National College of Ireland

Optimizing Resource Allocation in Cloud Computing Using Machine Learning

MSc Research Project
MS in Cloud Computing

Pranay Kumar Chittipolu Giri Pratap
Student ID: x23201827

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……PRANAY KUMAR CHITTIPOU GIRIPRATAP……

Student ID: ……x23201827……

Programme: ……Ms in Cloud Computing…… **Year:** …2024….

Module: …… MSc Research Project……..……

Supervisor: …… Dr. Shivani Jaswal ……..……

Submission Due Date: ……24th April 2025……..……

Project Title: …… Optimizing Resource Allocation in Cloud Computing Using Machine Learning…….…….

Wordcount: 7181 **Page Count**……23……

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: ……Pranay Kumar Chittipolu Giri Pratap……

Date: ……24th April 2025……

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):	

Table of Contents

Contents

Optimizing Resource Allocation in Cloud Computing Using Machine Learning

Pranay Kumar Chittipolu Giri Pratap
Student ID: x23201827
Email id: -x23201827@student.ncirl.ie

National College of Ireland

Abstract

Cloud computing enables on-demand access to shared computing resources, offering scalability, flexibility, and cost-efficiency for modern applications. However, efficient resource allocation remains a persistent challenge, as fluctuating workloads often lead to either over-provisioning, resulting in wasted resources, or under-provisioning, causing performance degradation. Traditional resource management strategies, such as rule-based and threshold-driven autoscaling, lack the predictive intelligence required to anticipate future demand accurately. This study addresses these limitations by implementing a cloud-native, predictive resource allocation framework using deep learning models integrated within Amazon Web Services (AWS). The proposed workflow involves collecting and preprocessing time-series data from over 1500 virtual machines (sourced from the GWA-T-13 Materna dataset), storing it in Amazon S3, and training forecasting models on EC2 instances using the Cloud9 IDE. Two models were implemented: a standard BiLSTM and an enhanced Attention-Centric BiLSTM Fusion model. The results demonstrate the superiority of the attention-based model, which achieved a lower Mean Squared Error (0.0043) and Root Mean Squared Error (0.0656), compared to the standard BiLSTM. The study successfully showcases how predictive modeling, when embedded in a secure and scalable cloud environment, can significantly improve resource utilization and support intelligent, cost-effective cloud infrastructure management.

Keywords: Cloud Computing, Resource Allocation, Amazon Web Services (AWS), BiLSTM (Long Short-Term Memory), Attention Mechanism

1. Introduction

2.1 What is Resource Allocation in cloud computing

Resource allocation in cloud computing refers to the strategic process of distributing computing resources—such as CPU power, memory, storage, and network bandwidth Shukur et al. (2020) among various users, applications, and services operating within a virtualized cloud environment. The core purpose focuses on achieving top-notch performance and scalability through resource adjustments that depend on current usage

patterns. Cloud computing stands different from standard computing since it provides automatic resource scaling capabilities to expand or reduce resources Catillo et al. (2023). Service Level Agreements (SLAs) get met alongside operational cost reduction while providers dodge between over-provisioning and under-provisioning through efficient resource allocation. The platform contains four major functionalities including autoscaling and load balancing as well as virtualization and scheduling algorithms. The established methods distribute workloads evenly between multiple virtual machines to avoid resource under- and over-utilization. The incorporation of predictive analytics enhances the forecasting capability of machine learning techniques for making predictive allocation decisions. Cloud-based applications receive added CPU and memory resources when they encounter traffic growth which enables subsequent shutdown of idle system components to minimize energy usage and expenses. Advanced resource allocation technology becomes essential as cloud systems grow sophisticated since it guarantees optimum system performance together with reliability and cost-effectiveness.

2.2 Aim of the study

This study aims to develop a cloud-native solution that increases the performance of resource allocation within cloud computing environments through the use of predictive forecasting techniques. By using the scalability and flexibility of Amazon Web Services (AWS) the study has been focused on designing and deploying a system capable of analyzing virtual machine (VM) performance data to forecast future resource demands. The main aim is to address the limitations of traditional reactive resource provisioning methods which mainly lead to either underutilization or unnecessary over-provisioning, both of which increase operational costs and reduce system performance. This study integrates advanced deep learning models, specifically BiLSTM and Attention-Centric BiLSTM, within the AWS ecosystem using services like EC2, S3, and Cloud9 to simulate a real-world cloud infrastructure. The ultimate goal is not just to build accurate forecasting models but to evaluate their effectiveness when embedded in a cloud environment for dynamic, data-driven decision-making. The outcomes of this study are intended to support cloud architects and service providers in implementing proactive resource management strategies, ensuring better performance, cost optimization, and scalability across dynamic workloads in distributed cloud systems.

