Optimizing the Resource Utilization in Cloud Computing Environment with Autoscaling using Machine Learning Methods

MSc Research Project
Cloud Computing

Sri Madhan Shettihalli Anandreddy
Student ID: x21230064

School of Computing
National College of Ireland

Supervisor: Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sri Madhan Shettihalli Anandreddy
Student ID:	x21230064
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Punit Gupta
Submission Due Date:	18/09/2023
Project Title:	Optimizing the Resource Utilization in Cloud Computing Environment with Autoscaling using Machine Learning Methods
Word Count:	6121
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sri Madhan Shettihalli Anandreddy
Date:	18th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

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

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

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

Optimizing the Resource Utilization in Cloud Computing Environment with Autoscaling using Machine Learning Methods

Sri Madhan Shettihalli Anandreddy
x21230064

Abstract

In the rapidly evolving cloud ecosystem, maintaining service levels while optimizing resource utilization is paramount. Autoscaling is a technique that can be used to dynamically adjust the resources allocated to an application to meet demand. This can help to improve performance, reliability, and cost-effectiveness. This research introduces a comprehensive study of various autoscaling algorithms applied in a simulated cloud environment. TPerformance of different algorithms for autoscaling including Moving Average, Random forest regressor (RF), Support vector regressor(SVR), Gated recurrent Unit (GRU) and Convolutional LSTM (ConvLSTM) were evaluated. Comprehensive metrics such as load vs. capacity, resource utilization, delayed load, and prediction errors were employed for evaluation. After performing various sets of experiment the most optimal algorithm for autoscaling has been identified. Notably, the random forest emerged as the top-performing algorithm, demonstrating proficiency in managing cloud resources effectively.

1 Introduction

In recent decades, there has been wide attention being empowered by cloud computing due to its extensive and on-demand applications as well as the dynamism identified with resource scalability. It is simultaneously perceived in the cloud environment that any application requires the potential for scaling virtual resources either up or down to achieve high performance as well as cost-effectiveness, wherein the enforcement of the "Service Level Agreement" (SLA) is acknowledged. However, ginning true elasticity, cost-effectiveness, and high-performance values can be considered challenging in terms of cloud applications and providers ensuring the service. With the continuous advocacy of cloud-based applications with the integration of virtual resources to enable fast and reliable network services to users, autoscaling has gained wide attention. According to the statement of Radhika and Sadasivam (2021), autoscaling can be identified as a reliable process that dynamically allocates resources and coordinates execution prerequisites. Meanwhile, with the increase in workload, there is an additional requirement of resources for the effective maintenance of performance within the domain of satisfactory SLAs. Despite the enhancement in cloud systems and autoscaling significance in resource allocation, the dynamism of the hybrid cloud environment requires improvement. It is in fact, this implication that retains an ever-lasting research need to escalate ideas and information for effective autoscaling of resources in the cloud computing environment.

Over the years, it is identifiable that there has been a tremendous increase in cloud applications despite relatable issues in resource management. Based on the evaluation of industrial experts and market researchers, the global presence of the cloud market has been estimated to be 545.8 billion dollars in 2022 and the projected growth in 5 years has been estimated to be 1240.9 billion dollars by 2027, thus predicting a CAGR rate of 17.9% respectively (Markets And Markets, 2022). Given this statistical data, it is clear that users with various criteria have heavily relied on cloud computing services because of the robustness that automation and agile infrastructure generate. In fact, claiming different activities supposedly performed in various organizations, IT experts present in these firms have simultaneously approached to introduce cloud services to ensure effective customer service delivery and experience enhancement, increased cost-effectiveness and ROI, and facilitated the remote work culture with cloud-based business engagement.

The increased spending considered to facilitate cloud-based services has enhanced large vendors to expand their business geographically and acquire a wide customer base with increased digital content and internet usage. Accordingly, planning a risk contingency and strain that may be experienced during various operations have been equally acknowledged as the potential source to drive the cloud market. Nevertheless, the opportunity served by cloud computing, the challenging aspect related to resource allocation in the hybrid cloud environment is an area of focus. Even though the service provided by the cloud is highly significant in recent times, a potential long-term relationship establishment between service providers and users depends on the effectiveness present within the service. Evidence-based information explained that cloud computing enables remote access to virtual services across any geographical region to end users. However, analysis and interpretation of "real-time" data have been identified to be a challenging aspect for cloud data analysts. It is even more difficult to determine the accurate amount of resources that are supposedly required for task execution. Moreover, with the presence of large data configuration, it is often observed that resources are underutilized which, therefore, reduces the cost-effective nature of the service. Contribution knowledge and focus on the current issue, researchers have vigorously spent their time and ideas in modeling and analyzing time-series data in different fields in cloud computing. Within the domain of minimizing costs and performing effective autoscaling methods to balance workload, different methods have been used in a simulated environment. Jananee and Nimala (2023) exclaimed that predictive analysis can help in reducing or eliminating losses like service unavailability, increased energy consumption, and end-user loss.

The overall understanding of the autoscaling process of the allocated resource, therefore, is contemplated as a challenging process that needs suitable approaches to be aligned efficiently in handling tasks and reducing challenges. The introduction of different algorithms has been considered to be a captivating research approach in the recent decade to introduce effective solutions in minimizing the above-identified challenges. Extensive surveying and related research works have produced enhanced information that can gear satisfying results although limitations can equally be perceived towards cloud end-user requirements. Over the years, various algorithms have been specified and exclaimed to be effective one over another by assessing different metrics such as overuse and underuse of cloud resources, delayed task counts, and accuracy. The proactive management of cloud computing resources provides both opportunities and challenging aspects to providers and users thus, requires manageable methods to be developed to overcome the challenges. It is already acknowledged that various methods including the development of algorithms have been introduced and implemented by experts to interfere in the complex multi-dimensional process of cloud resource handling. Therefore, this study has purposively drawn attention to identifying and implementing these algorithms using time series data. In fact, various related works have been developed by researchers for years to predict the scalability of the model. Bestowing information from different research works, it is perceivable that machine learning models, deep learning models, ensemble learning models, and some traditional methods are used and accounted for to gather insights on their relevance and accuracy.

1.1 Aim of the Study

The aim of this study is to comprehensively evaluate various autoscaling algorithms within a simulated cloud environment. By assessing the efficacy and performance of these algorithms, the goal is to identify the most suitable algorithm for dynamic cloud autoscaling management. Performance of moving average (Rule-based), random forest, SVR, GRU, and ConvLSTM will be evaluated in terms of load vs capacity, over and under-utilisation of resources, amount of delayed load, and error in model prediction. The results of the study will be discussed, along with their implications for the practical application of autoscaling algorithms.

