05	57.01	56.18
Avg	**61.83**	55.89	55.89	55.89	58.61	39.33	56.46	55.54	57.30	57.10	58.12

Table 3-4 The Comparison of Accuracy between CES and AP

Dataset	AP	CES
aerosol	20.99	54.03
alphabet	15.36	51.97
aquarium	15.08	70.17
banana	18.21	47.98
basket	27.35	56.28
blog	19.72	73.59
book	22.99	57.70
heartdisseaseh	39.80	66.33
glass	53.74	65.42
heap	59.35	79.35
wing	22.90	62.03
water	14.43	57.16

isons between CES and other cluster ensemble techniques on real-world data sets, and (2) accuracy and execution time comparisons between AP and CES.

As a consequence, Table 3-3 shows that, when compared to alternative clustering ensemble methods, CES has obtained promising results in accuracy evaluation on all datasets. Although CSPA, HGPA, MCLA, and CES achieved comparable accuracy of 70.17% in the dataset aquarium, WSCE, ECPCSHC, and CES also achieved comparable accuracy of 56.28% in the dataset basket, CES outperformed state-of-the-art clustering ensemble methods WSCE, ECPCSMC and ECPCSHC by 5.21%, 6.29% and 4.53% on average respectively. Additionally, CES beat all cluster ensemble techniques by an average of 5%. CES outperformed CESH by 7.32% and RCSSEDC by 6%. For the dataset

Table 3-5 The Comparison of Execution Time between CES and AP

Datasets	AP(Avg)	AP(Max)	CES(Avg)	CES(Max)
aerosol	5.6822	6.2422	2.6765	2.7853
alphabet	1.9892	2.8138	1.8246	1.8878
aquarium	2.2208	3.1873	2.2521	2.3075
banana	3.5394	4.6083	1.9807	2.0751
basket	5.0143	5.5720	1.9964	2.0163
blog	1.9467	2.9702	2.2170	2.2603
book	1.8947	2.2981	2.1448	2.2040
heartdisseaseh	0.3843	0.6017	0.4923	0.5328
glass	0.3037	0.5951	0.2521	0.2709
heap	0.1677	0.4678	0.1893	0.2066
wing	1.9915	2.8967	1.9963	2.0228
water	4.7009	5.4571	2.2588	2.3218
Avg	2.4863	3.1425	1.6901	1.7409

alphabet, QMI and, CESH obtained almost equal accuracy. Moreover, CES and RCSSEDC have obtained almost equal accuracy for the dataset alphabet. For the dataset basket, CES, CESHa and RCSSEDC obtained the nearest accuracy. Additionally, for the dataset blog, ECPCS HC, CESH and RCSSEDC acquired almost equal accuracy. For dataset wing, CES, ECPCS HC, CESH and RCSSEDC obtained almost equal accuracy. For dataset water, CES and CESH obtained almost equal accuracy. The employment of the same clustering functionality in both cluster ensemble phases may increase clustering accuracy by increasing the stability of clustering results. Because this research limits AP to creating the actual number of clusters in the suggested consensus function, this research notices a considerable improvement in high-dimensional data sets containing noises, such as aerosol, alphabet, aquarium, banana, basket, blog, book, wing, and water.

When compared to AP, CES has greatly improved in terms of accuracy and execution speed. When compared to AP, Tables 3-4 and 3-5 clearly indicate that CES obtained a considerable improvement in clustering accuracy and execution time. Furthermore, across all twelve datasets, CES has an average accuracy of 61.83%, but AP has an average accuracy of 27.49% with a 55.54%. When it comes to execution time, CES has significantly outperformed AP as shown in Table 3-5. CES has taken 3.4569 seconds, 0.926 seconds, 0.8798 seconds, 2.5332 seconds, 3.5332, 3.5557 seconds, 0.79099 seconds, 0.0941 seconds, 0.0689 seconds, 0.3242 seconds, 0.2612 seconds, 0.8739 seconds, and 3.1353 seconds less than AP in 10 rounds. Finally, on all real-world datasets, CES took 1.4016 seconds less than AP; moreover, the proposed technique required 44.60% less time to execute than AP. On some datasets, AP excels in terms of average time, but only by a small margin. CES, on the other hand, has spent 32.02% less time than AP when the average time consumed across all datasets is considered.

3.7 Summary

In this chapter, this research proposes a new cluster ensemble method (CES) that overcomes the limitations of traditional cluster ensemble methods, which rely on different clustering algorithms to generate base partitions in the ensemble generation step and a single partition in the consensus function, potentially causing compatibility issues in cluster ensemble architecture. Furthermore, dealing with the correctness of the final results was a major concern. On 10 real-world benchmark datasets, this research tested the suggested approach. The suggested clustering ensemble technique outperforms state-of-the-art clustering ensemble methods such as the CSPA, HGPA, MCLA, WSCE, EM, QMI, ECPCS MC, ECPSCS HC, CESH and RCSSEDC algorithms on average, according to the findings. The proposed cluster ensemble approach has numerous advantages. For starters, the framework is more compatible because both phases use the same clustering capabilities, which increases accuracy considerably over existing cluster ensemble methods. Second, it uses a newly proposed consensus function to combine base partitions into a single partition that uses cluster centre information present in a data set to limit AP to produce an actual number of clusters rather than a random number of clusters, resulting in a significant improvement in accuracy and execution time when compared to AP.

Researchers can benefit from the suggested cluster ensemble technique in a number of ways. In knowledge reuse, cluster ensemble is the ideal method for reclustering previously acquired knowledge or hidden patterns from the clustering algorithm. The suggested cluster ensemble technique may be utilised to reuse and recluster clustering algorithm knowledge using the same clustering algorithm. As a consequence, the overheads of incorporating another clustering method for the consensus function are avoided.

This research will improve the accuracy of CES in the future and compare it to sophisticated cluster ensemble techniques and datasets. This research will modify CES till it's on par with other cluster ensemble algorithms in terms of time complexity. Other cluster methods will be investigated, such as AP characteristics such as density peaks [136], which can greatly improve accuracy.

Chapter 4 Workload and Energy State Estimation in Data Centers

4.1 Introduction

Predicting load and energy consumption is also crucial in cloud computing systems. This chapter will focus on their research.

Cloud computing is an Internet-based computing paradigm that allows end-users to access on-demand services by virtualizing hardware resources in data centres [137]. Due to multitenant users, shifting workload conditions, and increasingly complicated infrastructures, resource management in a data centre is often a tough operation. Workloads in modern data centres are highly non-linear. According to an IBM survey, cloud applications' average CPU and memory use ranges from 17.76% to 77.99% [1]. According to a Google study, a cluster's CPU and memory utilisation could not surpass 60%, resulting in significant resource inefficiencies in Cloud data centers [2]. As a result of the workload's non-linear usage patterns, performance is erratic, energy consumption is excessive, and service quality is impaired (QoS). It also raises operating costs and reduces revenue for service providers. Because data centres are costly to develop and operate, resource utilisation must be maximised. While ensuring the application's Quality of Service, an intelligent resource prediction technique can successfully tackle the issue by increasing resource consumption and lowering operational expenses (QoS).

Based on a rich historical workload, a prediction system generates insights into the future demand for a given resource such as CPU, memory, disc, and network. These predictions can be used in data centres to deal with non-linear resource utilisation and energy consumption, as well as resource management decisions like resource provisioning and VM consolidation. For example, a resource provisioning method based on these future insights can handle resource allocation efficiently (i.e., allocating more and fewer resources to VMs based on their needs). Furthermore, rather than the existing reactive approach, decisions can be proactive (e.g., provisioning required resources beforehand to improve QoS and avoid bottlenecks such as resource bootup time). ML approaches can be used to produce workload forecasts, in this case, [21]. Because they are drawn from real features and capable of learning extremely non-linear workload behaviour caused by various factors in data centre environments, machine learning-based forecasts are ideal. Recent resource prediction works focus on CPU, and memory usage and ignore provisioned (requested) resources such as CPU and memory [26,138]. When a new VM is instantiated on a host, these provided resources also make a significant contribution to energy consumption [139]. Furthermore, resource metrics such as disc throughput are ignored, which has a direct impact on a host's energy usage [6]. Network throughput [75] is another important factor to consider when condensing virtual machines to conserve resources. Furthermore, a variety of machine learning methods have been tried to do this task, but no single machine learning algorithm is capable of effectively handling any non-linear workload. As a result, it would be advantageous to use an ensemble learning method involving several machine learning algorithms to predict both provisioned and used non-linear workloads with various metrics such as provisioned CPU, provisioned memory, CPU usage, memory usage, disc throughput, and network throughput.

Energy estimation, like workload forecasting, is critical in data centre resource management. Energy consumption is a major issue in data centres, and providers are working to reduce overall energy use through better resource management. In today's data centres, hosts feature a variety of sensors that monitor energy at the host level. Recent research has focused on calculating energy usage for each virtual machine (VM) using multiple power models [3,4]. However, calculating the energy consumption of VMs at the software level is difficult. For example, memory energy consumption is calculated based on the events raised by each VM on the last level cache of each core (LLC). To calculate energy consumption, this research needs to collect these LLC parameters, which makes computing the energy of each VM difficult [5]. Rather than estimating energy for each VM, this research looked at patterns of comparable VMs in various energy-consumption situations. This is accomplished by looking at the available energy usage features and using clustering analysis to find VMs with similar patterns.

To create the prediction models in this study, this research used real-world workload traces. This research mostly uses Bitbrains data [140], which covers provisioned and consumed resource performance of tens of thousands of VMs housed across many Clouds. Prediction modeling is proposed for two tasks, namely workload prediction and VM energy state estimate. The Resource Management System (RMS) and the Prediction Module are the two components of the system model. In this chapter, this research will show you how to use the Prediction Module. This chapter studies various machine learning algorithms for workload prediction in this regard, and the best models are selected for subsequent RMS activities. To deal with energy state estimation, this research uses an ensemble learning approach and proposes four different clustering methods from semi-supervised affinity propagation based on transfer learning (TSSAP), CLA based on transfer learning (TCLA), K-means based on transfer learning (TKmeans), and P-teda based on transfer learning (TP-teda). According to the tests, the TSSAP beat other approaches by achieving the highest cluster accuracy. In this strategy, this research also employs the Univariate selection method in conjunction with the ChiSquare (χ^2) test to choose the highly significant features associated with energy-consuming states. After that, this research clusters these features in the two-dimensional plane using t-Distributed stochastic neighbour embedding (t-SNE). Eventually, this clustered data is transferred to a different domain for further clustering analysis by using four clustering algorithms such as AP [121], CLA [141], Kmeans [142] and P-teda [143].

In summary, the following are the work's important contributions:

(1) This research proposes a machine learning-based intelligent prediction model for two tasks: workload prediction and energy state estimate.

(2) Using characteristics from a cloud-hosting distributed data centre, this research investigates alternative machine learning algorithms for workload prediction in nonlinear situations. Performance measures such as provisioned CPU, provisioned memory, CPU utilisation, and memory utilisation, as well as disc throughput and network throughput, are among the features.

(3) This research describes an ensemble learning approach to VM-level energy status estimation that incorporates four proposed clustering approaches for identifying comparable groups of VMs based on VM-level variables that may affect energy usage.

(4) GRU has the lowest RMSE values for all features in the workload prediction models.

(5) In comparison to previous clustering models, TSSAP achieves a substantial accuracy of 53.80% in identifying VMs' classes in the energy state estimate models.

The rest of this chapter is organised as follows: Section 4.2 discusses the relevant literature for this project. Section 4.3 explains the motivations for this work as well as the implications of resource management in the cloud. A resource management model is proposed in section 4.4. The used cloud workload traces are described in section 4.5. Section 4.8 is where performance and results are analysed. Finally, Section 4.9 concludes this chapter and provides the future directions.

4.2 Related Work

Machine learning-based prediction has been extended to a wide range of applications. Workload prediction and E-state prediction are two tasks performed by the model. The important work associated with both tasks is mentioned below.

First, this research addresses related work for the first task of the proposed model. [26] used the Random Forest approach to forecast CPU use in disclosed traces of Microsoft Azure VM workloads in the proposed system Resource Central (RC). This solution collects VM features and uses machine learning to understand these behaviours offline before delivering online predictions to various resource managers via a client-side library. Based on the machine learning technique Linear Regression, [144] suggested an evolutionary approach to create an effective prediction model for CPU use for adaptive resource provisioning in the cloud (LR). It can assist e-commerce apps with resource management scheduling and capacity planning that is dynamic and proactive. They worked using data from the TPC-W benchmark. In Google Cloud workload traces, [145] employed a tailored support vector regression method to estimate CPU and memory usage with the goal of proactively resource provisioning to keep resource utilisation and service level agreements (SLAs) at an acceptable level. Based on each host's historical data, [146] utilised Linear Regression to forecast short-term future CPU use. During live VM migration, this procedure was utilised to identify whether a host was overloaded or underloaded based on expected future CPU utilisation. Some VMs relocate to other hosts when a host becomes overcrowded, and when it becomes underloaded, it goes into sleep mode to save energy. VM consolidation is the term for this procedure. The k-nearest neighbour regression method was used by [147] to forecast CPU utilisation in a real-world PlanetLab workload. For this workload, the CPU usage performance of over a thousand VMs was measured at 5-minute intervals. During the VM consolidation process, this forecast was used to reduce energy consumption. [90] employed many criteria such as CPU, memory, and bandwidth use rather than simply CPU utilisation for prediction utilising Multiple Regression using real workload traces in the VM consolidation process to improve energy consumption. From two real workload traces, Google cluster and PlanetLab, [138] employed a regression method to forecast CPU and memory use. During the VM consolidation process, they assess both current and future resource utilisation to determine whether a host is overloaded or not, minimising wasteful VM migration and lowering a host's energy consumption.

This section discusses the related work for the model's second task. Joulemeter is a virtual machine power metering system suggested by [85] that evaluates energy consumption at the VM level using VM resources at runtime. They offered power models that measured energy at the VM level using virtualized platform resources such as CPU, memory, and disc. [148] developed a linear energy model in the GreenClouds project that described the behaviour of a single host and contained numerous components, such as CPU, RAM, and HDD, all of which contribute to a single host's total energy consumption. [149] presented a VM power metering approach based on performance events counter values from resources such as the CPU and RAM because VM energy consumption cannot be detected by any power sensor. The possibility and difficulty of constructing models for black-box online monitoring in VM power metering were investigated in [150], which offered a linear model to track the system's power. [151] presented a linear model for calculating total energy consumption based on static and dynamic resource consumption. [3] proposed an energy-based cost model in the TANGO project, which uses energy consumption as the major parameter in relation to VM's actual resource usage. A tree-regression-based method for calculating the power consumption of VMs on the same host was reported in [4]. [152] displayed a two-dimensional lookup table for each VM. The table includes CPU utilisation, last-level cache (LLC) miss rate, and the power value estimated from CPU utilisation and LLC miss rate.

4.3 Motivation: Intricacies in Cloud Data Center's Resource Management

Resource management is an important part of running a distributed cloud data centre. Because of the presence of multi-tenant users and their diverse workloads, calculating workload levels and energy consumption is difficult. The hosts in cloud data centres have varied amounts of virtual machines at any one time. As a result, the host's workload and energy usage fluctuate. It's critical to study the non-linearity of VM workloads, as well as host characteristics like whether they're over- or under-utilized, and make resource management decisions based on that information (e.g., resource provisioning and VM consolidation). To reduce energy and optimise resource utilisation, data-driven solutions based on machine learning are being investigated. CPU, RAM, disc, and idle power all contribute to the total energy of a host [6]. All of the relevant contributing elements should be considered when estimating the host's energy usage.

When executing CPU-intensive programs, the CPU has a substantial impact on the host's energy usage. The authors of [85] conducted a series of tests and discovered that the CPU consumes 58% host's energy in mixed workloads. Memory accessing and page swapping at the host level account for 20% to 30% of a host's total energy, according to previous research [153] and [85]. Due to the necessity to gather LLC (last-level cache) events raised by each VM on each CPU, measuring the power of each VM at the VM level is difficult. [5] Disks, on the other hand, generate energy through spinning platters and disc head movement. In their research, [85] also presented a linear energy model based on disc read and write throughput. The two basic approaches to energy efficiency are resource provisioning and VM consolidation. By consolidating VMs to fewer hosts via VM migration while maintaining SLAs, VM consolidation attempts to enhance resource utilisation and energy efficiency [154]. These days, intelligent predictive VM consolidation is being implemented, which is thought to be more efficient. On the other hand, network throughput is a crucial parameter that can aid in VM consolidation to save energy indirectly by decreasing resources [75]. According to a study, computing as a service via cloud computing would generate 51,774GB/sec of internet traffic by 2020, which will have an impact on cloud networks [73]. As a result, this element will affect VM migration time and breach SLA [74] in the event of dynamic VM placement. Some researchers, on the other hand, explored projecting CPU utilisation exclusively in the situation of VM consolidation in order to conserve energy [137]. As a result of the foregoing facts, elements such as memory and disc throughput, as well as network throughput, should be considered for VM consolidation prediction in order to conserve energy.