2.3 Motivation for Research

With the rapid adoption of cloud computing across industries some powerful resource management has become an important challenge. Unpredictable workloads mainly lead to either underutilization or overprovisioning of cloud resources which impacts performance and operational costs. This study is motivated by the need to increase workload forecasting techniques to enable dynamic and intelligent resource allocation in cloud environments. By using advanced deep learning models mainly Attention-Centric BiLSTM architectures the study aims to improve prediction accuracy and support real-time optimization.

2.4 Research Questions

What impact does predictive resource utilisation forecasting have on optimizing cloud infrastructure performance, and how can deep learning models like BiLSTM and Attention-based BiLSTM be integrated within AWS services to automate resource provisioning, and why is this important for reducing operational costs and preventing resource underutilisation in dynamic cloud environments?

2.5 Objectives of the Research

The research questions for this report are:

1. To analyze the impact of integrating predictive forecasting models on cloud infrastructure performance by measuring improvements in resource utilization efficiency, provisioning accuracy, and overall system responsiveness in a simulated AWS environment.
2. To design and implement a scalable cloud-native architecture using AWS services (EC2, S3, Cloud9) for hosting and executing predictive analytics pipelines focused on virtual machine resource utilization.
3. To integrate and evaluate deep learning models, specifically BiLSTM and Attention-Centric BiLSTM, within the cloud environment for forecasting key resource metrics using time-series VM trace data.

2.6 Outline of the Report

This report is structured into the following sections:

1. **Introduction:** This chapter introduces the concept of resource allocation in cloud computing, defines the problem, states the aim and motivation of the study, and outlines the research question and objectives.
2. **Literature Review:** This reviews existing methods of resource allocation and load balancing in cloud environments, including heuristic, metaheuristic, and machine learning-based approaches. Highlights research gaps that this study addresses.
3. **Research Methodology:** Describes the dataset used (GWA-T-13 Materna), AWS cloud components (EC2, S3, Cloud9), and preprocessing techniques including feature extraction, visualization, normalization, and rolling window implementation.
4. **Design Specification:** Justifies the cloud-centric architecture and illustrates the proposed workflow for predictive resource allocation using deep learning models on AWS.
5. **Implementation:** Details the cloud infrastructure setup, including configuration and use of EC2 instances, Cloud9 IDE, and S3 buckets for data storage and processing.

6. **Evaluation:** This will presents experimental results from BiLSTM and Attention-Centric BiLSTM models. Compares model performance and highlights improvements over previous studies.
7. **Conclusion and Future Work:** Summarizes key findings, reflects on the benefits of predictive cloud resource management, and proposes enhancements including AWS SageMaker integration, real-time scaling, and Docker/Kubernetes deployment.

2. Literature Review

2.1 Traditional Resource Allocation Approaches

Traditional resource allocation approaches in cloud and workflow scheduling have relied on heuristic and meta-heuristic methods to resolve the complexity of NP-hard scheduling problems. Below are some studies that have used heuristic-based methods and hybrid genetic algorithms as their different techniques. The methods pursued multiple goals which included reducing scheduling duration and cutting down costs while promoting performance advancements. The strategies of task duplication and task prioritization along with local search and dimensionality reduction have yielded positive results in scientific workflows alongside multi-dimensional resource scheduling according to these approaches. The ongoing improvements fail to solve scalability problems or adaptability limitations and computational overhead issues in such systems hence making real-time cloud environments demand more dynamic intelligent solutions.

The competency-based workforce allocation model proposed by Shojaei et al. (2024) optimizes construction project scheduling by addressing human resources allocation to project activities effectively. The research develops a three-objective mathematical planning model to achieve three goals which include project time reduction and workforce competency enhancement alongside cost reduction. Experts combined with decision-making methods helped identify and rank essential competency requirements through organized decision-making processes. A combination of GAMS software along with Non-Dominated Sorting Genetic Algorithm II (NSGA-II) with Multi-Objective Particle Swarm Optimization (MOPSO) algorithms forms the proposed approach. The approach uses exact and heuristic methods to solve sample problems and develop two meta-heuristic algorithms through Taguchi design techniques. The experiments show that NSGA-II and MOPSO respectively handle workforce allocation problems and NSGA-II achieves better results than MOPSO particularly when dealing with large-scale problems. The main drawback stems from using expert opinion to select criteria while also facing challenges due to high solution processing requirements of the proposed algorithms in time-sensitive project environments. The research shows that integrating competency-based methods with advanced optimization tools produces valuable outcomes for construction project management systems.