1.2 Research Question

In the context of cloud computing, the ability to adapt to fluctuating workloads is crucial for efficiency and performance. Effective autoscaling requires algorithms that can predict and respond to dynamic load patterns, allocating resources precisely when needed. This study delves into the comparative evaluation of several autoscaling algorithms, seeking to address the following research question.

- Which autoscaling algorithm—be it moving average, random forest, SVR, GRU, or Conv LSTM—offers the most efficient and accurate response to dynamic load patterns in a simulated cloud environment?

The overview of the information contributed in this chapter can be acknowledged as informed aspects of the importance of cloud computing services and the dynamism contextualized in the autoscaling method of resources. The information is inferential and aided to research further to deduce in-depth insights that are explicit to understand how the autoscaling process works and what challenges can be experienced during resource allocation and effective task scheduling. Further, it is necessary to explore different methods and algorithms introduced in contemporary times to develop diversified solutions that are helpful enough to reduce or eliminate the challenges and essentially draw a comparison between the models to determine the rate of effectiveness under different parametric values.

2 Related Work

The specification of the second chapter contributes to proliferating information from published studies that have successively highlighted the significance of cloud computing services. Since the past decade, it has been identified that computational devices are upscaled with a rapidly escalating demand for cloud computing. This information, in fact, serves as a paradigm to identify how cloud computing has eased access to virtual resources and provided extensive network and software services to users. With the retention of responsiveness, scalability as well as efficiency, cloud computing offers high-performance computing services. However, challenges have been explored in the previous chapter while focusing on resource allocation and workload balancing. Thus, this chapter has provided information from the literature to identify the solutions demonstrated by experts to resolve the issues and essentially create dynamism in the effective resource allocation process.

2.1 Autoscaling Using Traditional Rule-based Methods

There has been extensive research effort given to implicate and understand how cloud computing service is embraced by organizations to offer a shift in their information technology (IT) operations. The review of empirical studies focused on the traditional method of predicting resource utility and optimization in clouds explained that the CAPEX model, often used by organizations to procure dedicated hardware certainly perceives a depreciable rate with the operational expenditure model (OPEX) Chouliaras and Sotiriadis (2022). This model shares cloud infrastructure and allows users to pay for needed resources without procuring any dedicated hardware over time. It is indeed a significant feature of cloud computing and offers appropriate resource utility by users upon paying fixed prices based on the pricing model. Despite the dynamism and effectiveness of cloud features, it has been acknowledged that with continuous increases in cloud resources demand, complexities in effective resource allocation and balancing workload become a challenging aspect. Biswas et al. (2017) introduce a hybrid model based on a reactive and proactive approach that can essentially address the issue by scaling resources on the basis of users' demand.

Another study presented by Chouliaras and Sotiriadis (2022) has described the significance of a threshold-based scaling model indicated as a "Performance-aware Auto-scaler for Cloud Elasticity" (PACE) that uses both the reactive and proactive methods. Dealing with extensive cloud resources and their optimization using this rule-based model retains a pre-defined threshold to avoid system failures. Accordingly, it ensures applications related to performance requirements. Overall, this method applies threshold-based scaling rules for avoiding failures at the time of workloads. The auto-scaling process using this method is indeed reliable, however, it is equally essential to explore whether the model is sufficient enough while calculating mean absolute error, load capacity value, and others. In this regard, the information is not sufficient in the literature, which, therefore, marked the need for other efficient solutions. The Queuing Model introduced by Srivastava and Kumar (2023) has exclaimed the effectiveness of the model in providing dynamic scalability as well as scaling of virtual resources while reducing economic costs and the threat of violating Service Level Agreements (SLAs). The work is performed in a simulated environment using Cloudsim. As per the experimental observation, better performance is achieved with this model compared to existing works.

2.2 Autoscaling Using Machine Learning Algorithms

Apart from the understanding of the application of different threshold-based scaling rule models in the auto-scaling process for resource optimization, contemporary approaches have persistently shaped the process with more comprehensive solutions to bridge existing gaps. In this regard, different machine learning models are introduced that provide extensive solutions in the auto-scaling process. Jananee and Nimala (2023) introduced the reinforcement learning model - an advanced machine learning algorithm that serves as great potential in the automatic resolution of decision-making problems under a complex and uncertain cloud environment. The model has appeared to provide a promising solution in the auto-scaling process wherein it provides effectiveness to learning transparency, dynamism, and adaptability to the resource management process and acquired policies to execute potential applications Jananee and Nimala (2023). Motivated by this and other works, researchers have further aligned information on the efficiency of other machine-learning models in the auto-scaling of cloud computing resources. The demand for predicting cloud computing resources has been considered a vital task because it helps in optimizing the resource management process. Nawrocki and Osypanka (2021) explained that nonetheless the active approach to cloud computing, and irresistible challenges can be experienced by users in urgent accurate predictions, which, therefore, increases economic costs. The author further explained that introducing a novel framework that can detect the anomaly and improve the auto-scaling process is essential. A model based on anomaly detection method and machine learning configuration is achieved to enhance a cost-effective strategy and minimize QoS constraints in resource management. Experiment results obtained from the study demonstrate that cost optimization and effective resource management retain an estimated accuracy of 51%-85% respectively.

Indeed, it has been identified that cloud computing has gained immense attention in business organizations to enable cost-effective virtual resource usage. It is, therefore, essential to enhance resource optimization which is otherwise considered to be an important issue. Osypanka and Nawrocki (2022) explained that various methods have been introduced over the years to promote the dynamic scalability of cloud resources however; the dependability on a single factor such as computing power has marked an unsatisfactory result with those approaches. Therefore, a particular focus on machine learning models has been appraised by researchers as cost-optimal and effective resource allocation solutions. Previously, some machine learning models and either individually or in combination have been explored to address the challenges. Osypanka and Nawrocki (2022) demonstrated a solution that is a combined novel approach based on anomaly detection, particle swarm optimization (PSO), and machine learning. The model is based on a hybrid approach which has aimed to provide a cost-effective solution for cloud computing resource optimization. Depending on the experimental result achieved from the test Microsoft cloud computing environment for 10 months, cost reduction is estimated to be 85% compared to any other existing models or methods.

It is essentially a motivating approach of experts to introduce novel methods and retain effective approaches one after another to increase cloud efficiency in the organization. However, given the increased complexity and enormous data-driven activities performed in recent times, it is understandable that challenges in resource management and increased workload are common scenarios in the cloud computing environment. The allocation of dynamic resources and auto-scaling process are important in cloud computing. Herein the proper prediction of workload and allocating resources accordingly need higher accurate prediction models. Sharifian and Barati (2019) in their study demonstrated a hybrid machine-learning prediction algorithm that addresses the identifiable issues. The hybrid model introduced in the focused study is based on support vector regression which is further tuned with a chaotic PSO algorithm. Based on the time-frequency scales obtained from the time series workload, the last detailed component achieved from the time series perceives a higher frequency which can create noise. Therefore, to scale down the effect, a "generalized autoregressive conditional heteroskedasticity" (GARCH) model is adopted for the prediction process. The assessment of the hybrid model has provided an experimental result wherein the mean absolute error is estimated to be 29.93%, 29.91% & 24.53% respectively. The result obtained herein shows that the identified model is effective in the prediction process although the accuracy improvement is much better than individual GARCH, ANN, and SVR algorithms.