Furthermore, resource provisioning is the distribution of physical resources based on a forecast in order to maximise resource use and energy efficiency. This estimation, which is based on the forecast of future resource behaviour, can help with resource provisioning efficiency. This estimate, which is based on future resource behaviour predictions, can aid in more effective resource provisioning. The bulk of research focused only on the utilisation of physical resources like CPU, memory, storage, and network bandwidth for a prediction-based estimate in resource provisioning [155]. However, while making forecasts, the current study does not take into consideration both supplied and consumed resources. The power models in [139] reveal that provided CPU and memory have a linear connection with energy consumption when a host instantiates a new VM. As a result, resource provisioning based on estimating the combined provisioned and utilised resources

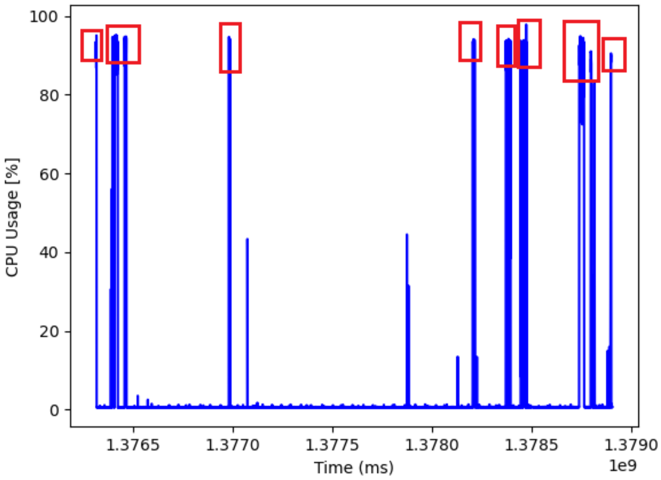


Figure 4-1 VM-1: FastStorage: CPU Usage [%]

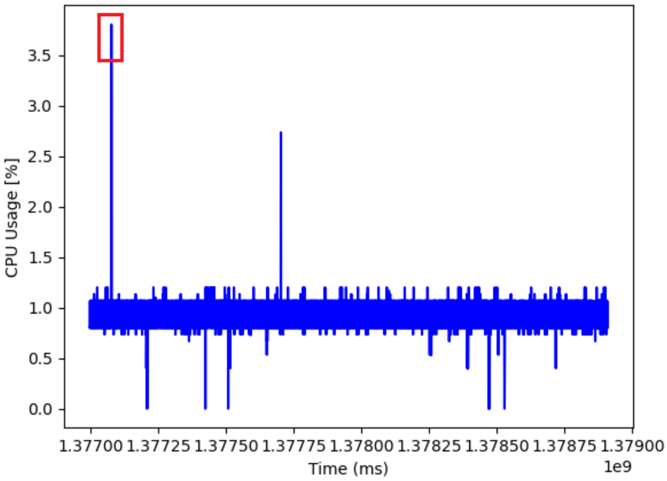


Figure 4-2 VM-2: FastStorage: CPU Usage [%]

can give IaaS service providers a better idea of how much energy they can save.

This research conducted a case study on workload traces to better understand the complexities of a host's power consumption. Figure 4-1 and Figure 4-2 show the CPU utilisation (%) of two separate VMs taken at 5-minute intervals in a fastStorage trace derived from Bitbrain's data collection. It is regarded as over-utilized if a host's peak CPU utilisation exceeds a fixed threshold (e.g., 80%) [156], and it is considered under-utilized if it goes below a chosen threshold (e.g., 30%) [137]. This research looked at 1250 VMs' fastStorage data over a month. For example, in Figures 4-1 and 4-2, peak CPU utilization is between 80% and 100% and 3.5% to 4% for two different VMs, respectively. The CPU capacity for both of these VMs in the same trace is the same. As a result, it is clear that during a given month, CPU utilization for VM-1 reached up to 97.87%, while CPU utilization for VM-2 could not exceed 3.8%, indicating that a host is either over-utilized or under-utilized in this long run. The CPU has a direct linear relationship with the host's total energy usage, according to [6]. It means that if a host's VM's CPU is overworked, it uses a lot of energy, and if it's underworked, its processing capacity is wasted while still using a lot of energy in the form of idle power.

Both scenarios, such as workload forecasting and energy state estimate, are crucial for a data center's energy efficiency and must be handled. As a result, monitoring the energy of each VM in relation to the total energy of a host is a good idea [6]. Each component of a host, such as the CPU, RAM, and disc, contributes to the total energy of the host. Thus, awareness of energy consumption at the VM level can assist energy monitoring of hosts, but measuring the energy consumption of VM devices at the software level is exceedingly challenging. Because LLC (last-level-cache) events triggered by each VM on each core must be collected at the VM level, measuring [5] becomes more challenging. As a result, rather than evaluating the energy of each virtual machine, the patterns of similar virtual machines that are over-utilized or under-utilized will be investigated. To find VMs with similar patterns, clustering analysis might be performed. The focus of the research is on automation. Thus, this research employs a machine learning approach such as clustering to teach the machine these states automatically. Clustering automatically discovers similarities between features and classifies data into similar and different categories. Based on the factors discussed above, this research considers the four different cases of peak CPU utilization as low, medium, high, and critical, respectively, 0%-40%, 40%-70%, 70%-95% and above 95%. The cases low and (high, critical) correspond to under-utilized and over-utilized i.e., low and high, critical energy-consuming states denoted by "E-state" (see Table 4-1). This type of analysis is done by looking at which VMs are correctly separated utilising the four clustering algorithms that have been proposed. The technique isn't confined to these limits; testing with alternative ranges depending on workload data demonstrates this.

Table 4-1 E-states with CPU Utilization

Peak CPU Utilization (%)	E-state
0-40	Low
40-70	Medium
70-95	High
Above 95	Critical

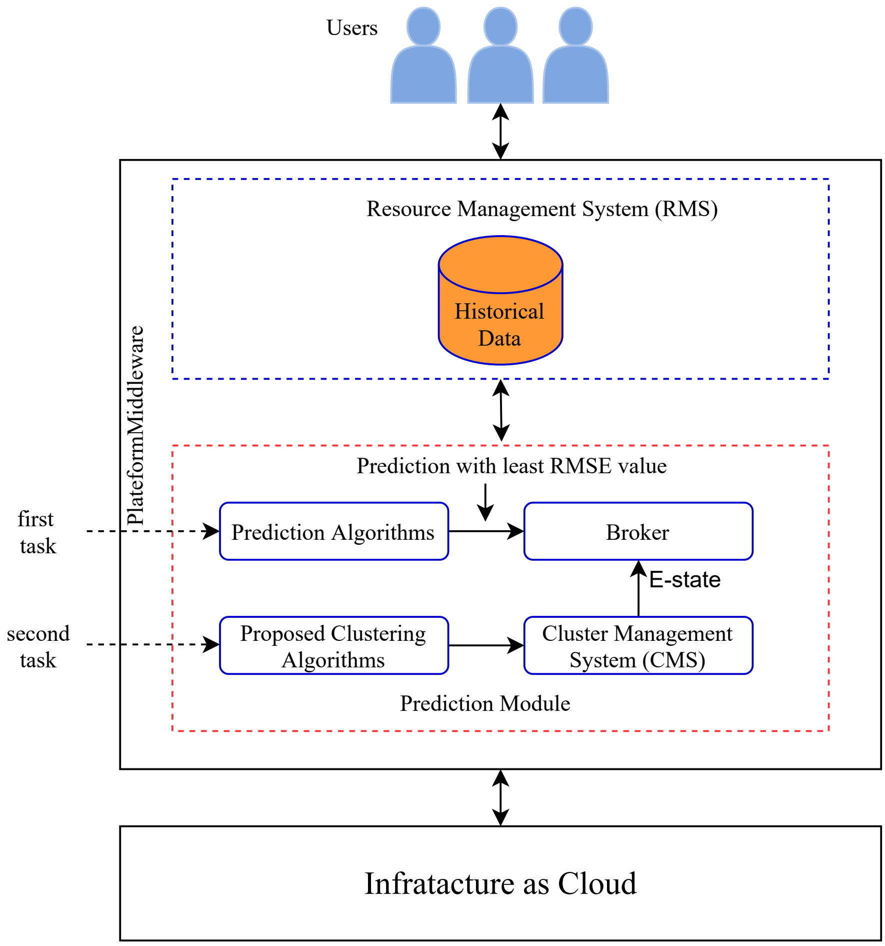


Figure 4-3 Proposed System model

4.4 System Model

A cloud platform is made up of numerous physical machines that provide end-users with on-demand services, and virtualization techniques are used to deploy applications on these actual machines. A diagram of the system model is shown in Figure 4-3. This research picked a data-driven, machine-learning strategy that uses historical application workload to learn from the past and forecast future workload levels and energy states of virtual machines. ML algorithms learn from historical data and assist in data-driven decision-making.

The work is divided into two categories:

(1) Workload prediction, for which this research evaluates multiple machine learning algorithms before selecting the model with the lowest RMSE.

(2) For detecting which virtual machines are in low and high-energy-consuming states, this research offers four distinct clustering algorithms for categorising virtual ma-

chines based on energy-related variables.

This research employs real workload traces to deal with these two activities, which comprise numerous information such as provisioned (CPU, memory) and used resources (CPU, memory, disk, and network throughput). The proposed model's main component is the *Prediction Module*. The Resource Management System (RMS) can make decisions for different resource management tasks in cloud data centres; also, it makes energy management decisions with the help of the *Cluster Management System* from the *Prediction Module*. This research only presents the implementation of its *Prediction Module* in this chapter. The future work will include the performance of *RMS* for resource provisioning, VM consolidation, and other management functions based on the output of the *Prediction Module*. Therefore, the following subsections discuss the critical components of its *Prediction Module*.

4.5 Workload Traces

The data used to train the model in the data centre domain [157] is as good as the data used to train the ML-based prediction system, and training data can include application and physical level variables. Physical resources include CPU, Memory, IO, and other host-level resources, whereas application features include CPU cycles, cache metrics, and other features. This research uses two traces representatives that comprise a business-critical workload that was gathered from a distributed cloud hosting data centre and released by [140]. A service provider that specialises in managed hosting and business computation for corporations obtains a business-critical workload. Table 4-2 shows the details of this business-critical workload, while Table 4-3 shows the definitions of each feature. In these traces, which are taken every five minutes, the vCloud Operation tools record seven performances per VM.

Table 4-2 Distributed Cloud Data Center's Trace for this Work

Trace	VMs	Collection Period	Memory	CPU Cores	Collection Interval
fastStorage	1250	30 days	17729 GB	4057	5 Minute
Rnd	500	90 days	5485 GB	1444	5 Minute

These two traces capture data for 1750 virtual machines (VMs) over 5000 cores and 20 TB of memory over the course of four months, totaling over 5 million CPU hours, making them long-term and large-scale time series. FastStorage, the first trace, has 1250 virtual

Table 4-3 Definition of Features in Workload Trace

Features	Definition (Average)
R_{CPU}	Provisioned CPU capacity [MHZ]
U_{CPU}	CPU usage [MHZ]
R_{memory}	Provisioned memory capacity [KB]
U_{memory}	Memory usage [KB]
D_r^{th}	Disk read throughput [KB/S]
D_w^{th}	Disk write throughput [KB/S]
N_r^{th}	Network received throughput [KB/S]
N_t^{th}	Network transmitted throughput [KB/S]

machines (VMs) connected to SAN storage devices, and its performance was monitored for a month. The Rnd trace, which has 500 virtual machines (VMs) connected to substantially slower Network Attached Storage (NAS), has been examined for three months. By averaging each performance reported for each VM [158], the dataset is smoothed. For one month, this research computes 1250 entries as the average of each feature for each VM in fastStorage trace, and for three months, this research computes 500 entries for Rnd. As a result, this research has 1500 entries in total for Rnd.

4.6 Workload Estimation Using Prediction Algorithms

This research chooses regression-based methods for the workload prediction to estimate a numerical output variable like CPU utilisation, which have also been used in earlier work on non-linear workloads such as (Linear Regression (LR), Ridge Regression (RR)) [155], ARD Regression (ARDR) [159], ElasticNet (EN) [160]. This research also chooses a deep learning method, recurrent neural networks (RNNs) with gated units (GRUs) [92], because it outperforms regular RNNs with other units [161]. This research always takes the average of each VM resource across one month and three months of data for fastStorage and Rnd traces and generates estimates for these VMs based on that. In Section 4.2, the justification for choosing the average value was explored. Furthermore, as shown in Figure 4-1, the peak CPU utilisation in red rectangular boxes rapidly decreases after a short time interval, making it feasible and efficient to train the ML model using the average of each VM and forecast the average prediction value of each VM provisioned and used resources based on this learning. This research utilises the sci-kit learn [162] package to implement all of the ML methods, and the Keras [163] package to create the deep learning method GRU. The arguments for each of the ML methods are set to their default values in this implementation. For RR, the parameters are $alpha = 0.2$ and normalise = true. All ML-regression algorithms are trained using numerous features to predict the target variable. For example, if the objective variable is set to average CPU utilisation, the remaining features for training the ML regression algorithms are chosen from the traces. Furthermore, this research evaluates the goodness of fit of various techniques using the Root Mean Square Error (RMSE) metric, which is a typical evaluation metric in regression-based situations [164].

As a result, the model will be more accurate if the RMSE values are lower. In addition, the model is examined to be more precise if its RMSE value is adjacent to 0.

The performance of several ML-regression approaches and deep learning methods is shown in Tables 4-6 and 4-7. The RMSE value for various features of the selected traces is represented in these results. The deep learning method GRU has very low RMSE values, meaning that residuals or prediction errors are reduced and predictions are more accurate, as shown in these tables. Furthermore, similar results have been obtained using several regression approaches. Because the GRU results are the most promising and have the lowest RMSE. As a result, this research will focus more on this algorithm in Section 4.6.1 to investigate it more and explain it.

4.6.1 Learning with Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a gated recurrent unit introduced by [92] (GRU). This research trains the model with 80% of the goal variable, such as CPU utilisation, and the trained model predicts with 20% of the target variable in this manner. The GRU architecture is defined by the following equations:

$$\begin{aligned}
u_t &= \sigma(W_u x_t + U_u h_{t-1} + b_u), \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r), \\
\tilde{h}_t &= \tanh(W_h x_t + U_h h_{t-1} r_t + b_h), \\
h_t &= u_t h_{t-1} + (1 - u_t)\tilde{h}_t
\end{aligned} \tag{4-1}$$

The vectors u_t and r_t, for example in Equation (4-1), correspond to the update and reset gates, respectively. The state of the vector at time t is represented by h_t. The activation function of both gates is *sigmoid* function which is represented by σ. This function is in charge of limiting the range of values for u_t and r_t from 0 to 1. Furthermore, a *hyperbolic tangent* tangent function evaluates the candidate state $\tilde{h}_t$. The GRU network is fed with input x_t (in the case, a vector of CPU usage (U_{CPU}) values) and the feed-forward connections W_u, W_r, and W_h, as well as the recurrent weights U_u, U_r, and U_h. Before nonlinearities in the network, the trainable bias vectors b_u, b_r, and b_h are included.

In addition, the Pandas and Numpy [165] libraries are used to load workload traces as a pandas data frame and convert integer numbers to floating-point values that are better suited for neural network operations. The MinMaxScaler is used to rescale the data from 0 to 1. The original dataset is then used to convert the dataset to a new shape, with the look-back parameter set to 1, indicating the number of prior time steps to be utilised as input variables to predict the following period [155]. Furthermore, this model has one input layer, one hidden layer, and one output layer, with one input, five neurons, and one output forecast as optimised results. In addition, the model can be enhanced by adding more neurons to produce better results. Finally, the network is trained for epochs=100 and batch size=64 using mean square error as a loss function and Adam as an optimiser [166]. As mentioned in Section 4.8.2, this research has optimised performance for certain hyperparameters. In the training phase, this research also uses the validation data parameter, which is the date on which the loss and other model metrics are validated at the end of each epoch, but the network is not trained on it. The RMSE is used to evaluate the model's performance on test data after it has been fitted.

4.7 VM Energy State Estimation Using Clustering Algorithms

To build similar groups of VMs, this research proposes four clustering strategies. This research forecasts a VM's energy state, such as low, high, or critical. These models would aid in resource management decisions that would improve resource efficiency.

These techniques are based on the four clustering algorithms listed below:

(1) AP [121]: This is an exemplar-based algorithm that is used to propose TSSAP.

(2) CLA [141]: Every data point in this algorithm is given a mass and is linked to a special force called the local resultant force (LRF) generated by its neighbours.

(3) Kmeans [142]: This algorithm aims to group n data points into K classes, with each data point being a neighbour of the cluster centre closest to it.

(4) P-teda [143]: This algorithm is designed to handle high-frequency data. This method incorporates the TEDA theory concept and inherits all of its benefits.

This research employs the [167] transfer learning methodology in all of these methods to learn resilient clusters for the target domain utilising knowledge from the source domain.

As a result, as explained in Section 4.7.1.1 for TSSAP, this research supplies identical source domain information to all methods. Apart from transfer learning, this research limits the affinity propagation (AP) approach to produce a number of clusters equal to the actual number of clusters and add semi-supervised learning utilising pairwise [98] and non-matrix factorization [168]. The proposed approaches are known as semi-supervised affinity propagation based on transfer learning (TSSAP), CLA based on transfer learning (TCLA), K-means based on transfer learning (TKmeans), and P-teda based on transfer learning (TP-teda). This research will largely focus on TSSAP in the following subsections to discuss it in detail because it has achieved encouraging clustering results.

TSSAP is a semi-supervised clustering algorithm that uses a tiny amount of supervised data in the form of partial labels to supervise unsupervised data and generate more accurate related clusters. The information regarding similar clusters is supplied to the *CMS* since the results are promising. Before sending the clusters of VMs to the *Broker*, which then sends these three clusters to the *RMS* for further analysis, it divides them into Low, High, and Critical energy-consuming states. The proposed workflow of TSSA is depicted in Figure 4-4. The operation of this approach is discussed in full below.

4.7.1 Transfer Learning-based Semi-supervised AP

Transfer learning Transfer learning [169] is a sort of learning that focuses on learning robust classifiers for a target domain utilising knowledge from a source domain. This research employs the univariate feature selection technique, which involves utilising univariate statistical tests such as the chi-square test [94] to choose the best features. It's used to test if the class label is independent of a specific feature in the context of feature selection on a labeled dataset.

After getting the best characteristics that are most connected to class labels, this research utilises t-Distributed stochastic neighbour embedding (t-SNE) [169] to reduce high-dimensional features to two-dimensional features via a matrix of pair-wise similarities. It efficiently separates data into clusters, which this research then clusters further using a modified AP approach that enhances clustering accuracy.

Modified AP In AP, the input parameters are similarities $S(i,j)$ between data points and preference p, which is the median or minimum of calculated similarities. As a result, two-dimensional features derived from the t-SNE operation were used to calculate input similarities using Euclidean distances like $S(i,j) = -||xi - xj||$ and preference, $p = min(S)$.