2.3 Time Series Methods for Autoscaling

It is already aware that dynamic resource scales and optimization in cloud computing is a challenging process with increasing workload. With the increasing popularity of cloud platforms in business organizations, the issue has become more prominent. Thus, different contemporary approaches are developed to potentially address these challenges (Bhagavathiperumal, 2020). Apart from the significant machine learning-based resource optimization, time series methods such as the ARIMA model are also applied in the prediction of workload. Kumar, (2018) has demonstrated a number of contemporary approaches among which the amalgam of time series forecasting models like ARIMA & ETS has gained much importance due to their better performance in improving the prediction accuracy by 2.6% compared to the individual ARIMA model. It is an effective approach that ensures the relevance of time-series forecasting although future work is encouraged to be performed further on these forecasting models. Tran et al. (2018) in their study has demonstrated the performance of a multivariate fuzzy long-short-term-memory (MF-LSTM) model which uses multivariate time-series data to enhance the prediction process regarding cloud resource consumption. This study has partially demonstrated the significance and effectiveness of the time-series forecasting approach in auto-scaling and the tested result from Google traced data has proven its efficiency as well as feasibility in a more convincing manner.

2.4 Deep Learning methods for Auto scaling

At present, the cloud is an influential computing platform that has been deployed by many developers to enhance web applications in data centers. While aggravating information on many comprehensive models that can improve the automatic scaling process and balance the workload in data centers, challenges are increasingly persistent due to the complexity that exists in recent data-driven approaches. According to Nawrocki et al. (2023)the prediction of computing resource use in different systems ensures proper optimization of the resource management process. With the popularity gained by the cloud in recent times, it has become more urgent to accurately predict workload balancing techniques for dynamic resource scaling. The suitable approach to deep learning methods has gained attention in this regard among which some studies are empirically reviewed in this chapter. Dang-Quang and Yoo (2021)in their study discussed a bidirectional LSTM model (Bi-LSTM) in the prediction of HTTP workloads for the future. As per the observation of the experimental result, the Bi-LSTM model has achieved better performance accuracy than other neural architectures and state-of-the-art techniques. Moreover, it has offered a 530-600 times faster speed of prediction compared to ARIMA models regarding different workloads.

Another study conducted by Dogani et al. (2022)has demonstrated accurate prediction models that can efficiently optimize cloud resource allocation as well as avoid SLA violations. Herein, a CNN model is applied in the extraction of "hidden spatial features" from every correlated variable and finally, the extracted features are induced within the GRU network to optimize attention mechanisms for the extraction of temporal correlational features. According to the experimental result obtained, the DL method has improved the prediction accuracy by 2-28% compared to baseline techniques and existing research. Likewise, another study conducted by Ahamed et al. (2023)discussed different deep-learning models and neural networks and conduct an experimental analysis to determine the prediction accuracy of resource allocation in clouds. Depending on the performance evaluation the LSTM model has exhibited a better performance than other models. Despite all these illustrations from the literature, the research work partially contributed to the specification of performance accuracy of DL models in the prediction of workload management and resource allocation optimization. Therefore, more focus on this particular segment is needed to vividly produce in-depth insights into these algorithms in the prediction process.

2.5 Resource allocation and deallocation using Ensemble Methods

Apart from the consideration of various individual and hybrid models, a focus on ensemble learning methods has equally received enough attention due to its extensive application across different prediction and classification processes. In order to ensure increased demand for cloud computing services, cloud service providers have been configuring different methods that can optimize available resource utilization and workload management. Leka et al. (2023) in this regard have presented a discussion on a PSO-based ensemble learning method that enhances a workload forecasting advent by employing blended "ensemble learning strategies". Based on the predictability result, it can be explained that the model has performed better than most of the existing models and even compared to LSTM, GRU, and other sub-models.

3 Methodology

Autoscaling is essential to efficiently manage variable workloads, preventing system overloads and resource wastage. Without proper resource scaling, services can face disruptions, slow performance, and wasted resources. Our novel methodology addresses these challenges with a predictive approach. Our methodology consists of a client-side generating loads and a server-side equipped with an intelligent autoscaler, processing and balancing the demands seamlessly. as shown in Figure 1.

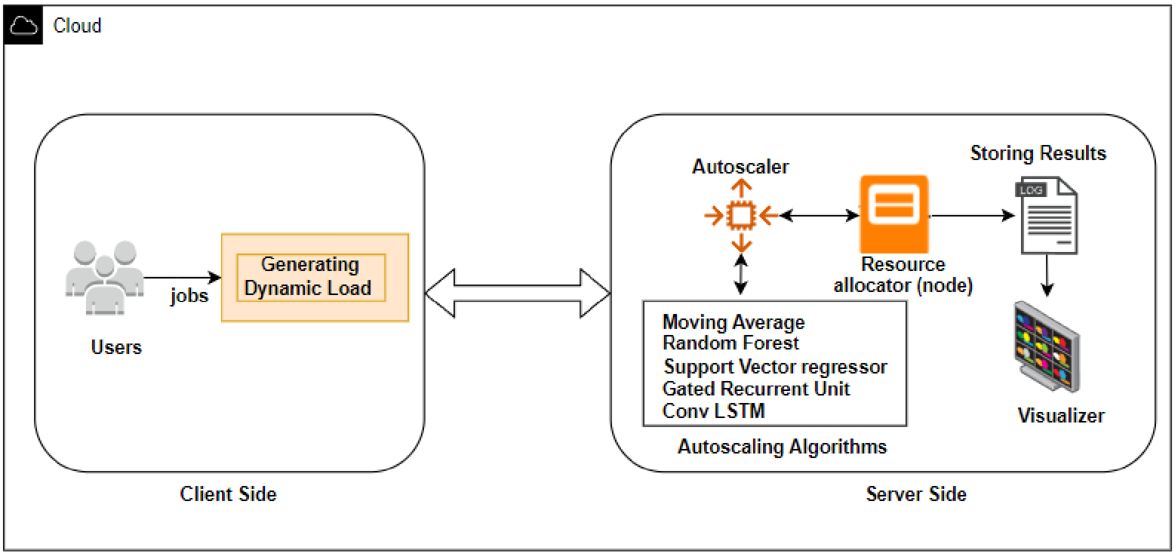

Figure 1: Methodology Diagram for Autoscaling

3.1 Client Side

The client side in cloud computing environment plays a pivotal role in generating and assigning loads to jobs, simulating real-world scenarios where applications might experience variable demand. This consists of a user base that generates jobs from various devices. This can be laptops, mobiles, tablets, or any other internet-enabled device. As per our architecture, the client side will generate a load with the combination of sine wave, cos wave and adding some random variance to it for defining the cycles. This will create a load that varies over time, which is more realistic and similar to the sinusoidal workload pattern in Real time cloud environment than a constant load. At each cycle, one or more than one jobs can be generated. This will create a variable workload on the server. The jobs with associated load will be assigned to a cloud server for processing. The cloud server will then scale up or down its resources based on the workload. Overall, it can be said that the client side is responsible for generating the load in the form of a job, which needs to be processed by the cloud server.