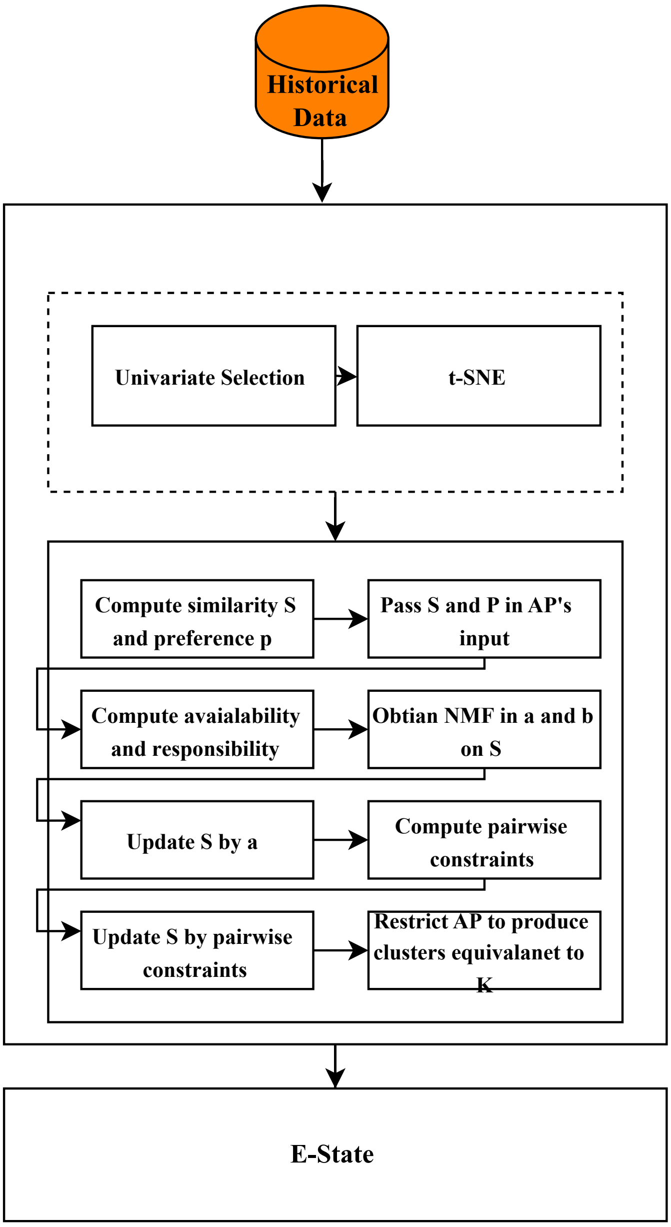

Historical Data
Univariate Selection
t-SNE
Compute similarity S and preference p
Pass S and P in AP's input
Compute avaialability and responsibility
Obtian NMF in a and b on S
Update S by a
Compute pairwise constraints
Update S by pairwise constraints
Restrict AP to produce clusters equivalanet to K
E-State

Figure 4-4 Workflow of TSSAP

The number of classes in data is not used as a parameter in AP, which results in a random number of exemplars. It could have an impact on AP's clustering performance. Therefore, in AP's input, this research passes this supervised information, such as the number of classes K, along with $S(i,j)$ and p. The real-valued messages $a(i,k)$ and $r(i,k)$ are then computed by AP. This research uses non-matrix factorization (NMF) [168] to update similarities at this point.

Definition 1: Assume this research has a X matrix with m features and n samples.

NMF decomposes matrix X into two matrix $A(m \times q)$ and $B(q \times n)$ such as that,

$$X \approx AB \tag{4-2}$$

In detail, it can be expressed as,

$$X = AB + e \tag{4-3}$$

The matrix norm of $X - AB$ is computed as e. X is made up of similarities between data points calculated using the $S(i,j)$ Euclidean distance. As a result, the NMF method is applied to $S(i,j)$, which is decomposed into A and B. These elements are upgraded repeatedly in order to reduce the estimation error $X \approx AB$. Various operations, such as Euclidean distance, can be used to calculate the distance between AB and X,

$$d_{euc}(A,B) = \frac{1}{2}||X - AB||^2 \tag{4-4}$$

and similarities $S(i,j)$ is updated with matrix A, such that

$$S(i,j) = A(i,j) \tag{4-5}$$

This research employs semi-supervised learning to provide extra supervision to the updated similarities from NMF. Semi-supervised learning combines labeled and unlabelled components to bridge the gap between unsupervised and supervised learning. The learning rate increases when unlabeled input is paired with supervised data. Semi-supervised clustering has lately acquired popularity, in which limited supervision is supplied by leveraging various side information strategies to boost precision in unlabeled data partitions, such as instance-level constraints, partial labels, and relative distance comparisons. This method makes use of the instance-level constraints introduced by [98] to improve the accuracy of the results. These constraints indicate that two data points must link $must - link$ if they are in the same cluster, but cannot link $cannot - link$ if they are in different clusters.

Definition 2: Assume that a data set $Y = \{y_1, y_2, ...y_n\}$ exists, and that the cluster information is represented by a set $\gamma \subset Y \times Y$, where $\gamma = m_l \cup c_l$, and that for $(i,j) \in (1, 2, ..., n)$,

$$\begin{aligned} m_l &= \{(y_i, y_j) \in Y \times Y : y_i \text{ and } y_j \in \text{same cluster}\} \\ c_l &= \{(y_i, y_j) \in Y \times Y : y_i \text{ and } y_j \in \text{different clusters}\} \end{aligned} \tag{4-6}$$

The constraints for each pair of data points were determined using 30% of the actual labels, and the similarities obtained from NMF were updated again using these constraints. Similarities $S(i,j)$ for two data points (y_i, y_j) are updated with 1 or 0 if they are in the same cluster or not, respectively, for $(i,j) \in (1, ..., n)$, such that,

$$(y_i, y_j) \in m_l \Rightarrow S(i,j) = 1 \ \& \ (y_i, y_j) \notin m_l \Rightarrow S(i,j) = 1 \tag{4-7}$$

Finally, a fine set of prospective exemplars is created using this similarity matrix, as shown in Equation (4-7). By introducing a modest quantity of supervised data K into AP's input, this research also confined AP to producing a random number of exemplars. As a result, the accuracy of clustering improved. Algorithm 1 shows the TSSAP pseudo code:

Algorithm 4-1 Pseudo code of the proposed clustering approach TSSAP

```
Input: Features, labels, No. of clusters K
Output: E-state
$R = [\,], temp = [\,], X = [\,], f_1 = [\,], f_2 = [\,];$
/* Transfer Learning                                          */
$f_1 \leftarrow \chi^2(Feature, labels)$
Select highest 4 in $\chi^2$ score features from $f_1$
$x \leftarrow tsne(f_2, euclidean)$
/* Semi-supervised AP                                         */
$S(i,j) \leftarrow Euclidean(x_i, x_j)$
$p \leftarrow \min(S)$
Pass $S(i,j)$ and $p$ in AP's input
/* Execute AP, $iter = 10$ times                              */
Compute $A(i,k)$ and $R(i,k)$
$(a,b) \leftarrow nnmf(S(i,j))$
$S(i,j) \leftarrow a(i,j)$
$s \leftarrow .3(lables)$
for i = 1 to length(s) do
    for j = i + 1 to length(s) do
        if $(x_i, x_j) \in c_l$ then
            $S(i,j) \leftarrow 0$
        else
            $S(i,j) \leftarrow 1$
        /* where $C$ denotes cannot-link constraints          */
return idx
E-state ← idx
```

4.7.1.1 Learning with Semi-supervised Affinity Propagation based on Transfer Learning (TSSAP)

For the proposed method's learning, this research chooses one month of fastStorage trace data, which comprises the performance of 1250 VMs running in a dispersed data centre. The data from these VMs was analysed based on peak CPU use and categorised into different ranges, as mentioned in Section 4.2. Because peak CPU utilisation significantly influences under- and over-utilization, i.e., low and high, essential energy consumption in the case of VMs allocated to hosts, these ranges correlate to distinct energy-consuming states. As a result, the suggested method's input is based on the characteristics of the 1250 VMs. As a result, analysing similar patterns based on these performances and seeing the findings with these ranges would be realistic and reasonable. The proposed approach's initial step is to use the univariate selection method to analyse the attributes. This method's SelectKBest uses a χ^2 test with $k = 4$ to select the best 4 features with a χ^2 score. This research uses t-Distributed Stochastic Neighbor Embedding (t-SNE) to cluster the selected features such as R_{CPU}, U_{CPU}, R_{memory}, and U_{memory} in a 2-dimensional plane after capturing the best features related to defined E-state. The data is subsequently sent to the input of the improved AP model, which clusters the data more precisely into different energy-consuming states.

$$Predicted\ Class \xleftrightarrow[\text{E-state}]{\text{1250 VMs}} Cluster(R_{CPU}, U_{CPU}, R_{memory}, U_{memory}) \tag{4-8}$$

In detail, this information is used to compute a similarity matrix using pairwise Euclidean distance with n number of data points, resulting in a $n \times n$ similarity matrix S. Following that, the preference parameter p is set to $p = min(S) / iter \times 0.3$, with $iter$ denoting the iteration number, which is $iter = 10$. To improve accuracy, the preference parameter p can be tweaked with different input values. In this case, $p = \min(S) / iter \times 0.3$ provides optimal performance. The parameters S and p, as well as the number of classes (K) and labels *labels*, are passed into AP's input for further calculations in order to evaluate the final clusters. TSSAP also generates anticipated labels for VM partitions. i.e., the expected energy consumption states of each of the 1250 virtual machines. Standard evaluation criteria such as micro-precision are used to compare these anticipated labels to actual labels. This measure was chosen because it assesses the accuracy of clustering techniques by comparing real labels to anticipated labels [132]. As a result, because this research creates distinct ranges based on peak CPU use, it also applies to the scenario.

4.8 Performance Evaluation

In this section, this research assesses the performance of the proposed model in conjunction with the prediction module, as well as compares the results.

4.8.1 Experimental Setup

The tests are run on a system with a 1.90 GHz Intel(R) Core(TM) i3-4030U processor and 4 GB of main memory. The Prediction Module of the proposed model does two tasks: (1) implements various prediction algorithms using PyCharm Community 2020.2, and (2) predicts E-state using the provided clustering approaches using PyCharm Community 2020.2 and Matlab R 2019a.

To execute all ML-based regression algorithms, this research uses the sci-kit learn package [162]. GRU is also implemented using the Keras [163] deep learning library. This research uses [140], a real-world dataset from Bitbrain, which this research discusses in Section 4.4. This dataset was chosen because it includes both supplied and used resources that fulfill the criteria for the first targeted task, and it also includes real-world cloud infrastructure utilisation patterns. Because the model requires the most promising forecasts from all of the prediction algorithms applied, all of the prediction algorithms are compared using RMSE to see which one has the fewest residual errors when compared to actual data. This research utilises Pycharm 2020.2 to extract knowledge from one-month data from the fastStorage trace using Pycharm's chi-square test and Matlab's t-sne for the second challenge. The data is then sent into Matlab's programming tool, where it is clustered using four distinct algorithms: modified AP, CLA, Kmeans, and P-teda. All of the proposed clustering approaches are examined with micro-precision in order to discover the most promising outcomes. In Section 4.8.2, this research primarily focuses on TSSAP results because, in this instance, it has produced the most promising clustering results in contrast to other offered approaches.

4.8.2 Analysis of Results

In the suggested model, the prediction Module is employed to handle two tasks: workload prediction and E-state prediction. This research will start with the workload prediction results. The workload prediction module can experiment with various machine learning approaches to generate workload forecasts for various workload kinds. This research investigates different machine learning (ML) methods and a deep learning method, such as LR, RR, ARDR, EN, and GRU, for predictions on different types of workloads, including provisioned and utilized. The lower the RMSE, the more accurate the forecast. The parameters for each ML method and GRU are detailed in Sections 4.6, while the outcomes of predicted cases in terms of performance measure are shown in Tables 4-4 and 4-5 for fastStorage and Rnd traces, respectively. These tables show the RMSE values achieved using the various methods. Tables 4-4 and 4-5 show that none of the ML algorithms, LR, RR, ARDR, and EN, fit the dataset well and produce consistent predictions. By observing the results, it is concluded that when the workload feature has a small digit value, the RMSE value is very low.

Table 4-4 FastStorage: RMSE Values of Different Algorithms in Predicting Different Features

Features	GRU	LR	RR	ARDR	EN
R_{CPU}	3.46	417.66	1899.17	418.29	605.21
U_{CPU}	0.44	911.95	1002.97	1886.30	923.67
R_{memory}	9.29	9448342.05	7930792.37	8929672.01	9089470.08
U_{memory}	372.42	365978.61	384364.48	366817.80	366030.14
D_r^{th}	0.37	3176.39	3192.39	3245.52	3183.62
D_w^{th}	0.06	325.19	323.77	338.36	324.26
N_r^{th}	0.33	53.17	54.18	68.22	53.37
N_t^{th}	0.23	93.83	93.22	101.90	94.21

Table 4-5 Rnd: RMSE Values of Different Algorithms in Predicting Different Features

Features	GRU	LR	RR	ARDR	EN
R_{CPU}	5.22	449.59	1541.14	442.51	449.59
U_{CPU}	1.79	754.13	772.98	1134.96	766.42
R_{memory}	9.85	24629258.91	25318293.39	24638065.90	24616865.76
U_{memory}	1262.93	441097.03	440142.87	433219.81	437761.33
D_r^{th}	1.59	746.05	722.07	744.29	750.16
D_w^{th}	0.76	407.76	396.37	435.44	403.39
N_r^{th}	0.9	664.48	670.55	666.83	664.10
N_t^{th}	0.61	603.97	604.41	605.55	609.79

The deep learning method GRU, on the other hand, accumulates extremely few RMSE values for all features, compared to other ML algorithms that have very large RMSE values suggesting poor performance. When it comes to modeling workload time series, GRU outperforms ML regression methods. The availability of two vectors that de-

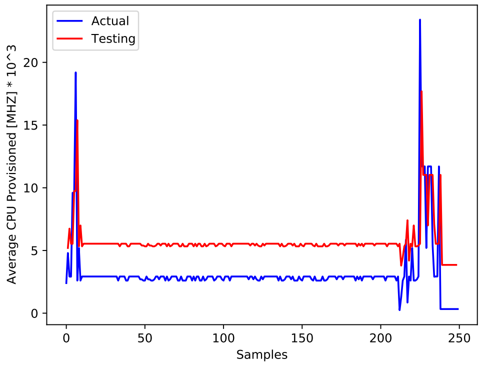


Figure 4-5 Fast Storage: R_{CPU}

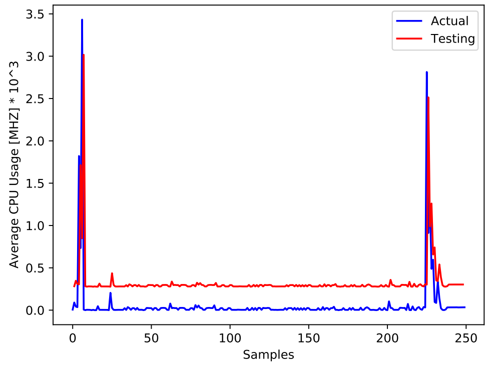


Figure 4-6 Fast Storage: U_{CPU}

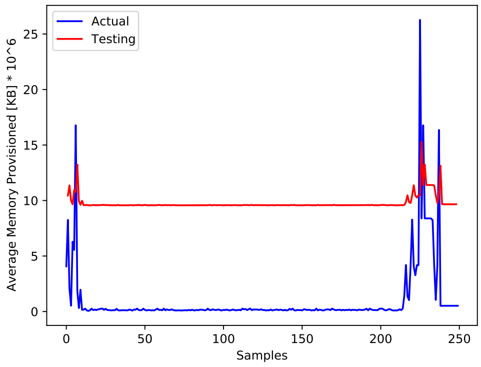


Figure 4-7 Fast Storage: R_{memory}

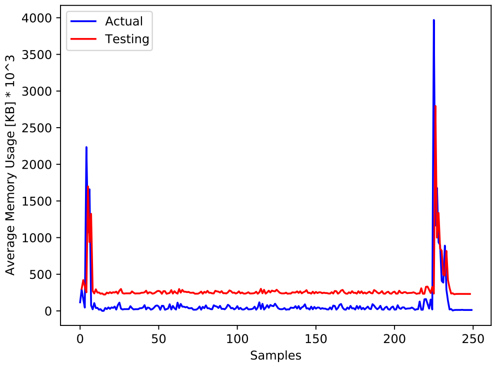


Figure 4-8 Fast Storage: U_{memory}

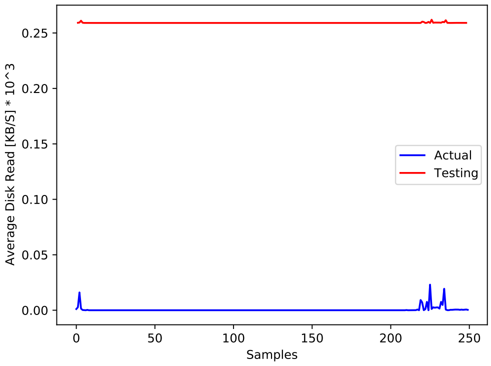


Figure 4-9 Fast Storage: D_r^{th}

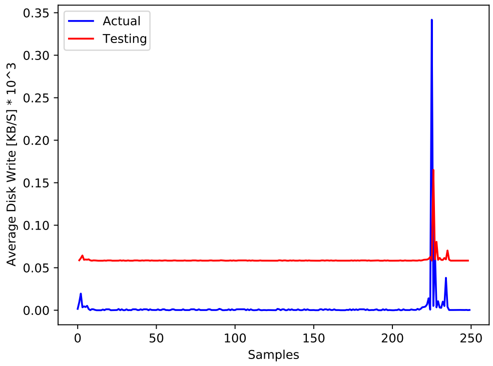


Figure 4-10 Fast Storage: D_w^{th}

cide what information should be transmitted to the output is one of GRU's most essential aspects. They are remarkable in that they can be taught to remember information from the past without it being washed away over time or information unrelated to the forecast being removed.