3.2 Server Side

The server side is the backbone of the autoscaling architecture, responsible for handling and processing incoming jobs from clients. It dynamically adjusts resources based on predictive analytics to meet fluctuating demands. Efficient management on this side ensures timely task execution, optimal resource utilization, and overall system robustness. On the server side, a variety of components are present, and each component is discussed in detail in further sections of paper.

3.2.1 Autoscaler

The autoscaler is the main and most crucial component of the architecture. Based on the outputs from the algorithms, the autoscaler predicts future workload. If the predicted load exceeds the current capacity, resources are scaled up; if it's less, they're scaled down. Vertical scaling is implemented in this work, where rather than adding more nodes (horizontal scaling), the capacity of an existing node is adjusted (either increased or decreased). This can be in terms of CPU, memory, or I/O capacity. The autoscaler, based on its predictions, decides on the scaling action. This dynamic adjustability is crucial for optimal resource utilization. To identify the most suitable autoscaling algorithm, five different algorithms were deployed within the autoscaler, Where the performance of each algorithm will be evaluated in our simulated environment and the best algorithm will be identified. The algorithms used for autoscaling in this research are Moving Average, Rand Forest Regressor (RF), Support Vector Regressor(SVR), Gated Recurrent Unit (GRU) and Convolutional Long short term memory (Conv-LSTM). Each autoscaling algorithm will be discussed in detail in the respective chapters. 4.

3.2.2 Node

A node is a virtual machine that is available on the server end. It has a certain amount of resources, such as CPU, memory, and storage. The autoscaler can adjust the capacity of the node by adding or removing resources. For example, if the load is increasing, the autoscaler might add more CPU or memory to the node. The node is responsible for processing the load that is sent to it by the client side. It does this by running the jobs that are sent to it. The node also maintains a history of the jobs that it has processed. This history can be used by the autoscaler to make predictions about the future load.

3.2.3 Logging and Performance Analysis

Logging acts as the cloud's memory. This component maintains records of events, operations, and errors related to the processing tasks handled by nodes. This information can be accessed by the performance analyzer to analyze the performance of the node and the autoscaling algorithms. On the other hand, the Performance analyzer is responsible for extracting the raw from logging and calculating the meaningful metrics and insights to understand and optimize the functioning of the autoscaling algorithms. The performance analyzer will also be responsible for creating interactive charts and graphs in order to understand the performance of autoscaling algorithms in our simulated environment in a more efficient way. The performance analyzer will be used to calculate metrics such as Load vs Capacity, resource overutilization and underutilization, delayed load amount, and mean error for each autoscaling algorithm.

4 Design Specification

This section will delve into the architectural details of each algorithm employed as an auto-scaling technique.

4.1 Moving Average

This algorithm analyzes the load generated by the last 10 cycle jobs and generates the average of it as capacity. This is a simple algorithm that is easy to understand and implement. However, it is not very accurate for predicting future capacity, especially if the load is fluctuating rapidly. However, it is an efficient approach for handling the linear load. The moving average can be calculated using the following formula as shown in Figure 4.1.

$$\mathrm{SMA} = \frac{A_1 + A_2 + ... + A_n}{n}$$

Where A_1, A_2 and A_n are the load generated by each job and 'n' represents the total number of jobs. The Moving Average algorithm can predict future loads. By analyzing the average load over the previous n cycles, the system can make informed decisions about scaling resources up or down.

4.2 Random Forest Regression (RF)

Random forest regressor is a supervised learning algorithm that uses ensemble learning to predict a continuous value. It works by creating multiple decision trees and then averaging their predictions. This helps to reduce overfitting and improve the accuracy of the predictions. It is more sophisticated approach, this algorithm provides predictions about future capacities. As an ensemble model, it uses multiple decision trees to enhance its accuracy. In this online algorithm setup, the model continually updates itself, taking in the history of the past 9 cycles to predict the 10th cycle's resource requirements. The architecture of random forest regressor is shown in Figure 2.

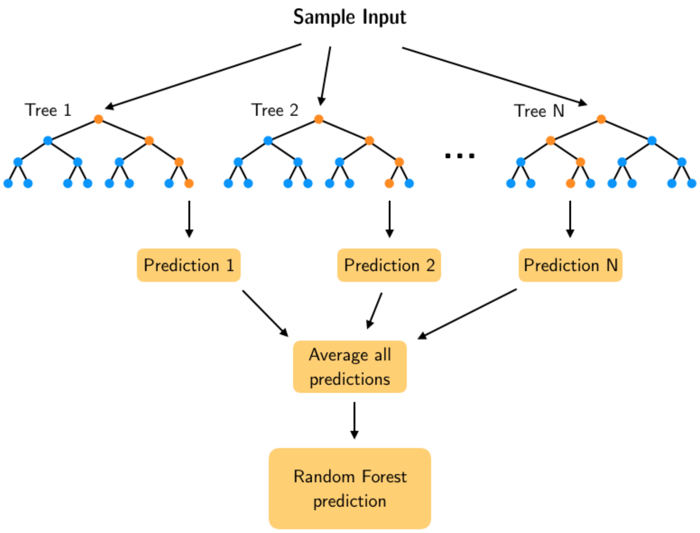

Figure 2: Random Forest Regression Architecture Nedjati-Gilani et al. (2017)

4.3 Support Vector Regressor (SVR)

Support vector regressor (SVR) is a supervised learning algorithm that uses the support vector machine (SVM) algorithm to predict a continuous value. SVR works by finding the best hyperplane that separates the data into two classes, with the margin being as wide as possible. It uses linear kernel to map the data into higher-dimensional space by minimizing the coefficient of regression function. The predicted value for a new data point is then the value of the hyperplane that is closest to the data point. An online Support Vector Regressor (SVR) algorithm is employed in accordance with our architecture. Similar to Random Forest Regression, it uses the data from 9 cycles to predict the 10th, ensuring that the model remains adaptive and responsive. The architecture of SVR is shown in Figure 3.