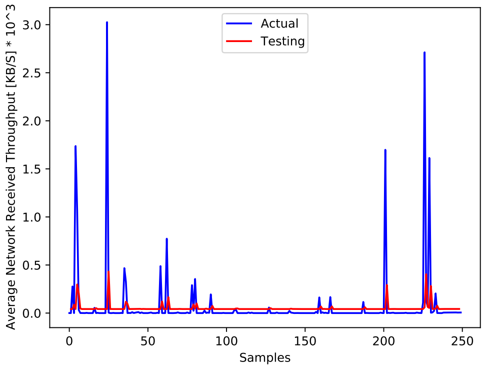


Figure 4-11 Fast Storage: N_r^{th}

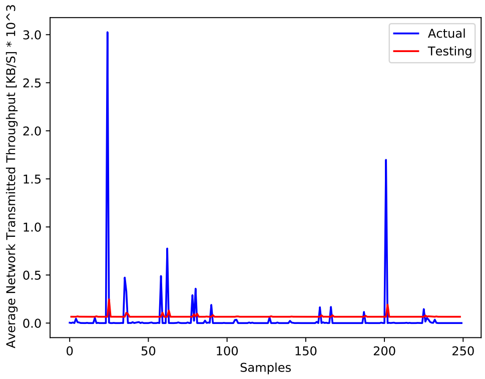


Figure 4-12 Fast Storage: N_w^{th}

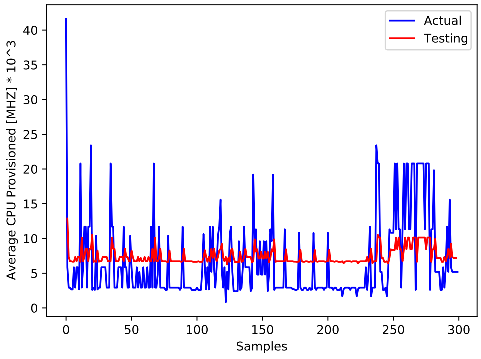


Figure 4-13 Rnd Storage: R_{CPU}

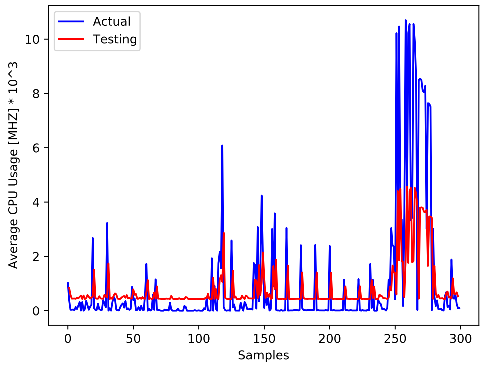


Figure 4-14 Rnd Storage: U_{CPU}

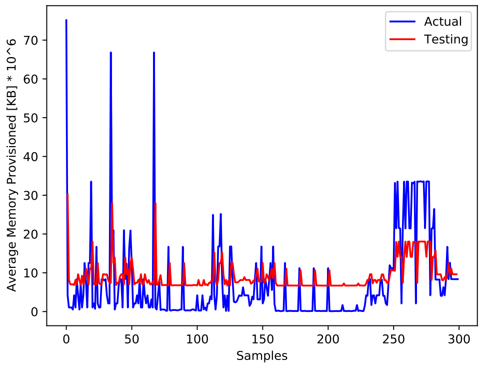


Figure 4-15 Rnd Storage: R_{memory}

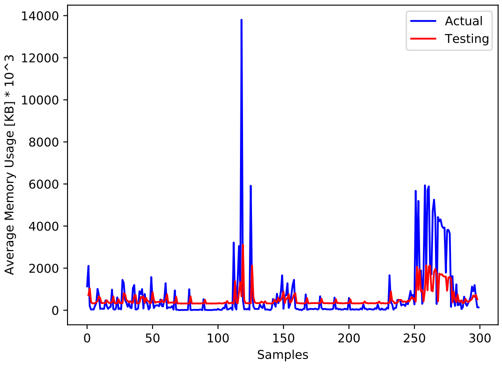


Figure 4-16 Rnd Storage: U_{memory}

GRUs perform better as a result of their ability to retain track of context-specific temporal connections between task features for longer periods of time while making future predictions. The results also suggest that GRU delivers greater accuracy when the dataset is large. The model can extract more patterns and modify the layer weights more precisely

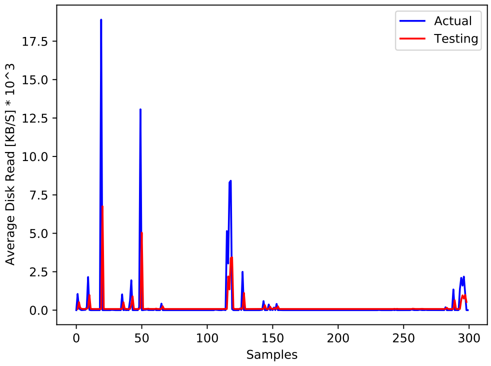


Figure 4-17 Rnd Storage: D_r^{th}

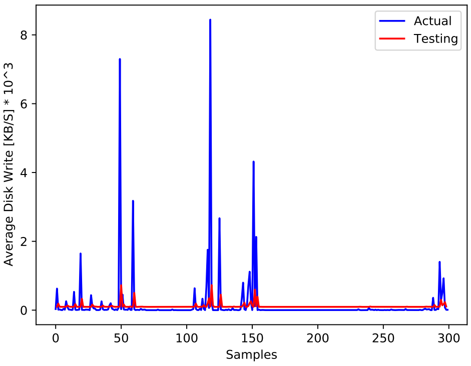


Figure 4-18 Rnd Storage: D_w^{th}

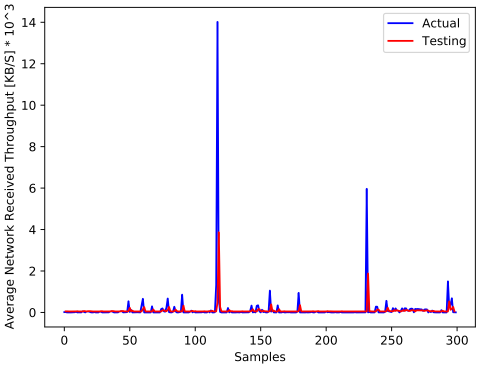


Figure 4-19 Rnd Storage: N_r^{th}

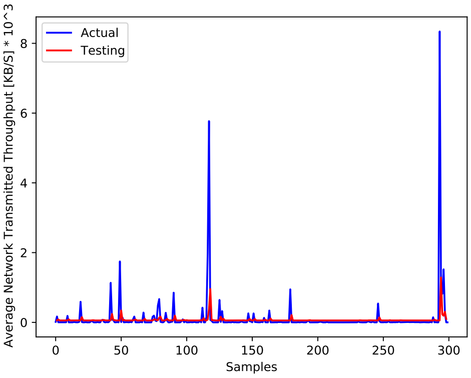


Figure 4-20 Rnd Storage: N_w^{th}

Table 4-6 GRU Model Training at Different Hyper Parameters

epochs	batch_size	Train score	Test Score
10	32	2.09	0.81
40	32	1.77	0.39
100	32	1.76	0.46
10	64	2.66	1.26
40	64	1.79	0.52
100	64	1.76	0.44

as more data is collected, whereas with classic regression algorithms, the smaller the data, the greater the accuracy. The accuracy of standard regression methods is reduced by a huge dataset, as seen in the tables.

Furthermore, this research can expect fewer residual errors in prediction offered by GRU with a larger training dataset and more hyperparameter adjustment if this research uses a better infrastructure, such as a GPU cluster. This research does not want to use hyperparameter optimization to acquire the optimal model; instead, this research would like to supply a generic model that can be used with other models. To train the GRU model, this research employed different hyperparameter values, such as epochs and batch size. Epochs are the number of iterations across which the input data is delivered. The batch size parameter specifies the number of samples to be updated per gradient update; it is set to 32 by default. Table 4-6 shows how the model is trained using various hyperparameters.

It is considered that if the model is well-trained on the data, it will perform better. The model trained at epochs=100 has the least trained RMSE score in both batch sizes, 32 and 64, as shown in Table 4-6. The amount of samples per gradient update is either 32 or 64. The model will definitely train faster with a batch size of 64 than with a batch size of 32. In order to train the model with the optimum performance, this research uses the epochs=100 and batch size=64 tuning case for all features. Because the GRU results have been proven to be promising, this research chooses to represent them aesthetically. The visual representations of GRU findings for fastStorage and Rnd are shown in Figures from 4-5 to 4-20. The total samples for fastStorage and Rnd are 1250 and 1500, respectively. Eighty percent of the data is used to train the model, and twenty percent is used to test it. As a result, the findings of 250 and 300 samples for both traces can be plainly seen. The actual (blue) and anticipated (red) statistics are clearly shown in both pictures. During epochs=100, the training and validation loss graphs for each feature of both traces are displayed in Figures 4-21 to 4-36. The model performs similarly on both training and validation data, as seen by the loss graphs. The learning should be terminated if these two loss plots begin to move consistently. All subfigures in Figures from 4-21 to 4-36 display consistent movement at epoch =100, demonstrating that the model has learned extremely well. By fine-tuning hyperparameters, the model can be trained more effectively.

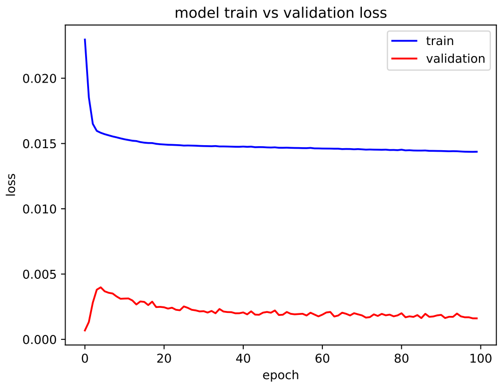


Figure 4-21 Fast Storage: R_{CPU}

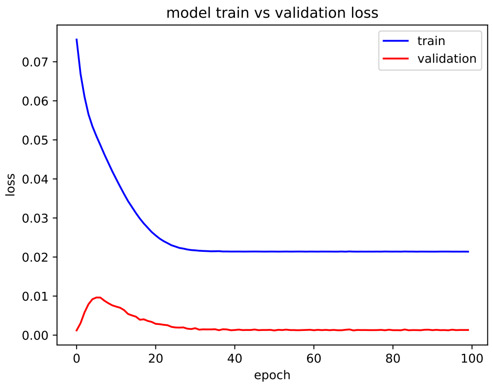


Figure 4-22 Fast Storage: U_{CPU}

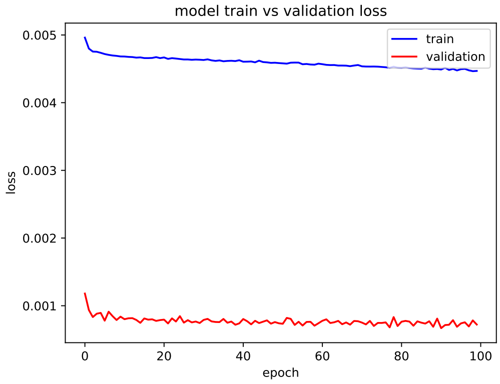


Figure 4-23 Fast Storage: R_{memory}

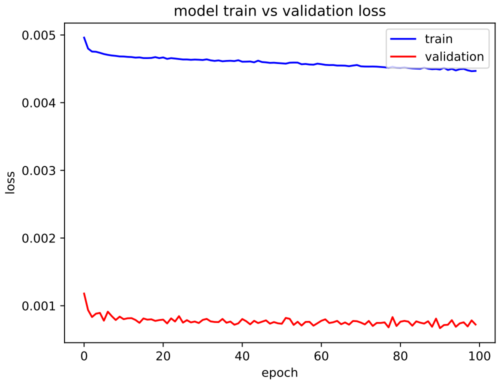


Figure 4-24 Fast Storage: U_{memory}

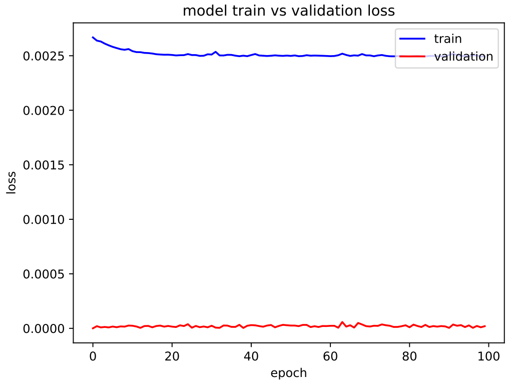


Figure 4-25 Fast Storage: D_r^{th}

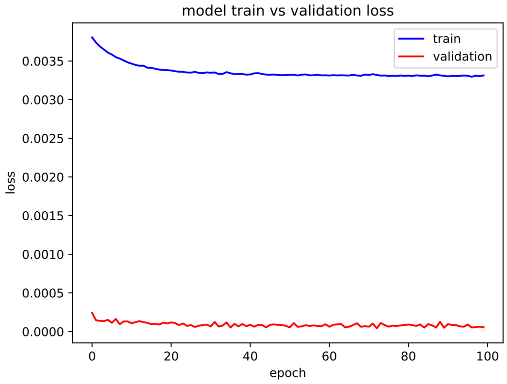


Figure 4-26 Fast Storage: D_w^{th}

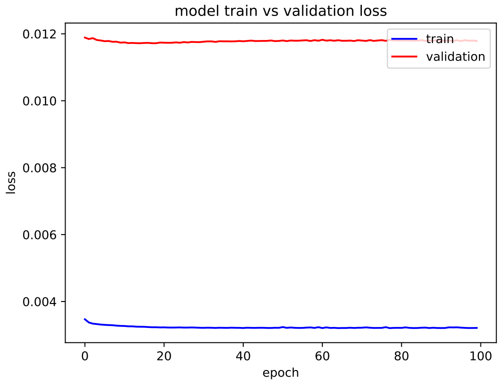


Figure 4-27 Fast Storage: N_r^{th}

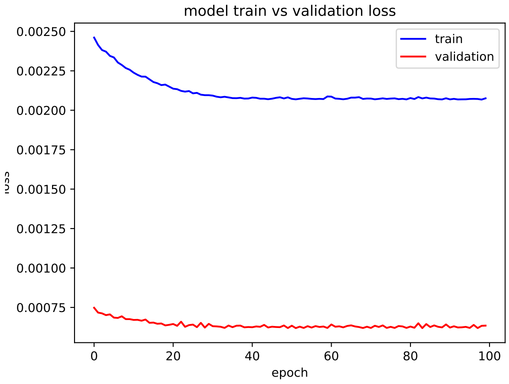


Figure 4-28 Fast Storage: N_w^{th}

Now this research will talk about the results of the E-state estimation. This research proposes four different clustering algorithms to cluster similar types of VMs based on their energy-consuming state, i.e. E-state, and compare the forecasting results obtained by the proposed methods. This research selects one-month data from fast-

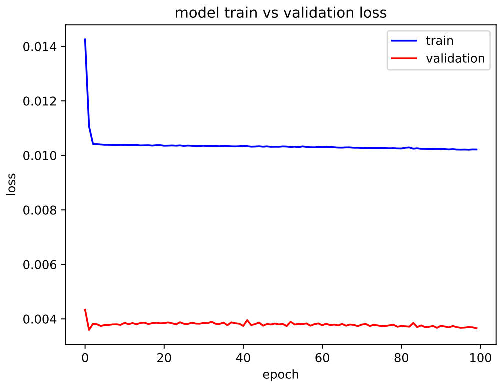


Figure 4-29 Rnd Storage: R_{CPU}

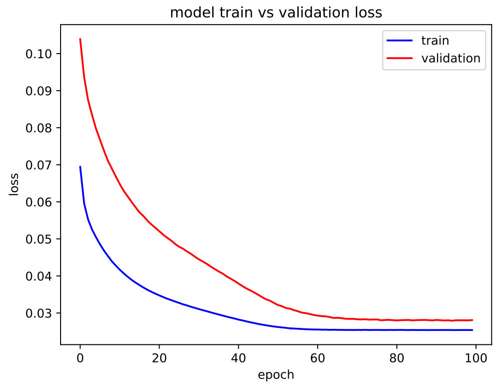


Figure 4-30 Rnd Storage: U_{CPU}

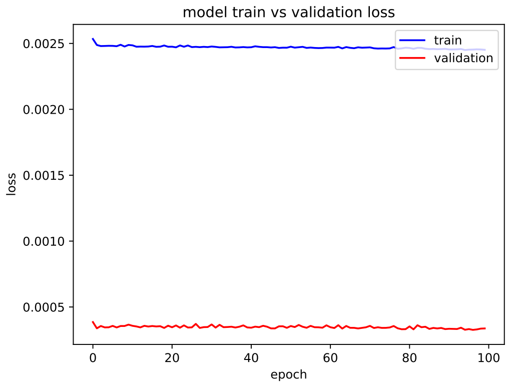


Figure 4-31 Rnd Storage: R_{memory}

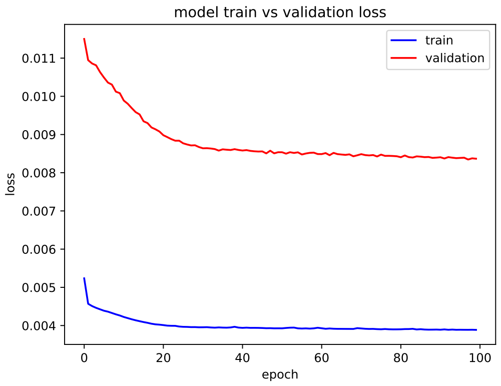


Figure 4-32 Rnd Storage: U_{memory}

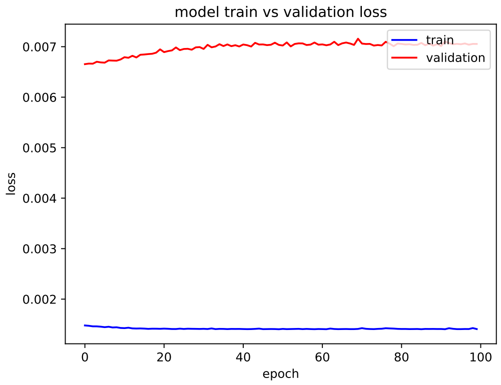


Figure 4-33 Rnd Storage: D_r^{th}

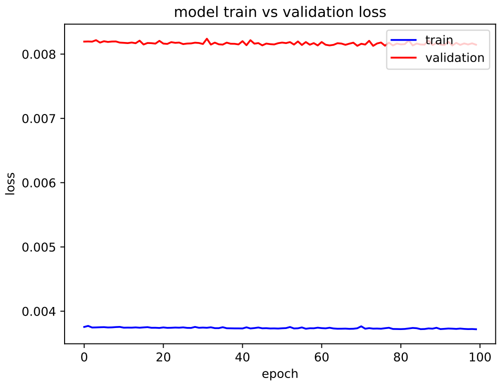


Figure 4-34 Rnd Storage: D_w^{th}

Storage traces, which includes 1250 VMs with various features such as R_{CPU}, R_{memory}, U_{CPU},U_{CPU},D_r^{th},D_w^{th},N_r^{th},N_t^{th}. As discussed in Section 4.2, this research also defines different energy-consuming states. This research uses the univariate selection method on these features, along with the χ^2 test, to find the best four features to use on these range labels,

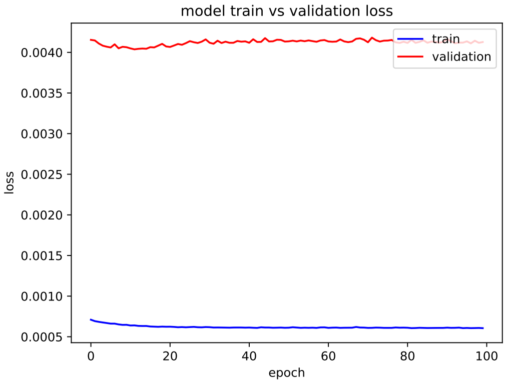


Figure 4-35 Rnd Storage: N_r^{th}

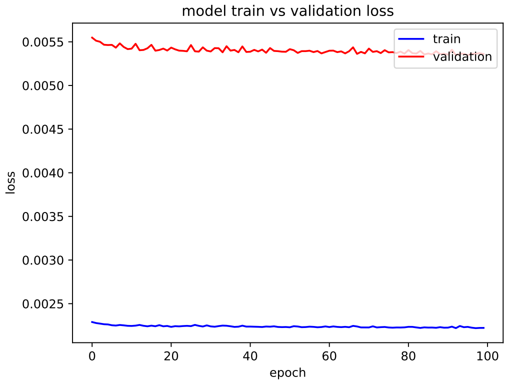


Figure 4-36 Rnd Storage: N_w^{th}

and to ensure that they are independent of other features. During this test, the variables R_{CPU}, U_{CPU}, R_{memory} and U_{memory} appear with the highest χ^2 score of $3.234e^{+5}$, $3.644e^{+5}$, $1.374e^{+9}$, and $1.032e^{+8}$, respectively. As can be seen, provided resources like *RCPU* and *Rmemory* have an impact on a host's energy usage. As a result, the suggested clustering algorithms use these selected features to locate similar groups of VMs based on these features, which have been shown to be accurate and precise. On the specified dataset, the proposed clustering algorithms, TCLA, TKmeans, and TP-teda, achieved 12.48%, 50.88%, 51.20%, and 66.48% accuracy, respectively, as shown in Figure 4-37. Furthermore, TSSAP has the highest clustering accuracy of all of them, at 66.48%. The AP clustering technique, which has an accuracy of 8.32%, is used to propose TSSAP. As a result, TSSAP outperforms AP by 87.48% and outperforms the average of other proposed approaches by 53.80% since it incorporates two types of learning, transfer learning and semi-supervised learning, into its functionality. As this research learns the ideal attributes that have the greatest impact on energy consumption, the accuracy of transfer learning improves. Furthermore, this research limits the AP technique to produce the actual number of VM clusters rather than a random number of VM clusters using minor side information such as the number of classes. Pairwise constraints, a semi-supervised method, have also contributed to an increase in accuracy. Nearly 831 of the 1250 virtual machines are successfully detected in energy-consuming modes, according to the 66.48 percent accuracy. Although, because CPU and memory are the largest energy consumers in a host [139], this research chooses only the best four features by performing a χ^2 test. Disk throughputs, on the other hand, contribute to a host's energy usage; thus, if more features are included, the precision of locating similar VMs can be improved. The accuracy can be improved by

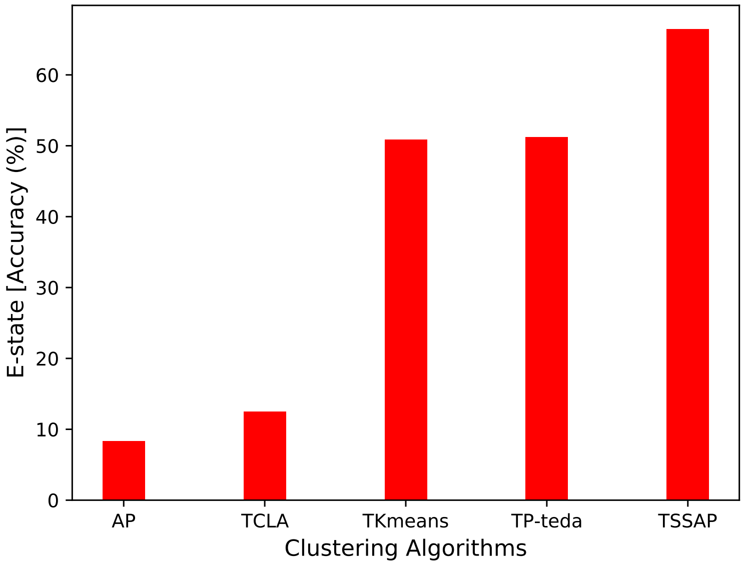


Figure 4-37 Clustering Accuracy Comparison for E-state Evaluated Using Micro-precision

applying multiple state-of-the-art clustering works and several sorts of ML algorithms to analyse the features instead of the χ^2 test and the t-SNE. This research is mostly interested in putting its new energy-saving notion into practise, which is to discover similar virtual machines based on features that affect energy use largely at the VM level.

4.9 Summary

This research investigated workload and energy state estimation in cloud data centres in this chapter. Predicting workload in advance has grown difficult due to the high non-linearity of data centre workload, and existing ML-based workload prediction systems primarily evaluate the utilisation metrics CPU and memory, ignoring other crucial characteristics. Provisioned resources like CPU and RAM are also responsible for energy consumption when a new VM is created on a host, in addition to real utilisation levels. Furthermore, disc and network throughput have an impact on the host's energy consumption. Data centre energy management requires visibility into energy utilisation. The virtualized platform, unlike the host in modern data centres, does not have any built-in sensors to monitor energy use. Furthermore, evaluating the energy usage of VM resources like CPU, memory, and disc at the software level is challenging. However, the current work proposes energy models that quantify energy at the VM level using the VM resource performance of CPU, memory, and disc. However, in order to evaluate memory energy usage, this research must gather the last-level cache (LLC) events triggered by each VM on each core, which is incredibly difficult to obtain, further complicating the measurement.

In this case, this research provided a machine learning-based model with a Prediction Module to handle the two tasks mentioned above. This research looked at a variety of machine learning algorithms, including LR, RR, ARDR, EN, and GRU, a deep learning method. Its predictions, which are based on the best-performing model, assist RMS in making effective decisions. Instead of monitoring each VM's energy consumption, this research came up with the new notion of classifying comparable VMs into various groups based on variables that affect energy consumption in the second job. Because clustering analysis is a powerful tool for analysing data similarities, this research chooses it as the method of choice for this work. TSSAP, TCLA, TKmeans, and TP-teda are four different clustering methods this research suggested to discover related groups of different energy-consuming states (E-state). The following are the primary advantages of the model: (1) It is evaluated using real workload traces that include both provisioned and utilised resources, as well as all metrics performance such as provisioned CPU, provisioned memory, CPU utilisation, memory utilisation, disc throughput, and network throughput; (2) It is efficient and adaptable because it can select the best results from a variety of machine learning methods; and (3) It makes use of semi-sustainably provisioned CPU, provisioned memory, CPU utilisation, memory utilisation, disc throughput, and network through

Based on the best-performing findings of multiple ML approaches given in this work, this research plans to incorporate the RMS component of the model for resource provisioning and VM consolidation in the future. In order to increase workload forecast accuracy and performance across all measures, more sophisticated models will be investigated. Several grouping and learning methods, such as kernel learning rather than paired constraints, will be investigated in the future.

Chapter 5 Ambient Temperature Prediction of Hosts in Cloud

5.1 Outline

According to the research review in Chapter 2, another key system state that needs to be predicted is temperature, which will affect the effective energy conversion rate of the entire system. In the previous chapter, we studied the prediction of system load and energy consumption. In this chapter, we mainly consider the prediction of temperature in the system environment.

In recent years, cloud computing has revolutionized computing, but its data centres hosting cloud services consume an enormous amount of energy. Hyper-scale cloud data centers have a critical problem with thermal management. Hotspots result from an increased host temperature increase cooling costs and affect reliability. It is imperative to accurately predict host temperatures in order to manage resources effectively. Due to thermal variations in the data center, estimating temperature is a non-trivial task. Current solutions for estimating temperature are inefficient because of their computational complexity and inaccuracy. The use of ML to predict temperature using data is a promising approach. Additionally, researchers are putting consistent efforts into improving it. Current works do not consider the train and test root mean square error (RMSE) values to ensure consistent, reliable, and accurate predictions. The aim of this chapter is to present a model for predicting ambient temperatures (a combination of CPU and inlet temperatures) based on a hybrid GRU and Recurrent neural network. Models can learn from single input normalized data, which must be predicted along with both train and test RMSE values observed in order to ensure the validity of the proposed model. This study performed the experiments on a benchmark dataset from the University of Melbourne that consists of several physical machine features and compared them with state-of-the-art algorithms.

5.2 Introduction

The cost of operating a server is increasingly influenced by thermal concerns. A primary factor limiting peak performance is thermal effects. As a result of this metric, the server may be able to carry out reasonably long intervals of execution at a maximum amount of heat and power, with short-lived crossings over this threshold (on the order of microseconds) being allowed. Server racks in modern cloud data centers can consume

1,000 watts each and reach temperatures exceeding 100 degrees Celsius [170]. The electricity spent by a host is lost as heat into the environment, and the cooling system is designed to remove this heat and keep the temperature of the host below the threshold [93]. An increase in host temperature is a bottleneck for a data center's normal operation because it raises cooling costs. It also causes hotspots, which have a negative impact on system reliability due to cascading failures produced by damaged silicon components.

Commercial data centres are under tremendous pressure to minimise cooling costs and carbon emissions. A new paradigm is emerging in response to this pressure, allowing for higher inlet water temperatures for the liquid cooling frequently used in these systems. These systems are designed to make use of the server processors' thermal headroom (also known as the guard band).

As a result, this study requires accurate estimates of thermal dissipation and power consumption of hosts based on workload level to limit the danger of peak temperature ramifications and to save a considerable amount of energy. Accurately predicting a host temperature in a steady-state data centre, on the other hand, is a difficult challenge [157, 171]. As a result, estimating the host temperature in the face of such differences is critical for effective thermal management. To sense the CPU and ambient temperature, sensors are installed on both the CPU and the rack. These sensors can be used to determine the present state of the environment's temperature. However, for crucial RMS operations like resource provisioning, scheduling, and regulating the cooling system parameters, anticipating future temperature based on changes in workload level is equally important. To reduce energy use and costs, data-driven solutions based on ML are being investigated. Google has published a list of their work in this direction [21], in which they use ML to optimise numerous of their large-scale computing systems in order to save money, and energy, and improve performance. Furthermore, recent research has looked into ML algorithms for predicting data centre host temperature [93, 157, 172], They do not, however, look at both the train and test root mean square error (RMSE) values of temperature predictions in order to get trustworthy, consistent, and accurate results. To say that the model has trained well on a large data set, the train and test RMSE values should be almost comparable.

This study uses data from our university's proprietary research cloud for this. On the basis of this data, a data-driven model for temperature prediction is proposed. As a result, this chapter contributes to the creation of a temperature prediction model based on a recurrent neural network that can more accurately predict the ambient temperature, which is a mix of CPU and intake temperature.

The proposed model learns and extracts the pattern from workload rather than using simple statistics like mean and others. The patterns discovered are then used to make more predictions. The model is trained using several epochs in order to reduce train and validation loss. These predictions can also be used to suggest a dynamic scheduling approach for lowering the peak temperature of a data centre host [93]. The model was compared to the deep neural network and long-term short-term memory models on a benchmark data set of fifty hosts. With a significant reduction in mean squared prediction error and mean absolute prediction error, the proposed model beat the other two prediction models. A system model has been shown in Figure 5-1.

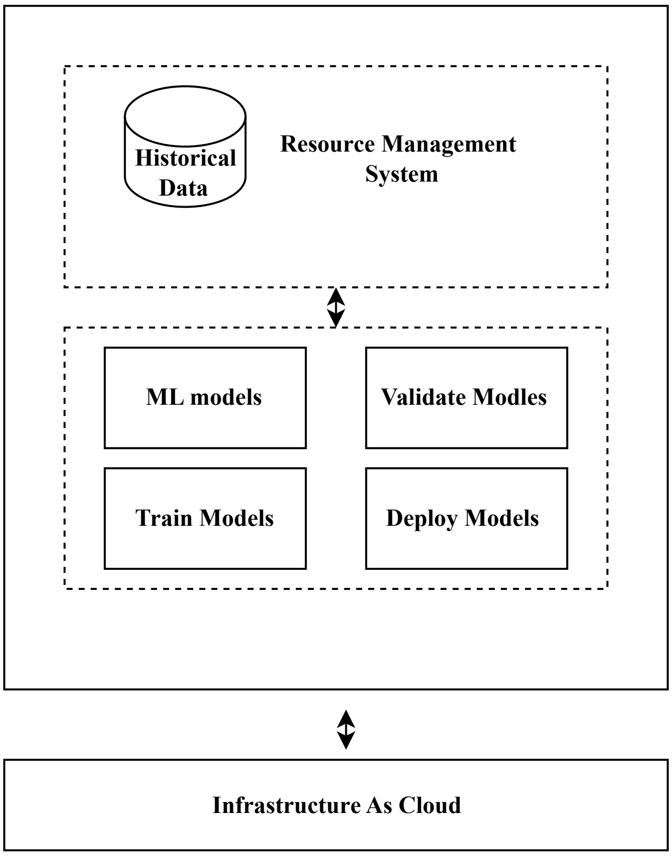


Figure 5-1 System Model

In summary, the following are the major contributions of the work:

(1) Hybrid GRU -RNN-based model is proposed that gives more accurate predictions, as well as tests and trains RMSE values, demonstrating its efficacy in making consistent and dependable predictions.

(2) When the results of the proposed model prediction are compared with those of

two other state-of-the-art prediction algorithms values for almost 8 hosts.

(3) A private cloud from the University of Melbourne is used that includes host information to estimate ambient temperature. These forecasts can be used to develop a dynamic scheduling strategy for lowering peak host temperature and other resource management tasks.

The rest of this chapter is laid out as follows: Predictive modeling is depicted in Section 3. The fourth section discusses performance evaluation. Finally, Section 5 brings this chapter to a conclusion while also pointing to future research areas.

5.3 Related Work

Weatherman, a predictive thermal mapping system for data centres, was introduced by the researchers [173]. They looked at the impact of workload distribution on data centre cooling and temperature settings. These models are intended to detect thermal abnormalities and manage workload at the data centre level, with no regard for accurate temperature forecasting. The researchers demonstrated the use of Artificial Neural Networks to identify thermal anomalies (ANNs) [174]. They employ Self-Organizing Maps (SOM) to detect anomalous data centre behaviour from a previously trained trustworthy performance. They tested their method with anomaly traces from a real data centre. The researchers investigated various ML classifiers for configuring various hardware counters to improve energy efficiency for a specific application [175]. Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Random Forest (RF) were among the 15 classifiers examined. This research just looks at energy as a metric for optimization, ignoring the thermal aspect. Furthermore, these studies are limited to HPC data centres, where temperature estimate for application-specific applications is performed, which necessitates access to application counters. They improved their method in a follow-up paper [157], including more efficient models such as lasso linear and Multilayer Perceptron (MLP). Predictive models are accurate and perform well in data centre resource management, according to the findings. The Gaussian process-based host temperature prediction model in HPC data centres was proposed by researchers [176]. To run the HPC test programs and collect the training data, they employed a two-node Intel Xeon Phi cluster. They also presented a greedy approach for application location in order to reduce thermal variations throughout the system. Many scholars have recently researched thermal management using theoretical analytical models [177,178]. When compared to the real numbers, these models that estimate temperature using mathematical relationships are not accurate enough. Numerous applications have been used to identify the best settings or system configurations to achieve energy efficiency. However, researchers researched ML approaches particular to temperature prediction and suggested the Gaussian process-based host temperature prediction model in HPC data centres [176]. They collected the training data and ran the HPC test programs on a two-node Intel Xeon Phi cluster. They also suggested a greedy approach for the placement of applications in order to reduce heat differences throughout the system. They improved their solution in an extended research [157] by incorporating more effective models like the lasso linear and Multilayer Perceptron (MLP). The findings demonstrate that predictive models are reliable and effective in terms of data centre resource management.

5.4 Predictive Modelling

5.4.1 DataSet

An ML-based prediction model is only as good as the data it was trained on. To train the model in the data centre domain, training data can contain application and physical level features [157]. This research has utilised a dataset from the University of Melbourne private cloud [93]. Instruction count, CPU cycle count, cache metrics (read, write, and miss), and other features are available in this dataset. Accordingly, physical features include host-level resource usage (CPU, RAM, I/O, etc.) and several sensor readings (power, CPU temperature, fan speeds). Physical characteristics include host-level resource consumption (CPU, RAM, I/O, and so on) as well as a variety of sensor readings (power, CPU temperature, fan speeds, and so on). A brief summary of this data is presented in Table 5-1.

Table 5-1 Real-world Data Sets Taken from Different Sources

Hosts	VMs	Cores	Memory	Duration	Interval
75	650	9600	38692 GB	90 days	10 Minute

It contains logs from 75 physical hosts with an average of 650 virtual machines. The data is kept for three months, with the log interval set to ten minutes.