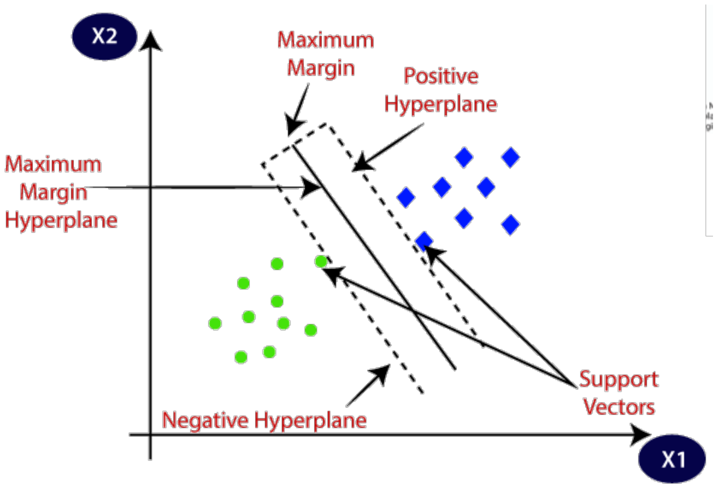

Figure 3: Support vector Regression Architecture Cardoso-Fernandes et al. (2020)

4.4 Gated Recurrent Unit (GRU)

A type of recurrent neural network optimized for sequence data and capture long terms dependencies in the data. This model is designed to address the vanishing gradient problem of traditional RNNs, making them more effective for longer sequences. GRUs have two gates: an update gate and a reset gate. The update gate controls how much of the previous state is used to update the current state. The reset gate controls how much of the previous state is forgotten. GRU is trained once on a large historical dataset, and it bases its predictions on that pre-existing knowledge. Architecture of GRU is shown in Figure 4.

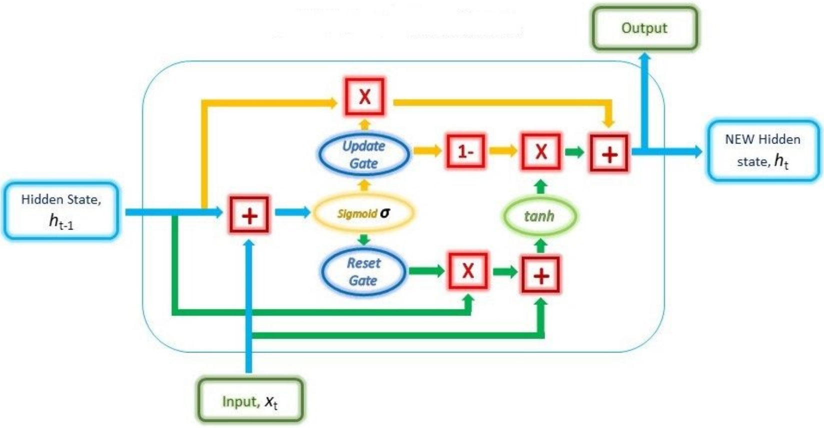

Figure 4: Gated Recurrent Unit Architecture Zhao et al. (2019)

4.5 Convolutional Long Short term Memory (Conv-LSTM)

A fusion of Convolutional Neural Networks (CNN) and LSTM, Conv-LSTM brings the spatial feature detection capability of CNNs and the temporal pattern detection of LSTM together. Conv-LSTM is a successful model which can even extract the features from images, videos, and audio. Similar to GRU, this model also will be trained once on large historical data and will be utilized for making future predictions. The architecture of Convolutional LSTM is shown in Figure 5.

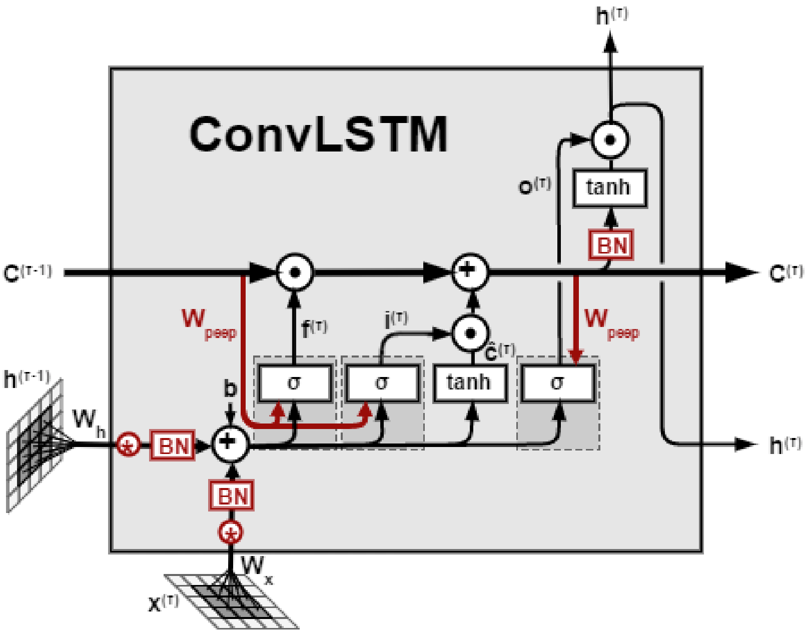

Figure 5: Convolutional LSTM Architecture Shi et al. (2022)

5 Implementation

Autoscaling is a crucial task in cloud computing architecture. An efficient autoscaling algorithm can prevent the resources from being under-utilized and over-utilized. In this research, a simulated environment was developed using Python on AWS Cloud. Where a certain set of Python libraries has been used to achieve our objectives. Other than that for implementing the autoscaler 5 different algorithms also has been implemented the algorithms are Moving Average, Random forest regressor, support vector regressor, Gated Recurrent unit, and Convolutional LSTM. Cloud sinusoidal workload pattern was chosen for implementation. The synthetic dataset is generated with the combination of a sine wave and cos wave where numpy, datetime, maths, and random python libraries have been used. To achieve concurrency and implement real-time processes in the project, the threading library was utilized. For implementing the machine learning-based autoscaling algorithmsSklearn and scikit-learn libraries were used and for implementing the deep neural network-based auto-scaling algorithms TensorFlow and keras libraries were used. For representing the log data with a graph and for implementing the performance analyzer, matplotlib library was used. Amazon S3 bucket is used to store the generated graphs. The boto3 Python library is used to programmatically interact with the S3 bucket and upload the graphs. Every function and library serves a special purpose in developing the overall project. The main objective of this research is to efficiently utilize the resources by minimizing the over-provisioning and under-provisioning of resources. The overall architecture of simulated environment has been developed by considering the real-world cloud computing infrastructure used for autoscaling. In this research, Vertical scaling is implemented where instead of adding more virtual machines, more resources such as CPU, ram, or storage are added to a single server itself. The system with the following configuration is required to execute our cloud computing platform for autoscaling.