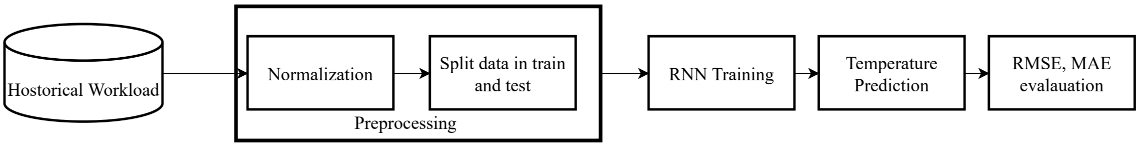


Figure 5-2 Model Workflow

5.4.2 Prediction Models

The prediction method entails a number of phases as shown in Figure 5-2. Data X is normalised in the range (0, 1) in the preprocessing step using Eq (1). Where X_{min} and X_{max} are minimum and maximum values respectively obtained from the dataset. The network receives the value of normalised data x as input. Following that, the network is trained and evaluated, and temperature predictions are made.

$$x_i = \frac{X_i - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{5-1}$$

The predictive model is made up of three layered neural networks as shown in Figure 5-3. The neural network is made up of logical units known as neurons [179]. Different nodes use the Rectified Linear Unit (ReLU) activation function. Eq. (2) can be used to determine the activation function for a node. The model is trained using the supervised learning method. Section 3 delves into the topic of training in greater depth. To anticipate incoming workload on the data centre at time instance n + 1, the predictive model extracts patterns from the real workload and analyses n prior workload values. The RNN model and other comparison methods, DNN and LSTM, are implemented using the Sci-kit learn package. This research has selected epoch = 100, batch size = 30, verbose = 1 parameter setting for RNN. The reason for taking epoch =100 has been explained in further sections.

$$y = \max(0, X) \tag{5-2}$$

In this chapter, a hybrid model that combines GRU and RNN models is proposed. In order to provide a more accurate and trustworthy forecast on the streamflow data set, this hybrid model is considered to take advantage of both the strengths of the GRU and RNN models as well as their characteristics and learning capabilities. Train and test data sets are split 8:2 in ratio. The data set was split into two sets, a training data set and a testing data set, each of which contains a number of processes, in order to validate the

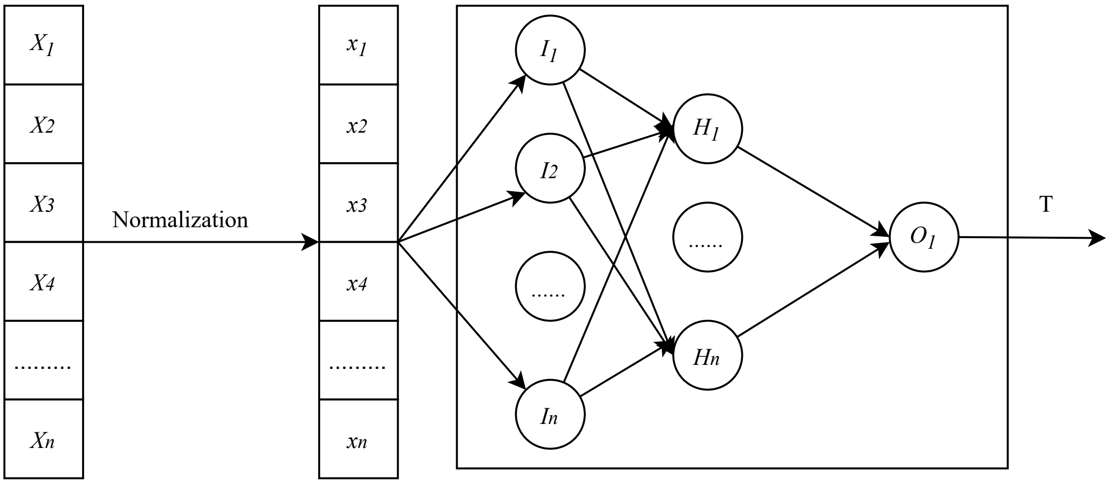


Figure 5-3 Workload Forecaster

efficacy of the suggested hybrid model. The training part is the first step, and it entails normalising the data set between 0 and 1 using Min Max Scaler, a straightforward method for fitting data into a predefined boundary. The time series data input is segmented in the subsequent phase using a sliding window to calculate the prediction accuracy, which was set at 3. As it is concatenated and built into a fully connected layer, the RNN model's output is fed into the GRU model to create a single, final output. The hybrid model is used to train and test the network, and each of the fully connected hybrid models has its epoch, batch size, and verbose set to 31 and 2, respectively. Since this is the multi-layer perception used throughout the study, the configuration chosen for the hybrid model in this study is 1-2-1, or one input layer, two hidden layers with the first hidden layer having 5 GRU neurons and the second hidden layer having 5 GRU neurons, and an output layer. The performance of the model was assessed at the training and testing phases, respectively, using the RMSE of the training and testing data set. The proposed hybrid model's flowchart, which includes hybrid blocks with a fully connected hidden layer and shows the data flow from the input to the output state, is shown in Figure 5-4. The GRU-RNN is compared with blackhole algorithm [180], deep learning [181], differential evolution [182] and back-propagation algorithm [183].

The recorded ambient temperature observed over time is retained as historical data and used as input in order to construct a neural network model for temperature prediction. The input data is then separated into two parts: training data and testing data. The training data is used to train the prediction model, while the testing data is used to evaluate the model's prediction accuracy. This research used 80% of the data for training purposes and 30% for testing purposes. The accuracy of the model over unseen patterns is evaluated and measured using the Root Mean Squared Error (RMSE) once it has been trained. 8 hosts are selected from the dataset and compared RMSE values with state of art algorithms.

Each host in the cluster has two CPUs that are managed by the same operating system in the specified dataset. This research wants to create a model for each host that appropriately represents its thermal behaviour. As a result, rather than estimating CPU temperature alone, this research forecasts the host ambient temperature (T), which is a combination of inlet and CPU temperature [184]. There are various reasons to evaluate ambient temperature rather than CPU temperature. First, by integrating the inlet and CPU temperatures, thermal fluctuations caused by both the intake and CPU temperatures may be captured [93]. Second, rather than individual CPU temperature, cooling settings knobs in data centres are modified based on host ambient temperature [173].

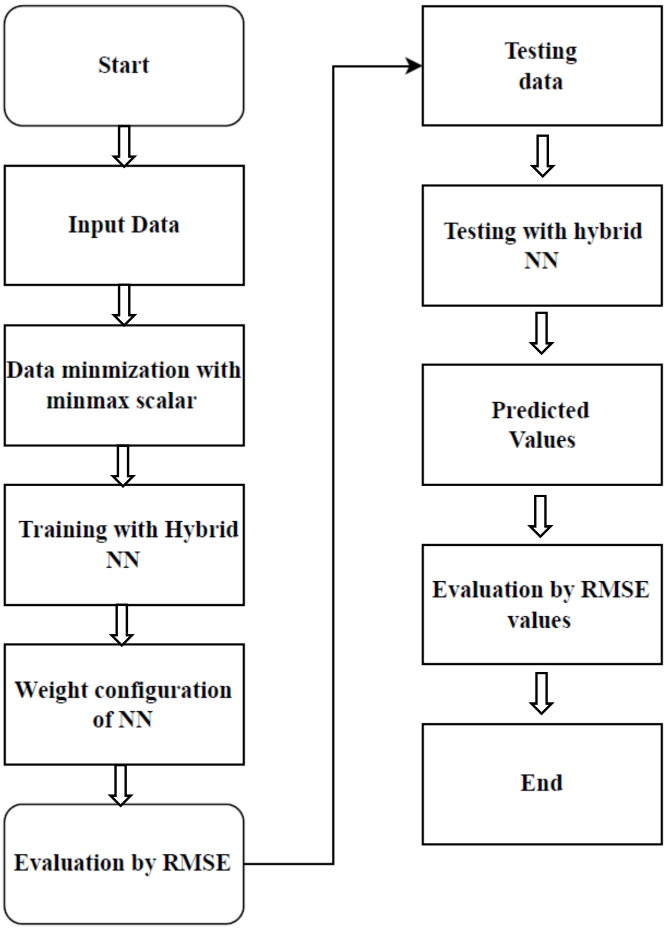
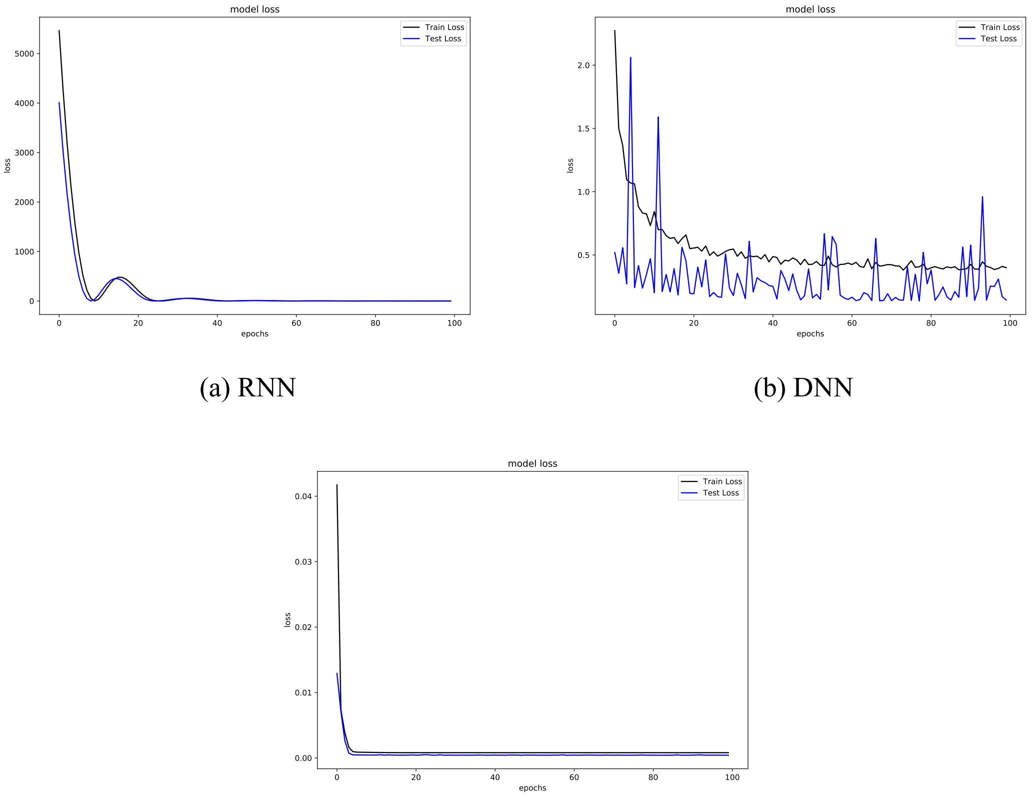


Figure 5-4 Flowchart of Hybrid Model



(a) RNN

(b) DNN

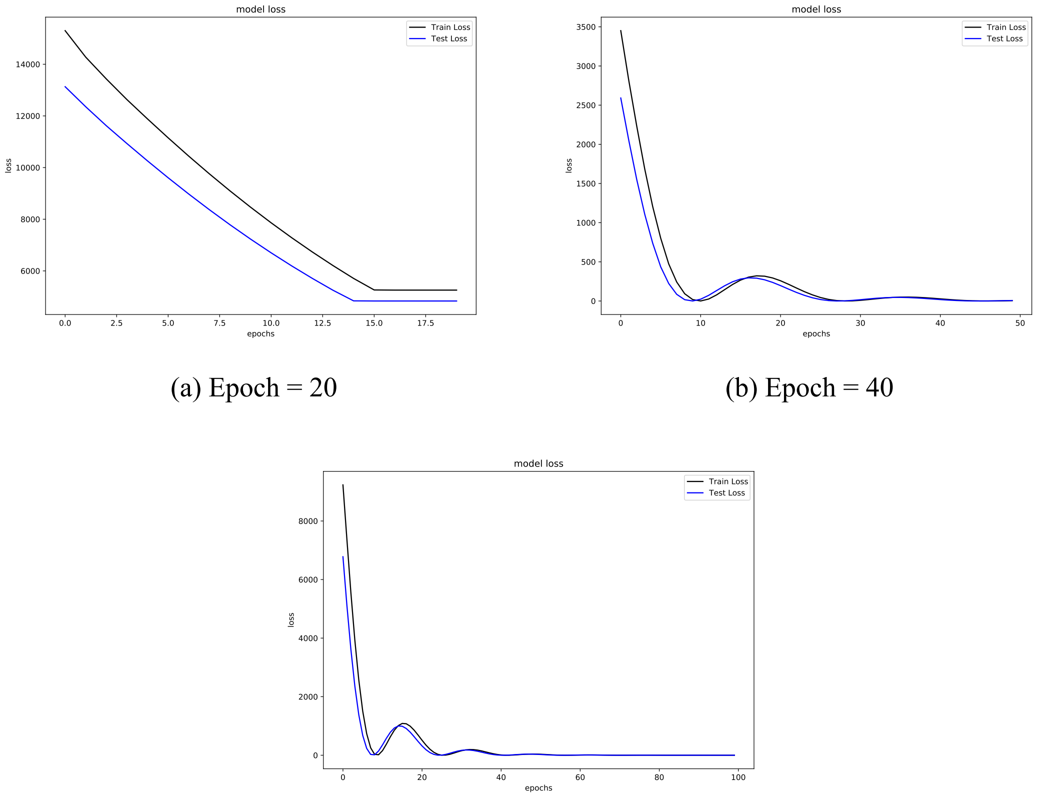
(c) LSTM

Figure 5-5 Training Process Using Different Models for a Host

5.5 Performance Analysis

5.5.1 Experimental Setup

The tests are run on a system with a 1.90 GHz Intel(R) Core(TM) i3-4030U processor and 4 GB of main memory. The proposed model's Prediction model uses PyCharm Community 2020.2 to design several deep learning algorithms with the proposed method GRU-RNN and predicts ambient temperature. This research uses the sci-kit learn [185] package to implement all techniques. These methods are also implemented using the Keras [163] deep learning framework. Because the model requires the most promising forecasts from all of the prediction algorithms applied, all of the prediction algorithms are compared using RMSE to see which one has the fewest residual errors when compared to actual data. For GRU-RNN, the input data is normalised using a min-max scalar. For all algorithms, the parameter settings are set by default.



(a) Epoch = 20

(b) Epoch = 40

(c) Epoch = 100

Figure 5-6 Training Process of the Model Using Different Epoch Numbers for a Host

5.5.2 Evaluation of Results

In the proposed approach, the prediction model is employed to handle a task: workload prediction, i.e., ambient temperature prediction. This research studies and compares the proposed model for temperature prediction with the algorithms mentioned above. For each model, the RMSE values are calculated. The lower the RMSE, the more accurate the forecast. The train and test RMSE values obtained using the various approaches are shown in these tables. The compared algorithms have extremely high RMSE values, implying that they are incapable of providing accurate and dependable predictions. In contrast to these techniques, which have large RMSE values indicating poor performance, RNN only obtains a few train and test RMSE and MAE values for all hosts. RNNs have a sense of memory, which aids them in remembering what happened previously in the time series data, allowing them to obtain context and detect correlations and patterns. In this case, RNN performs better as the model has learned during training very well with a smooth graph coinciding at epoch =100 as shown in Figure 5-5 (a). The model train vs validation loss graphs are shown in Figures from 5-5 (a-c). The loss in Figure 5-5 (c) is obviously degrading smoothly and coinciding at the terminal point. They have RMSE values of 0.16 and 0.15 for both Train and Test, which are nearly equal for host-5. The model in Figure 5-5 has been less adequately trained. They had RMSE values of 0.85 and 0.96 in training and testing, as well as MAE values of 0.63 and 0.61 in training and testing, which are not nearly equal. For LSTM, the RMSE values are 0.55, and 0.44, while the MAE values are 0.33 and 0.25. As a result of the model's excellent training, RNN performed consistently on the dataset. The results also suggest that RNN gives greater accuracy when the dataset is large. A huge dataset reduces the accuracy of LSTM AND DNN, as shown in the table. Furthermore, this research can expect reduced residual errors in prediction offered by RNN with a larger training dataset and more hyperparameter adjustment if this research uses a better infrastructure, such as a GPU cluster.

To train the RNN model, this research employed different hyperparameter values for different epochs. Epochs are the number of iterations across which the input data is delivered. Figure 5-6 depicts how the model is trained using different epoch numbers for a host, such as 20, 50, and 100. It is considered that if the model is well-trained on the data, it will perform better. The model trained at epochs = 100 has the least trained RMSE score and MAE values, as shown in Figure 5-6 (c). The loss graph in Figure 5-6 (c) coincides, indicating that the model has been well-trained. The highest RMSE and MAE values are shown in Figures 5-6 (a) and 5-6 (b). For these reasons, this research sets all algorithms to epochs = 100. The model performs similarly on both training and validation data, as seen by the loss graphs. The learning should be terminated if these two loss plots begin to move consistently. Figure 5-6 (c) displays a steady movement at epoch = 100, demonstrating that the model has learned quite well. By fine-tuning hyperparameters, the model can be trained more effectively.

GRU-RNN outperformed BA by 4.42%, DL by 2.35%, DE by 9.59% and BPA by 2.66% in train RMSE values. For test RMSE values, GRU-RNN outperformed BA by 3.05, DL by 2.37, DE by 2.56, and BPA by 6.29. For host 1, the proposed hybrid model achieved equal RMSE values with DL in Table 5-2. For host 3, BA, DL, DE and BPA obtained almost equal RMSE values. Moreover, host 4 also obtained the nearest equal RMSE values for BA, DL, DE and BPA. For host 5, BA DL DE obtained almost equal RMSE values. For host 6, BA, and DE obtained almost equal RMSE values. For host

7, BA, and DE obtained almost equal RMSE values. In Table 5-3, for host 3, BA, and BPA obtained almost equal RMSE values. For host 4, DL and DE obtained equal RMSE values. For host 5, BA and DL obtained equal RMSE values. For host 8, DL and DE obtained equal RMSE values.

Table 5-2 Train: RMSE Values

Hosts	GRU-RNN	BA	DL	DE	BPA
HOST 1	4.5	5.1	4.5	1.2	10.55
HOST 2	1.7	75.1	7.9	11.9	10.9
HOST 3	9.8	24.9	25.3	24.9	24.3
HOST 4	12.9	44.03	44.8	43.8	44.5
HOST 5	1.5	7.0	7.0	7.2	66.8
HOST 6	0.7	4.7	3.3	4.4	5.9
HOST 7	0.9	6.4	6.5	6.8	6.9
HOST 8	0.1	6.9	6.4	6.5	66.8
SUM	32.1	174.13	107.4	327.8	117.5

Table 5-3 Test: RMSE Values

Hosts	TGRU	BA	DL	DE	BPA
HOST 1	3.5	4.1	5.5	2.2	9.55
HOST 2	0.7	7.1	7.8	15.9	19.9
HOST 3	9.9	20.9	19.3	14.9	20.3
HOST 4	0.9	4.03	4.8	4.8	4.5
HOST 5	1.5	6.0	6.0	9.2	60.8
HOST 6	0.7	5.7	4.3	4.4	5.0
HOST 7	0.9	6.8	6.4	6.6	6.0
HOST 8	0.1	6.7	6.8	6.8	6.8
SUM	18.2	73.83	61.4	64.8	132.85

5.6 Summary

Temperature forecast accuracy can be utilised to reduce data centre energy consumption and operating costs. However, it is a difficult and time-consuming task to estimate the temperature in a data centre. Existing temperature prediction methods are imprecise and computationally expensive, and they do not take into account train and test RMSE values for trustworthy and consistent results. Optimal thermal management combined with accurate temperature prediction can lower data centre operating costs while also increasing reliability. Because this research was able to take into account CPU and intake airflow temperature variations through measurements, data-driven temperature estimation of hosts in a data centre can offer us a more accurate prediction than simple mathematical models. In this chapter, this research looks at a dataset with thermal variations and proposes a model based on recurrent neural networks RNN that takes into account both training and testing results to ensure that the predictions are stable. The train and test RMSE values are documented and compared to see if they are nearly similar or not to guarantee the proposed model's training validity. The proposed model achieved an average RMSE value that was 82.2% less than the average from DNN and 61.36% less than the average from LSTM respectively. As part of future work, this research intends to develop a dynamic scheduling technique driven by temperature prediction for energy-efficient execution of applications.

Chapter 6 Conclusions and Future Directions

The dissertation comes to a close with this chapter, which includes a summary of works and major contributions.

6.1 Summary and Conclusions

Clustering is the division of data into groups that are related and distinct. Clustering challenges are commonly affected by issues of accuracy across diverse datasets. As a result, finding a clustering algorithm that works in all cases is quite challenging. As a result, improved clustering algorithms that can deal with these limitations are required. This research focuses on semi-supervised clustering in this case, which combines some supervised side information with unsupervised data to improve accuracy. This research offers new clustering methods and shows how they can be used to anticipate energy consumption in cloud data centres. Cloud computing platforms enable highly networked resource-intensive business, scientific, and personal applications by providing on-demand and flexible access to elastic resources. Demand for cloud computing has risen in distributed, large-scale, and heterogeneous data centres. To achieve cloud computing sustainability, it is critical to manage resources and energy efficiently in such architecture. It is also vital to provide dependable services to application users by meeting their SLA criteria. In current cloud systems, state-of-the-art rule-based or heuristics-based Resource Management Systems (RMS) solutions have proven insufficient. This research focuses on machine-learning-based predictions using regression-based techniques, and this research presents semi-supervised clustering algorithms for predicting non-linear workload and energy consumption state. The following significant contributions to the state-of-the-art are made in this research:

The challenges of machine-learning-based resource management in a cloud computing environment were discussed in Chapter 2, as well as the various approaches that have been used to solve these challenges in recent years, as well as their benefits and drawbacks. In recent years, the number of studies looking into how to use ML techniques to predict workload, energy consumption, and other tasks has expanded considerably. To solve a variety of challenges, these tactics use a variety of ML methodologies.

6.2 Future Directions

(1) Clustering analysis, which does not need data labeling, might be used to characterise service renters as a future study topic. Based on historical resource needs, similar patterns of service renters may be automatically retrieved.

(2) The creation of a general ensemble framework for every type of dataset in cloud time series workload data is a future research aim. Deep learning (DL) is a fast-developing and wide-ranging research field including novel architectures.

(3) Two future research directions for avoiding non-linear resource utilisation in modern data centres are dynamic resource provisioning and dynamic VM consolidation, which take into account various types of VM resources such as CPU, memory, and bandwidth, current and future resource needs, and SLAs such as compute-intensive non-interactive jobs and transactional applications.

(4) The use of a fixed threshold for detecting overloaded hosts might result in inaccurate VM migration. If a VM's resource use degrades in a short period of time, there's no need to transfer it. In this case, the technique should include a dynamic resource utilisation threshold that prevents VM migration when it reaches a predetermined level, taking into account data from the near future. This is the next research path in VM consolidation for effective VM migration. In addition, VMs should be moved if there will be a protracted period of load reduction in the near future.

Acknowledgements

I give thanks to the All-Powerful Allah, the Most Generous, the Most Merciful, for giving me the patience and tenacity to move forward in the Right Direction Throughout My Life.

My warmest thanks and gratitude go out to Prof. Wenhong Tian of the School of Information and Software Engineering for his continuous support and direction throughout my project. His meticulous edits always ensured a more targeted approach to problem solutions during routine meetings and conversations. It gives me great pleasure to thank Prof. Rajkumar Buyya, Redmond Barry Distinguished Professor, of the CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Australia, for his support, encouragement, and sage advice as I worked on my research. He was a man of acute scientific temperament, which inspired me.

I would especially want to thank Prof. Lutfullah of the Chemistry Department at Aligarh Muslim University for his invaluable advice and support in motivating me to pursue a PhD. for his assistance and patience throughout my master's program at AMU, Dr. Gulzar-ul-Hasan, Assistant Professor, Madanapalle Institute of Technology & Science. Drs. Faez Iqbal Khan, Jalaluddin Khan, and Ahmad Neyaz Khan have been incredibly kind, knowledgeable, and encouraging throughout my PhD studies, and I want to show my gratitude to them. I want to thank everyone in my CloudSet Lab team, especially Kingsley Nketia Acheampong, Assefa Addis Abebe, and Mustafa R. Khadim, for their important help. For his assistance with this effort, I also want to thank Dr. Shashikant Ilgaer of TU Wien in Vienna, Austria. I'd like to thank Aslam Hasan Khan, the CEO of Sofyrus Technology and Private Limited in Aligarh, with whom I've had many fantastic experiences while conducting my research.

I want to express my gratitude to everyone on the CloudSet Lab team for their crucial assistance, especially Kingsley Nketia Acheampong, Assefa Addis Abebe, and Mustafa R. Khadim. I would like to thank Dr. Shashikant Ilgaer of TU Wien in Vienna, Austria, for his help with this effort. Aslam Hasan Khan, the CEO of Sofyrus Technology & Private Limited in Aligarh, with whom I have had many wonderful interactions while performing my study, is someone I'd want to thank.

References

[1] Birke R, Chen L. Y, Smirni E, et al. Data centers in the wild: A large performance study[J]. IBM Research, Zurich, Switzerland, 2012.

[2] Reiss C, Tumanov A, Ganger G. R, et al. Heterogeneity and dynamicity of clouds at scale: Google trace analysis[C]. Proceedings of the Third ACM Symposium on Cloud Computing, 2012, 1-13.

[3] Aldossary M, Djemame K. Energy-based cost model of virtual machines in a cloud environment[C]. 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), 2018, 1-8.

[4] Gu C, Shi P, Shi S, et al. A tree regression-based approach for vm power metering[J]. IEEE Access, 2015, 3: 610-621.

[5] Zhao-Hui Y, Qin-Ming J. Power management of virtualized cloud computing platform[J]. Chinese Journal of Computers, 2012, 6: 015.

[6] Lin W, Wu W, Wang H, et al. Experimental and quantitative analysis of server power model for cloud data centers[J]. Future Generation Computer Systems, 2018, 86: 940-950.

[7] Buyya R, Srirama S. N, Casale G, et al. A manifesto for future generation cloud computing: Research directions for the next decade[J]. ACM Comput. Surv., 2018, 51(5).

[8] Xu M, Tian W, Buyya R. A survey on load balancing algorithms for virtual machines placement in cloud computing[J]. Concurrency and Computation: Practice and Experience, 2017, 29(12): e4123.

[9] Bianchini R, Fontoura M, Cortez E, et al. Toward ml-centric cloud platforms[J]. Communications of the ACM, 2020, 63(2): 50-59.

[10] Singh A, Kumar J. Secure and energy aware load balancing framework for cloud data centre networks[J]. Electronics Letters, 2019, 55(9): 540-541.

[11] Kumar J, Singh A. K, Buyya R. Self directed learning based workload forecasting model for cloud resource management[J]. Information Sciences, 2020, 543: 345-366.

[12] Barroso L. A, Clidaras J, Hölzle U. The datacenter as a computer: An introduction to the design of warehouse-scale machines[J]. Synthesis lectures on computer architecture, 2013, 8(3): 1-154.

[13] Kumar J, Singh A. K, Buyya R. Self directed learning based workload forecasting model for cloud resource management[J]. Information Sciences, 2021, 543: 345-366.

[14] Li X, Qian Z, Lu S, et al. Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center[J]. Mathematical and Computer Modelling, 2013, 58(5-6): 1222-1235.

[15] Kumar J, Singh A. K. Cloud datacenter workload estimation using error preventive time series forecasting models[J]. Cluster Computing, 2020, 23(2): 1363-1379.

[16] Kumar J, Singh A. K, Buyya R. Ensemble learning based predictive framework for virtual machine resource request prediction[J]. Neurocomputing, 2020.

[17] Kumar J, Singh A. K. Workload prediction in cloud using artificial neural network and adaptive differential evolution[J]. Future Generation Computer Systems, 2018, 81: 41-52.

[18] Ilager S, Muralidhar R, Buyya R. Artificial intelligence (ai)-centric management of resources in modern distributed computing systems[J]. IEEE Cloud Summit, 2020.

[19] Shahidinejad A, Ghobaei-Arani M, Masdari M. Resource provisioning using workload clustering in cloud computing environment: a hybrid approach[J]. Cluster Computing, 2020, 1-24.

[20] Mao H, Schwarzkopf M, Venkatakrishnan S. B, et al. Learning scheduling algorithms for data processing clusters[M]. , 2019, 270-288.

[21] Jeff D. Ml for system, system for ml, keynote talk in workshop on ml for systems, nips[J]. , 2018.

[22] Yadwadkar N. J. Machine learning for automatic resource management in the datacenter and the cloud[D]. , 2018, .

[23] Mao H, Alizadeh M, Menache I, et al. Resource management with deep reinforcement learning[C]. Proceedings of the 15th ACM Workshop on Hot Topics in Networks, 2016, 50-56.

[24] Cao R, Yu Z, Marbach T, et al. Load prediction for data centers based on database service[C]. 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), 2018, 728-737.

[25] Chen S, Shen Y, Zhu Y. Modeling conceptual characteristics of virtual machines for cpu utilization prediction[C]. International Conference on Conceptual Modeling, 2018, 319-333.

[26] Cortez E, Bonde A, Muzio A, et al. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms[C]. Proceedings of the 26th Symposium on Operating Systems Principles, 2017, 153-167.

[27] Gao J. Machine learning applications for data center optimization (google white paper)[J]. , 2014.

[28] Sun X, Ansari N, Wang R. Optimizing resource utilization of a data center[J]. IEEE Communications Surveys & Tutorials, 2016, 18(4): 2822-2846.

[29] Manvi S. S, Shyam G. K. Resource management for infrastructure as a service (iaas) in cloud computing: A survey[J]. Journal of network and computer applications, 2014, 41: 424-440.

[30] Zhang J, Huang H, Wang X. Resource provision algorithms in cloud computing: A survey[J]. Journal of Network and Computer Applications, 2016, 64: 23-42.

[31] Braiki K, Youssef H. Resource management in cloud data centers: a survey[C]. 2019 15th international wireless communications & mobile computing conference (IWCMC), 2019, 1007-1012.

[32] Jennings B, Stadler R. Resource management in clouds: Survey and research challenges[J]. Journal of Network and Systems Management, 2015, 23(3): 567-619.

[33] Usmani Z, Singh S. A survey of virtual machine placement techniques in a cloud data center[J]. Procedia Computer Science, 2016, 78: 491-498.

[34] Helali L, Omri M. N. A survey of data center consolidation in cloud computing systems[J]. Computer Science Review, 2021, 39: 100366.

[35] Mell P. The nist definition of cloud computing[J]. In N. I. o. S. a. Technology (Ed.): U.S. Department of Commerce, 2011, : .

[36] Hamdaqa M, Tahvildari L. Cloud computing uncovered: a research landscape[M]. Elsevier, 2012, 41-85.

[37] Yakimenko O. A, Slegers N. J, Bourakov E. A, et al. Mobile system for precise aero delivery with global reach network capability[C]. 2009 IEEE International Conference on Control and Automation, 2009, 1394-1398.

[38] Wischik D, Handley M, Braun M. B. The resource pooling principle[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(5): 47-52.

[39] Amazon E. Amazon elastic compute cloud (amazon ec2)[J]. Amazon Elastic Compute Cloud (Amazon EC2), 2010, 5: 18-23.

[40] Espadas J, Molina A, Jiménez G, et al. A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures[J]. Future Generation Computer Systems, 2013, 29(1): 273-286.

[41] Piraghaj S. F, Dastjerdi A. V, Calheiros R. N, et al. A survey and taxonomy of energy efficient resource management techniques in platform as a service cloud[J]. Handbook of Research on End-to-End Cloud Computing Architecture Design, 2017, 410-454.

[42] Jula A, Sundararajan E, Othman Z. Cloud computing service composition: A systematic literature review[J]. Expert systems with applications, 2014, 41(8): 3809-3824.

[43] Whaiduzzaman M, Sookhak M, Gani A, et al. A survey on vehicular cloud computing[J]. Journal of Network and Computer applications, 2014, 40: 325-344.

[44] Toosi A. N, Calheiros R. N, Buyya R. Interconnected cloud computing environments: Challenges, taxonomy, and survey[J]. ACM Computing Surveys (CSUR), 2014, 47(1): 1-47.

[45] Jadeja Y, Modi K. Cloud computing-concepts, architecture and challenges[C]. 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), 2012, 877-880.

[46] Dillon T, Wu C, Chang E. Cloud computing: issues and challenges[C]. 2010 24th IEEE international conference on advanced information networking and applications, 2010, 27-33.

[47] Tuli S, Sandhu R, Buyya R. Shared data-aware dynamic resource provisioning and task scheduling for data intensive applications on hybrid clouds using aneka[J]. Future Generation Computer Systems, 2020, 106: 595-606.

[48] Jordan M. I, Mitchell T. M. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255-260.

[49] Janai J, Güney F, Behl A, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art[J]. Foundations and Trends® in Computer Graphics and Vision, 2020, 12(1–3): 1-308.

[50] Deng L, Li X. Machine learning paradigms for speech recognition: An overview[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2013, 21(5): 1060-1089.

[51] Olsson F. A literature survey of active machine learning in the context of natural language processing[J]. , 2009.

[52] Chin K, Hellebrekers T, Majidi C. Machine learning for soft robotic sensing and control[J]. Advanced Intelligent Systems, 2020, 2(6): 1900171.

[53] Stilgoe J. Machine learning, social learning and the governance of self-driving cars[J]. Social studies of science, 2018, 48(1): 25-56.

[54] Bhatia M. P. S, Kumar A. Information retrieval and machine learning: supporting technologies for web mining research and practice[J]. Webology, 2008, 5(2): 5.

[55] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction[M]. Springer Science & Business Media, 2009.

[56] Goodfellow I, Bengio Y, Courville A, et al. Deep learning[M]. MIT press Cambridge, 2016.

[57] Hartigan J. A, Wong M. A. Ak-means clustering algorithm[J]. Journal of the Royal Statistical Society: Series C (Applied Statistics), 1979, 28(1): 100-108.

[58] Guha S, Rastogi R, Shim K. Rock: A robust clustering algorithm for categorical attributes[J]. Information systems, 2000, 25(5): 345-366.

[59] Ding C, He X, Zha H, et al. Adaptive dimension reduction for clustering high dimensional data[C]. 2002 IEEE International Conference on Data Mining, 2002. Proceedings., 2002, 147-154.

[60] Kim Y. Convolutional neural networks for sentence classification[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, 1746-1751.

[61] Yin W, Schütze H. Multichannel variable-size convolution for sentence classification[C]. Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China, 2015, 204-214.

[62] Yang J, Yu K, Gong Y, et al. Linear spatial pyramid matching using sparse coding for image classification[C]. 2009 IEEE Conference on computer vision and pattern recognition, 2009, 1794-1801.

[63] Bazi Y, Melgani F. Gaussian process approach to remote sensing image classification[J]. IEEE transactions on geoscience and remote sensing, 2009, 48(1): 186-197.

[64] Ciregan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification[C]. 2012 IEEE conference on computer vision and pattern recognition, 2012, 3642-3649.

[65] Sen P. C, Hajra M, Ghosh M. Supervised classification algorithms in machine learning: A survey and review[M]. Springer, 2020, 99-111.

[66] Celebi M. E, Aydin K. Unsupervised learning algorithms[M]. Springer, 2016.

[67] Engelen J. E. V, Hoos H. H. A survey on semi-supervised learning[J]. Machine Learning, 2020, 109(2): 373-440.

[68] Kober J, Bagnell J. A, Peters J. Reinforcement learning in robotics: A survey[J]. The International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[69] Haghshenas K, Mohammadi S. Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic vm consolidation in data centers[J]. The Journal of Supercomputing, 2020, 1-18.

[70] Chun B, Culler D, Roscoe T, et al. Planetlab: an overlay testbed for broad-coverage services[J]. ACM SIGCOMM Computer Communication Review, 2003, 33(3): 3-12.

[71] Genez T. A, Bittencourt L. F, Fonseca N. Lda , et al. Estimation of the available bandwidth in inter-cloud links for task scheduling in hybrid clouds[J]. IEEE Transactions on Cloud Computing, 2015, 7(1): 62-74.