Resource	**Configuration**
Operating System	Linux
Main Memory (RAM)	4GB
Number of CPU Cores	8 (Virtual Cores)
Storage	30GB
Programming Language	Python3
Python Libraries	Numpy, Pandas, Threading, Matplotlib, Sklearn, Keras, Tensorflow

Table 1: Configuration Requirement of System

6 Evaluation

Autoscaling is vital for maintaining high performance, cost efficiency, and service availability in dynamically changing environments. Therefore, choosing the optimal autoscaling algorithm is an essential step. In this section, the performance of each auto-scaling will be evaluated with different metrics such as Actual load and Predicted Capacity, Over and under-utilization of resources, Amount of delayed load, and error in model prediction. Each metric will help us to understand the algorithm mechanism and its performance. In the further subsection, the study will delve into a detailed discussion of each metric and perform a comparative analysis of the obtained results.

6.1 Experiment-1 / Load Vs Capacity

Load Represents the actual demand or workload placed on a system at any given time. Whereas, Capacity indicates the maximum workload a system can handle without degrading performance. When the load exceeds capacity, it often results in system slowdowns, outages, or failures. If the system's capacity far surpasses the current load, it indicates under-utilization. Maintaining excessive capacity can be costly and inefficient. Ideally, the load should be close to capacity, which ensures that resources are utilized in an efficient way. The algorithm which will have the minimal distance between the load and predicted capacity will be considered the most optimal algorithm. More the difference between load and capacity poor is the performance of the auto-scaling algorithm.

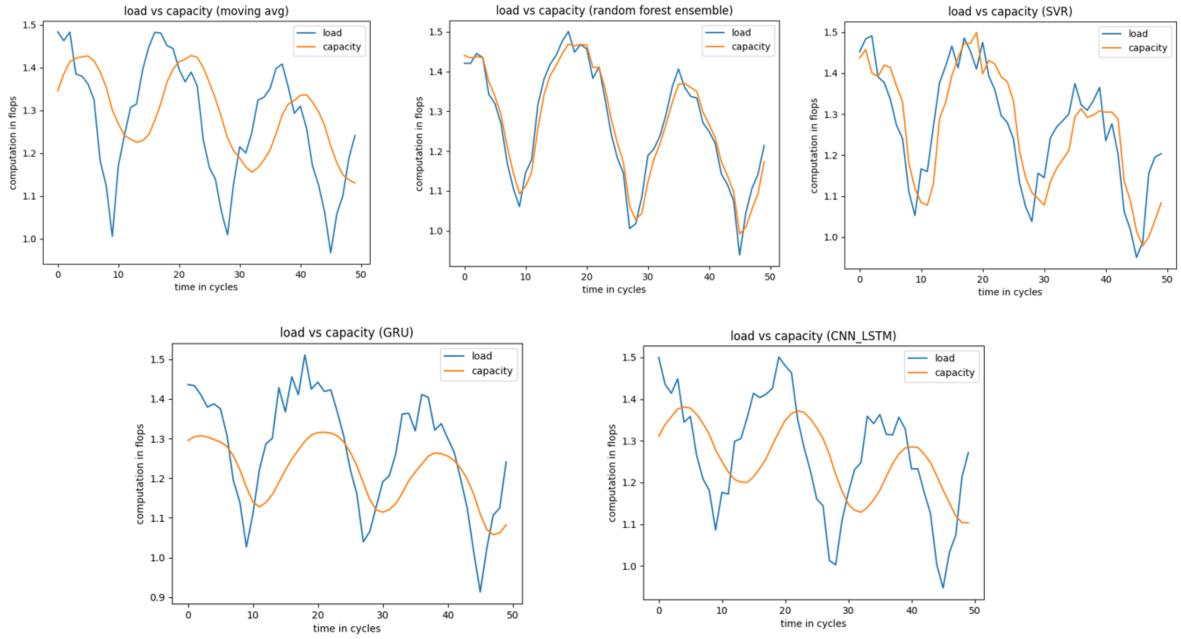

Figure 6: Actual Load vs Predicted Capacity

Figure 6, represents the load vs capacity of all the algorithms. After analyzing the results for each auto-scaling algorithm it has been observed that the Random forest regression algorithm generates the minimum difference between load and capacity, followed by the support vector regressor. Except for these autoscaling algorithms, the difference between load and capacity is quite high, which clearly indicates that resources are either over-provisioned or under-provisioned in the moving average, GRU, and Convolutional LSTM algorithm.

6.2 Experiment-2 / Over and Under Utilisation of Resources

Over-utilization occurs when the demand (load) on a system surpasses its available capacity or resources. Under-utilization occurs when the system's capacity or resources far surpass the actual load or demand. An optimal auto-scaling algorithm should minimize both the overutilization and underutilization of resources. In this experiment, Overuse and underuse of resources for each algorithm were calculated. Algorithms that achieve the more close values to zero represent that resources are minimally over or under-utilized. Figure 7 represents the Over and Underutilization of resources by each auto-scaling algorithm which is represented with the help of line graph.