[72] Duggan M, Duggan J, Howley E, et al. A network aware approach for the scheduling of virtual machine migration during peak loads[J]. Cluster Computing, 2017, 20(3): 2083-2094.

[73] Networking C. V. Cisco global cloud index: Forecast and methodology, 2016–2021[J]. White paper. Cisco Public, San Jose, 2016.

[74] Verma A, Ahuja P, Neogi A. pmapper: power and migration cost aware application placement in virtualized systems[C]. ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, 2008, 243-264.

[75] Shaw R, Howley E, Barrett E. An energy efficient anti-correlated virtual machine placement algorithm using resource usage predictions[J]. Simulation Modelling Practice and Theory, 2019, 93: 322-342.

[76] Brewer E, Ying L, Greenfield L, et al. Disks for data centers[J]. , 2016.

[77] Ilager S, Ramamohanarao K, Buyya R. Thermal prediction for efficient energy management of clouds using machine learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(5): 1044-1056.

[78] Nguyen T. H, Francesco M. D, Yla-Jaaski A. Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers[J]. IEEE Transactions on Services Computing, 2017.

[79] Yang H, Zhao Q, Luan Z, et al. imeter: An integrated vm power model based on performance profiling[J]. Future Generation Computer Systems, 2014, 36: 267-286.

[80] Garg S. K, Toosi A. N, Gopalaiyengar S. K, et al. Sla-based virtual machine management for heterogeneous workloads in a cloud datacenter[J]. Journal of Network and Computer Applications, 2014, 45: 108-120.

[81] Calheiros R. N, Masoumi E, Ranjan R, et al. Workload prediction using arima model and its impact on cloud applications’ qos[J]. IEEE Transactions on Cloud Computing, 2014, 3(4): 449-458.

[82] Amekraz Z, Hadi M. Y. Higher order statistics based method for workload prediction in the cloud using arma model[C]. 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), 2018, 1-5.

[83] Verma M, Gangadharan G, Narendra N. C, et al. Dynamic resource demand prediction and allocation in multi-tenant service clouds[J]. Concurrency and Computation: Practice and Experience, 2016, 28(17): 4429-4442.

[84] Subirats J, Guitart J. Assessing and forecasting energy efficiency on cloud computing platforms[J]. Future Generation Computer Systems, 2015, 45: 70-94.

[85] Kansal A, Zhao F, Liu J, et al. Virtual machine power metering and provisioning[C]. Proceedings of the 1st ACM symposium on Cloud computing, 2010, 39-50.

[86] Cao J, Fu J, Li M, et al. Cpu load prediction for cloud environment based on a dynamic ensemble model[J]. Software: Practice and Experience, 2014, 44(7): 793-804.

[87] Shyam G. K, Manvi S. S. Virtual resource prediction in cloud environment: a bayesian approach[J]. Journal of Network and Computer Applications, 2016, 65: 144-154.

[88] Ismaeel S, Miri A. Using elm techniques to predict data centre vm requests[C]. 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, 2015, 80-86.

[89] Zhu X, Goldberg A. B. Introduction to semi-supervised learning[J]. Synthesis lectures on artificial intelligence and machine learning, 2009, 3(1): 1-130.

[90] Abdelsamea A, El-Moursy A. A, Hemayed E. E, et al. Virtual machine consolidation enhancement using hybrid regression algorithms[J]. Egyptian Informatics Journal, 2017, 18(3): 161-170.

[91] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[92] Cho K, Merriënboer B. V, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[93] Ilager S, Ramamohanarao K, Buyya R. Thermal prediction for efficient energy management of clouds using machine learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(5): 1044-1056.

[94] Vora S, Yang H. A comprehensive study of eleven feature selection algorithms and their impact on text classification[C]. 2017 Computing Conference, 2017, 440-449.

[95] Kadhim M. R, Tian W, Khan T. Rapid clustering with semi-supervised ensemble density centers[C]. 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, 2019, 230-235.

[96] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, 785-794.

[97] Śmieja M, Struski Ł, Figueiredo M. A. A classification-based approach to semi-supervised clustering with pairwise constraints[J]. Neural Networks, 2020, 127: 193-203.

[98] Wagstaff K, Cardie C. Clustering with instance-level constraints[J]. AAAI/IAAI, 2000, 1097: 577-584.

[99] Majeed A. Improving time complexity and accuracy of the machine learning algorithms through selection of highly weighted top k features from complex datasets[J]. Annals of Data Science, 2019, 6(4): 599-621.

[100] Hewamalage H, Bergmeir C, Bandara K. Recurrent neural networks for time series forecasting: Current status and future directions[J]. International Journal of Forecasting, 2021, 37(1): 388-427.

[101] Lai G, Chang W.-C, Yang Y, et al. Modeling long-and short-term temporal patterns with deep neural networks[C]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, 95-104.

[102] Shih S.-Y, Sun F.-K, Lee Hyi . Temporal pattern attention for multivariate time series forecasting[J]. Machine Learning, 2019, 108(8): 1421-1441.

[103] Oord Avan den , Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[C]. The 9th ISCA Speech Synthesis Workshop. ISCA, p. 125., 2016, .

[104] Borovykh A, Bohte S, Oosterlee C. W. Conditional time series forecasting with convolutional neural networks[J]. arXiv preprint arXiv:1703.04691, 2017.

[105] Bai S, Kolter J. Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.

[106] Feurer M, Hutter F. Hyperparameter optimization[M]. Springer, Cham, 2019, 3-33.

[107] Alqurashi T, Wang W. Clustering ensemble method[J]. International Journal of Machine Learning and Cybernetics, 2019, 10(6): 1227-1246.

[108] Boongoen T, Iam-On N. Cluster ensembles: A survey of approaches with recent extensions and applications[J]. Computer Science Review, 2018, 28: 1-25.

[109] Wang C.-D, Lai J.-H, Philip S. Y. Multi-view clustering based on belief propagation[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(4): 1007-1021.

[110] Poteraş C. M, Mihăescu M. C, Mocanu M. An optimized version of the k-means clustering algorithm[C]. 2014 Federated Conference on Computer Science and Information Systems, 2014, 695-699.

[111] Poteraş C. M, Mocanu M. L. Evaluation of an optimized k-means algorithm based on real data[C]. 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), 2016, 831-835.

[112] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[113] Rafailidis D, Daras P. The tfc model: Tensor factorization and tag clustering for item recommendation in social tagging systems[J]. IEEE Transactions on Systems, Man and Cybernetics:Systems, 2012, 43(3): 673-688.

[114] Rajpathak D. G, Singh S. An ontology-based text mining method to develop d-matrix from unstructured text[J]. IEEE Transactions on Systems, Man and Cybernetics: Systems, 2013, 44(7): 966-977.

[115] Nie F, Shi S, Li X. Auto-weighted multi-view co-clustering via fast matrix factorization[J]. Pattern Recognition, 2020, 102: 107207.

[116] Jain A. K. Data clustering: 50 years beyond k-means[J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[117] Fred A. L, Jain A. K. Combining multiple clusterings using evidence accumulation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(6): 835-850.

[118] Su P, Shang C, Shen Q. A hierarchical fuzzy cluster ensemble approach and its application to big data clustering[J]. Journal of Intelligent & Fuzzy Systems, 2015, 28(6): 2409-2421.

[119] Yousefnezhad M, Zhang D. Weighted spectral cluster ensemble[C]. 2015 IEEE International Conference on Data Mining, 2015, 549-558.

[120] Topchy A, Jain A. K, Punch W. Clustering ensembles: Models of consensus and weak partitions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(12): 1866-1881.

[121] Frey B. J, Dueck D. Clustering by passing messages between data points[J]. science, 2007, 315(5814): 972-976.

[122] Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions[J]. Journal of Machine Learning Research, 2002, 3(Dec): 583-617.

[123] Yu Z, Zhu X, Wong H.-S, et al. Distribution-based cluster structure selection[J]. IEEE Transactions on Cybernetics, 2016, 47(11): 3554-3567.

[124] Huang D, Wang C, Peng H, et al. Enhanced ensemble clustering via fast propagation of cluster-wise similarities[J]. IEEE Transactions on Systems, Man and Cybernetics: Systems, 2018, (): 1-13.

[125] Haghtalab S, Xanthopoulos P, Madani K. A robust unsupervised consensus control chart pattern recognition framework[J]. Expert Systems With Applications, 2015, 42(19): 6767-6776.

[126] Ramasso E, Placet V, Boubakar M. L. Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites[J]. IEEE Transactions on Instrumentation and Measurement, , 64(12): 3297-3307.

[127] Kadhim M. R, Zhou G, Tian W. A novel self-directed learning framework for cluster ensemble[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(10): 7841-7855.

[128] Zhou P, Wang X, Du L, et al. Clustering ensemble via structured hypergraph learning[J]. Information Fusion, 2022, 78: 171-179.

[129] Hu J, Li T, Wang H, et al. Hierarchical cluster ensemble model based on knowledge granulation[J]. Knowledge-Based Systems, 2016, 91: 179-188.

[130] Iqbal A. M, Moh'd A, Khan Z. Semi-supervised clustering ensemble by voting[J]. arXiv preprint arXiv:1208.4138, 2012.

[131] Givoni I, Frey B. Semi-supervised affinity propagation with instance-level constraints[C]. Artificial Intelligence and Statistics, , 161-168.

[132] Zhou Z.-H, Tang W. Clusterer ensemble[J]. Knowledge-Based Systems, 2006, 19(1): 77-83.

[133] Wang H, Shan H, Banerjee A. Bayesian cluster ensembles[J]. Statistical Analysis and Data Mining: The ASA Data Science Journal, 2011, 4(1): 54-70.

[134] Li H, Wang M, Hua X.-S. Msra-mm 2.0: A large-scale web multimedia dataset[C]. 2009 IEEE International Conference on Data Mining Workshops, 2009, 164-169.

[135] Hancer E. A new multi-objective differential evolution approach for simultaneous clustering and feature selection[J]. Engineering Application of Artificial Intelligence, 2020, 87: 103307.

[136] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[137] Hsieh S.-Y, Liu C.-S, Buyya R, et al. Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers[J]. Journal of Parallel and Distributed Computing, 2020, 139: 99-109.

[138] Farahnakian F, Pahikkala T, Liljeberg P, et al. Energy-aware vm consolidation in cloud data centers using utilization prediction model[J]. IEEE Transactions on Cloud Computing, 2016.

[139] Roose P, Soltane M, Makhlouf D, et al. Predictions & modeling energy consumption for it data center infrastructure[C]. Advances in Intelligent Systems and Computing, 2018, 1-11.

[140] Shen S, Beek Vvan , Iosup A. Statistical characterization of business-critical workloads hosted in cloud datacenters[C]. 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015, 465-474.

[141] Wang Z, Yu Z, Chen C. P, et al. Clustering by local gravitation[J]. IEEE transactions on cybernetics, 2017, 48(5): 1383-1396.

[142] Jain A. K, Dubes R. C. Algorithms for clustering data[M]. Prentice-Hall, Inc., 1988.

[143] Gu X, Angelov P. P, Gutierrez G, et al. Parallel computing teda for high frequency streaming data clustering[C]. INNS Conference on Big Data, 2016, 238-253.

[144] Islam S, Keung J, Lee K, et al. Empirical prediction models for adaptive resource provisioning in the cloud[J]. Future Generation Computer Systems, 2012, 28(1): 155-162.

[145] Barati M, Sharifian S. A hybrid heuristic-based tuned support vector regression model for cloud load prediction[J]. The Journal of Supercomputing, 2015, 71(11): 4235-4259.

[146] Farahnakian F, Liljeberg P, Plosila J. Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers[C]. 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, 2013, 357-364.

[147] Farahnakian F, Pahikkala T, Liljeberg P, et al. Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers[C]. 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013, 256-259.

[148] Chen Q, Grosso P, Veldt Kvan der , et al. Profiling energy consumption of vms for green cloud computing[C]. 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 2011, 768-775.

[149] Wen C, Long X, Yang Y, et al. System power model and virtual machine power metering for cloud computing pricing[C]. 2013 Third International Conference on Intelligent System Design and Engineering Applications, 2013, 1379-1382.

[150] Krishnan B, Amur H, Gavrilovska A, et al. Vm power metering: feasibility and challenges[J]. ACM SIGMETRICS Performance Evaluation Review, 2011, 38(3): 56-60.

[151] Quesnel F, Mehta H. K, Menaud J.-M. Estimating the power consumption of an idle virtual machine[C]. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, 268-275.

[152] Jiang Z, Lu C, Cai Y, et al. Vpower: Metering power consumption of vm[C]. 2013 ieee 4th international conference on software engineering and service science, 2013, 483-486.

[153] Basmadjian R, Ali N, Niedermeier F, et al. A methodology to predict the power consumption of servers in data centres[C]. Proceedings of the 2nd international conference on energy-efficient computing and networking, 2011, 1-10.

[154] Li Z, Yu X, Yu L, et al. Energy-efficient and quality-aware vm consolidation method[J]. Future Generation Computer Systems, 2020, 102: 789-809.

[155] Hariharasubramanian M. Improving application infrastructure provisioning using resource usage predictions from cloud metric data analysis[D]. , 2018, .

[156] Hieu N. T, Francesco M. D, Ylä-Jääski A. Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers[J]. IEEE Transactions on Services Computing, 2020, 13(1): 186-199.

[157] Zhang K, Guliani A, Ogrenci-Memik S, et al. Machine learning-based temperature prediction for runtime thermal management across system components[J]. IEEE Transactions on parallel and distributed systems, 2017, 29(2): 405-419.

[158] Iqbal W, Berral J. L, Erradi A, et al. Adaptive prediction models for data center resources utilization estimation[J]. IEEE Transactions on Network and Service Management, 2019, 16(4): 1681-1693.

[159] Tajvidi M. Cloud resource provisioning for end-users: Scheduling and allocation of virtual machines[D]. , 2019, .

[160] Deepika T, Prakash P. Power consumption prediction in cloud data center using machine learning[J]. International Journal of Electrical and Computer Engineering (IJECE), 2020, 10(2): 1524-1532.

[161] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[162] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python[J]. , 2011, 12: 2825–2830.

[163] Ketkar N. Introduction to keras[M]. Springer, 2017, 97-111.

[164] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms[C]. Proceedings of the 23rd international conference on Machine learning, 2006, 161-168.

[165] Jolly E. Pymer4: connecting r and python for linear mixed modeling[J]. Journal of Open Source Software, 2018, 3(31): 862.

[166] Kingma D. P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[167] Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning[J]. Proceedings of the IEEE, 2020.

[168] Saylor J. E, Sundell K, Sharman G. Characterizing sediment sources by non-negative matrix factorization of detrital geochronological data[J]. Earth and Planetary Science Letters, 2019, 512: 46-58.

[169] Soni J, Prabakar N, Upadhyay H. Visualizing high-dimensional data using t-distributed stochastic neighbor embedding algorithm[M]. Springer, 2020, 189-206.

[170] Handbook-Fundamentals A. American society of heating[J]. Refrigerating and Air-Conditioning Engineers, 2009.

[171] Tang Q, Gupta S. K. S, Varsamopoulos G. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(11): 1458-1472.

[172] Luo Y, Wang X, Ogrenci-Memik S, et al. Minimizing thermal variation in heterogeneous hpc systems with fpga nodes[C]. Proceedings of the 2018 IEEE 36th International Conference on Computer Design (ICCD), 2018, 537-544.

[173] Moore J, Chase J. S, Ranganathan P. Weatherman: Automated, online and predictive thermal mapping and management for data centers[C]. Proceedings of the IEEE international conference on Autonomic Computing, 2006, 155-164.

[174] Aransay I, Sancho M, Garcıa P, et al. Self-organizing maps for detecting abnormal thermal behavior in data centers[C]. Processdings of the 8th IEEE International Conference on Cloud Computing (CLOUD), 2015, 138-145.

[175] Imes C, Hofmeyr S, Hoffmann H. Energy-efficient application resource scheduling using machine learning classifiers[C]. Proceedings of the 47th International Conference on Parallel Processing, 2018, 1-11.

[176] Zhang K, Ogrenci-Memik S, Memik G, et al. Minimizing thermal variation across system components[C]. Proceedings of the IEEE International Parallel and Distributed Processing Symposium, 2015, 1139-1148.

[177] Sun H, Stolf P, Pierson J.-M. Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters[J]. Future Generation Computer Systems, 2017, 71: 157-170.

[178] Cao T, Huang W, He Y, et al. Cooling-aware job scheduling and node allocation for overprovisioned hpc systems[C]. 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2017, 728-737.

[179] McCulloch W. S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.

[180] Campos Souza P. Vde , Soares E. A, Guimarães A. J, et al. Autonomous data density pruning fuzzy neural network for optical interconnection network[J]. Evolving Systems, 2021, 12: 899-911.

[181] Liu L, Xie C, Wang R, et al. Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features[J]. IEEE Transactions on Industrial Informatics, 2020, 17(11): 7589-7598.

[182] Ahmad M. F, Isa N. A. M, Lim W. H, et al. Differential evolution: A recent review based on state-of-the-art works[J]. Alexandria Engineering Journal, 2022, 61(5): 3831-3872.

[183] Saxena D, Singh A. K. A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center[J]. Neurocomputing, 2021, 426: 248-264.

[184] Moore J. D, Chase J. S, Ranganathan P, et al. Making scheduling” cool”: Temperature-aware workload placement in data centers.[C]. Proceedings of the USENIX annual technical conference, General Track, 2005, 61-75.

[185] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python[J]. the Journal of machine Learning research, 2011, 12: 2825-2830.

Research Results Obtained During the Study for Doctoral Degree

[1] **Khan T**, Tian W, Kadhim M. R, et al. A novel cluster ensemble based on a single clustering algorithm[C]. 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS), 2021, 127-135.

[2] **Khan T**, Tian W, Ilager S, et al. Workload forecasting and energy state estimation in cloud data centres: Ml-centric approach[J]. Future Generation Computer Systems, 2022, 128: 320-332.

[3] **Khan T**, Tian W, Zhou G, et al. Machine learning (ml)–centric resource management in cloud computing: A review and future directions[J]. Journal of Network and Computer Applications, 2022, 204: 1-17.