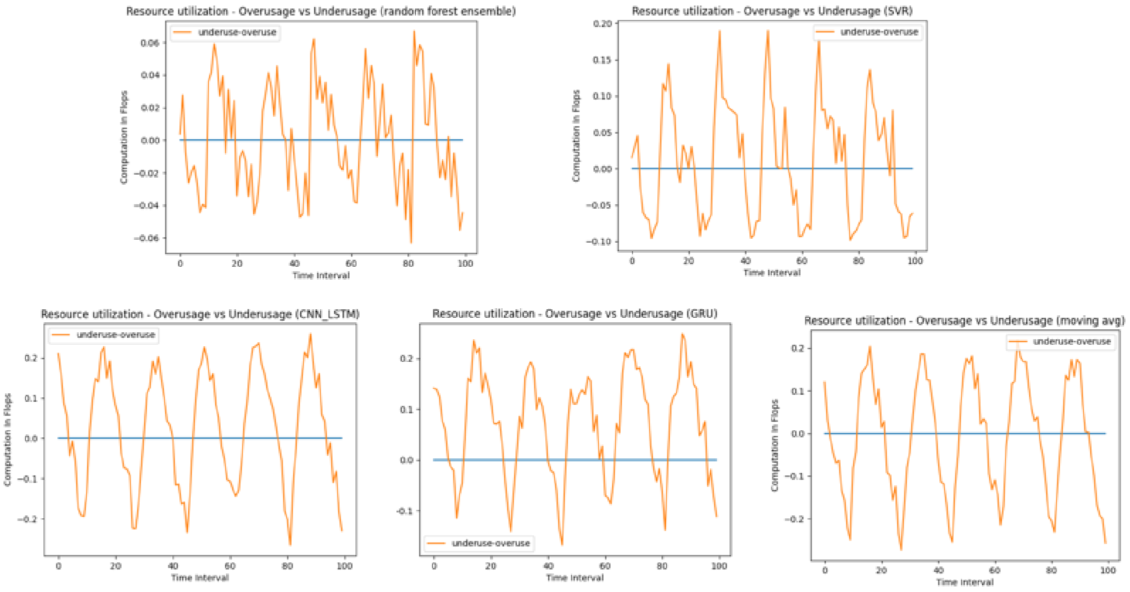

Figure 7: Over and Under Utilization of Resources

After carefully analyzing the obtained results from each algorithm as shown in Figure 7, it has been observed that the Random forest regressor has the minimum overutilization values of 0.06 and underutilization value of -0.04 which is quite less as compared to all the algorithms. Whereas, for the algorithms like moving average the highest number of over and utilization has been observed which is 0.2 and 0.3 which is quite high as compared to random forest regression. In terms of Over and Under-utilization of resources, SVR is the second highest performing model, which reduces the overutilization and underutilization of resources. Moving average not only can disrupt the cloud service also due to over-utilisation of resources the cost can be quite high for the customer, it is identified as the worst performing algorithm as compared to other autoscaling algorithms.

6.3 Experiment-3 / Amount of Delayed Load

The amount of Delayed Load is another important metric to compare the performance for auto-scaling algorithms. It reflects how much workload in a system is not processed immediately and gets delayed due to constraints or overloads. An increasing amount of delayed load can indicate deteriorating system performance and may result in system outage. slow response time and may increase the cost when resources are scaling up. Autoscaling algorithms with minimum delayed load will be considered as most optimal algorithm. Comparative analysis of delayed load is shown in Figure 8.

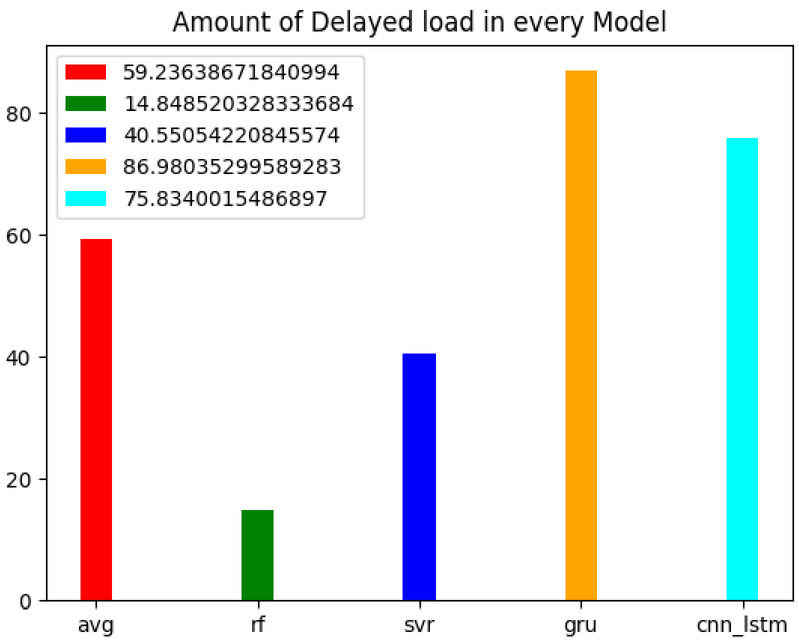

Figure 8: Amount of Delayed Load

Bar graphs shown in Figure 8 clearly represent the delay obtained from each algorithm. It has been identified that the highest delay has been obtained using GRU model, followed by Conv-LSTM, Moving average algorithm, SVR and then RF with the delayed load of 86.98, 75.83, 59.23, 40.55, and 14.84. From the results it is very clearly evident that with the random forest auto-scaling algorithm least amount of load has been delayed, Random forest is 5 times more efficient than other algorithms in terms of delayed load.

6.4 Experiment-4 / Error in Model Prediction

Auto-scaling relies on predictive models to anticipate load and optimally allocate resources. An error in these predictions can lead to either resource wastage (over-provisioning) or service disruptions (under-provisioning). The autoscaling algorithm with minimum error will be considered as a most optimal model which utilizes the resources efficiently. The Error Graph is represented with the help of bar graph in Figure 9.

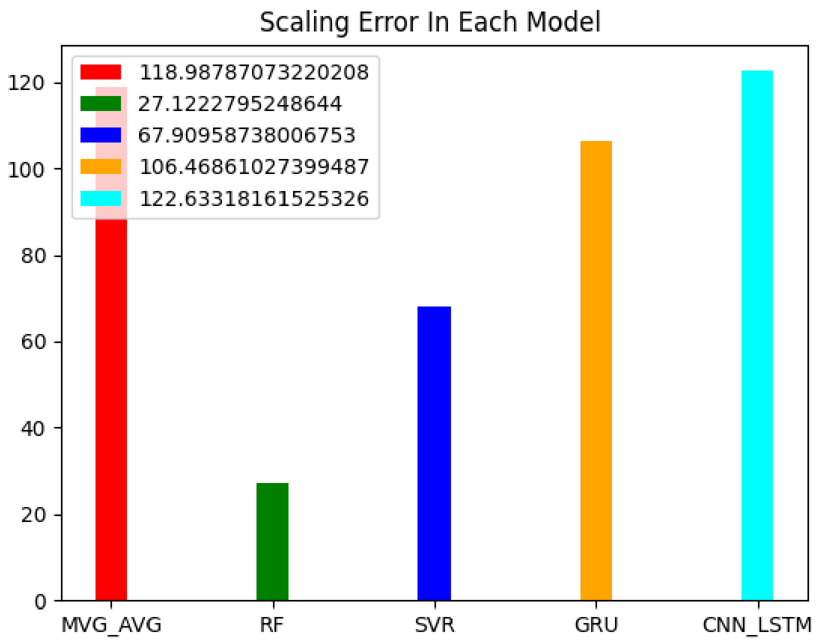

Figure 9: Error in Model Prediction

On analyzing the bar graph shown in Figure 9, it can be clearly seen that Random Forest is able to achieve the minimal error rate and the highest error rate has been obtained for CONV-LSTM and Moving average model. Also, SVR has performed better than GRU and is the second highest-performing autoscaling algorithm.

6.5 Discussion

Based on the above discussion, it is clear that random forest is the best-performing algorithm for autoscaling. Random forest exhibited the most balanced behavior across all metrics, indicating a robust understanding of the load patterns and resource allocation. Additionally, random forest is not as computationally expensive as deep learning models, such as GRU and ConvLSTM. SVR (Support Vector Regressor) also performed well but was not as effective as Random Forest. SVR's strength lies in its ability to handle non-linearities, but it may not capture temporal dependencies as effectively as time-series-based methods, hence falling short of Random Forest. Moving average is a simple model that does not take into account the underlying trends in the data. This has led to inaccurate predictions, especially during periods of high volatility, also it can not adjust the change in data and results in over or under-provisioning of resources. Both deep learning models, GRU (Gated Recurrent Unit) and Conv LSTM (Convolutional Long Short-Term Memory), were found to be among the least effective due to a lack of sufficient data. Thus it can be clearly said that, random forest's ensemble approach, efficiently balances complexity and expressiveness and can be considered as most optimal model for autoscaling.

7 Conclusion and Future Work

Autoscaling in cloud environments is crucial to efficiently handle varying loads, ensuring resource utilization optimization, cost-effectiveness, and consistent service performance. Proper autoscaling aids in maintaining SLAs and preventing service disruptions. In order to achieve the objective, a simulated environment was set up to compare several autoscaling algorithms. On the client side, a dynamic load was generated using a combination of sin and cos waves, and then assigned to the server's node. The server architecture comprised of an autoscaler and a node, with the autoscaler determining the node's capacity based on the predicted load. The server also included components for logging and performance analysis, essential for assessing the efficacy of the autoscaling. The evaluated algorithms were the moving average, random forest, SVR, GRU, and Conv LSTM. Among them, random forest stood out as the top performer across several metrics, including load vs. capacity balance, resource utilization, delayed load, and prediction error. SVR was the second-best performer, while GRU, Conv LSTM, and moving average were found to be less effective. In the future, instead of synthetic data the experiment can be performed over real cloud infrastructure data in order to enhance the accuracy of deep learning models. Other than that, the best properties of two or more models can be combined and an ensemble model can be developed like Conv-LSTM in this research for achieving the optimal results. The granularity of the logging system can be enhanced in order to capture micro-level behaviors and interactions.

References

Ahamed, Z., Khemakhem, M., Eassa, F., Alsolami, F. and Al-Ghamdi, A. S. A.-M. (2023). Technical study of deep learning in cloud computing for accurate workload prediction, *Electronics* **12**(3): 650.
URL: *https://doi.org/10.3390/electronics12030650*

Cardoso-Fernandes, J., Teodoro, A. C., Lima, A. and Roda-Robles, E. (2020). Semi-automatization of support vector machines to map lithium (li) bearing pegmatites, *Remote Sensing* **12**(14): 2319.
URL: *https://doi.org/10.3390/rs12142319*

Chouliaras, S. and Sotiriadis, S. (2022). Auto-scaling containerized cloud applications: A workload-driven approach, *Simulation Modelling Practice and Theory* **121**: 102654.
URL: *https://doi.org/10.1016/j.simpat.2022.102654*

Dang-Quang, N.-M. and Yoo, M. (2021). Deep learning-based autoscaling using bidirectional long short-term memory for kubernetes, *Applied Sciences* **11**(9): 3835.
URL: *https://doi.org/10.3390/app11093835*

Dogani, J., Khunjush, F., Mahmoudi, M. R. and Seydali, M. (2022). Multivariate workload and resource prediction in cloud computing using CNN and GRU by attention mechanism, *The Journal of Supercomputing* **79**(3): 3437–3470.
URL: *https://doi.org/10.1007/s11227-022-04782-z*

Jananee, M. and Nimala, K. (2023). Allocation of cloud resources based on prediction and performing auto-scaling of workload, *2023 International Conference on Artificial*

Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), IEEE.
URL: *https://doi.org/10.1109/iceconf57129.2023.10083865*

Leka, H. L., Fengli, Z., Kenea, A. T., Hundera, N. W., Tohye, T. G. and Tegene, A. T. (2023). PSO-based ensemble meta-learning approach for cloud virtual machine resource usage prediction, *Symmetry* **15**(3): 613.
URL: *https://doi.org/10.3390/sym15030613*

Nawrocki, P. and Osypanka, P. (2021). Cloud resource demand prediction using machine learning in the context of QoS parameters, *Journal of Grid Computing* **19**(2).
URL: *https://doi.org/10.1007/s10723-021-09561-3*

Nawrocki, P., Osypanka, P. and Posluszny, B. (2023). Data-driven adaptive prediction of cloud resource usage, *Journal of Grid Computing* **21**(1).
URL: *https://doi.org/10.1007/s10723-022-09641-y*

Nedjati-Gilani, G. L., Schneider, T., Hall, M. G., Cawley, N., Hill, I., Ciccarelli, O., Drobnjak, I., Wheeler-Kingshott, C. A. G. and Alexander, D. C. (2017). Machine learning based compartment models with permeability for white matter microstructure imaging, *NeuroImage* **150**: 119–135.
URL: *https://doi.org/10.1016/j.neuroimage.2017.02.013*

Osypanka, P. and Nawrocki, P. (2022). Resource usage cost optimization in cloud computing using machine learning, *IEEE Transactions on Cloud Computing* **10**(3): 2079–2089.
URL: *https://doi.org/10.1109/tcc.2020.3015769*

Radhika, E. and Sadasivam, G. S. (2021). A review on prediction based autoscaling techniques for heterogeneous applications in cloud environment, *Materials Today: Proceedings* **45**: 2793–2800.
URL: *https://doi.org/10.1016/j.matpr.2020.11.789*

Sharifian, S. and Barati, M. (2019). An ensemble multiscale wavelet-GARCH hybrid SVR algorithm for mobile cloud computing workload prediction, *International Journal of Machine Learning and Cybernetics* **10**(11): 3285–3300.
URL: *https://doi.org/10.1007/s13042-019-01017-1*

Shi, C., Zhang, Z., Zhang, W., Zhang, C. and Xu, Q. (2022). Learning multiscale temporal–spatial–spectral features via a multipath convolutional LSTM neural network for change detection with hyperspectral images, *IEEE Transactions on Geoscience and Remote Sensing* **60**: 1–16.
URL: *https://doi.org/10.1109/tgrs.2022.3176642*

Srivastava, A. and Kumar, N. (2023). Queueing model based dynamic scalability for containerized cloud, *International Journal of Advanced Computer Science and Applications* **14**(1).
URL: *https://doi.org/10.14569/ijacsa.2023.0140150*

Tran, N., Nguyen, T., Nguyen, B. M. and Nguyen, G. (2018). A multivariate fuzzy time series resource forecast model for clouds using LSTM and data correlation analysis, *Procedia Computer Science* **126**: 636–645.
URL: *https://doi.org/10.1016/j.procs.2018.07.298*

Zhao, H., Chen, Z., Jiang, H., Jing, W., Sun, L. and Feng, M. (2019). Evaluation of three deep learning models for early crop classification using sentinel-1a imagery time series—a case study in zhanjiang, china, *Remote Sensing* **11**(22): 2673.
URL: *https://doi.org/10.3390/rs11222673*