Another study given by Khanak et al. (2021) who have built a heuristic-based framework for Scientific Workflow Scheduling (SWFS) in cloud environments to optimize both scheduling results and cost expenditures. The research tackles the significant problem of finding near-optimal scheduling solutions in polynomial time because SWFAs require extensive data processing along with computational resources. The proposed model consists of three sequential stages which involve scientific workflow application and targeted computational environment and cost optimization criteria. The model testing occurred on the SIPHT workflow including three dataset size variations under three heuristic approaches that included single-based GA, PSO, and IWO together with hybrid-based HIWO and hyper-based DHHA. The model delivered its best results regarding job time completion and cost expenditure for both small and big datasets and security-based algorithms produced superior performance metrics than other options when working with medium-sized datasets. A possible drawback to this method involves the way heuristic strategies perform under varying dataset conditions because no one solution achieves best results for every case. Real-world cloud dynamics cannot be sufficiently evaluated through the use of simulation-based evaluation techniques. The research provides a functional cost-optimized workflow scheduling system available for scientific cloud implementations.

The Hybrid Heuristic-Based List Scheduling (HH-LiSch) algorithm which was proposed by Shirvani and Talouki (2021) strives to optimize makespan performance in heterogeneous cloud computing (HCC) environments by providing efficient dependent task scheduling for different virtual machines (VMs). The study tackles the NP-Hard problem of providing optimal assignment of dependent tasks across heterogeneous systems because there is no known polynomial-time solution for this problem. This method develops new scheduling processes and includes virtual machine slot selection through insertion-based methods and a time reduction mechanism based on task cloning. The combined operation of scheduling system components produces efficiency. The algorithm underwent testing through simulations of six actual scientific workflows on an RTG platform using makespan as well as Schedule Length Ratio (SLR) and speedup and efficiency performance metrics for evaluation. HHLiSch achieves better scheduling efficiency and shorter makespan durations than existing approaches in the results analysis. The model produces dependent execution outcomes through implementation however its performance relies on workflow complexity and resource variability and there is no testing of its effectiveness under real-world cloud conditions. The proposed scheduling system brings crucial improvements to HCC workflow management because it enhances task positioning during a decreased execution period.

Another study workflows operating across different cloud environments received a heuristic-based scheduling solution from Talouki et al. (2022) to minimize makespan durations. This study addresses NP-Hard scheduling problems between VMs which handle tasks with dependencies while having different configurations and performance characteristics across heterogeneous platforms that also restrain task dependencies. The proposed

scheduling method incorporates HEFT-based duplication and OCTd and OCTu task scheduling organizations to boost the list scheduling mechanism. The system updates work toward better execution performance by matching tasks with suitable VMs while minimizing unproductive downtime periods. The proposed model achieved validation using four workflows including Molecular, LU-Like, FFT and Montage that exceeded existing methods when measuring performance based on speedup and makespan and Schedule Length Ratio and efficiency. In spite of optimal outcomes the algorithm features two main limitations caused by its dependence on exact cost calculations and inflexible task distribution rules that limit adaptability to dynamic cloud conditions. The study delivers an efficient workflow scheduling approach which shows promise to enhance the operations of heterogeneous clouds.

At last Zhou et al. (2023) developed GHW as a new solution based on Growable Genetic Algorithm with Heuristic-based Local Search and Random multi-Weights to resolve the Multi-Dimensional Resource Scheduling Problem (MDRSP) within cloud computing environments. Traditional genetic algorithms gain through the "growth stage" addition which creates adaptive evolution while the integration of heuristic-based local search improves solution quality. The GHW algorithm receives enhancement through GHW-NSGA II and GHW-MOEA/D hybrid models that integrate population regeneration elements and sorting schemes from NSGA II and MOEA/D base algorithms. The testing procedures involved simulation datasets combined with AzureTraceforPacking2020 to achieve minimized maximum resource utilization across dimensions along with decreased total energy consumption. The proposed approaches produced superior performance than standard NSGA II and MOEA/D according to results which demonstrates their practical applicability for real-world cloud scheduling operations. Nevertheless the algorithm performs better than before other limitations emerge because of its computational overhead in the growth mechanism and its capability to manage very large and dynamic cloud environments could be restricted. The investigation establishes an effective approach for flexible and efficient multi-objective scheduling applications in managing cloud infrastructure.

2.2 Machine Learning in Cloud Resource Allocation

A hybrid optimization strategy for virtual machine allocation which targets power reduction and resource optimization with load balancing serves as a solution for cloud data centers according to Kumar et al. (2021). A combined Genetic Algorithm (GA) and Random Forest (RF) model enters the discussion because researchers notice increasing cloud services demand with energy expansion difficulties for data centers. Training data that emerged from the GA enables the trained supervised RF model to predict suitable resource allocation strategies. PlanetLab workload traces serve as the basis to demonstrate practical use of the proposed model. The hybrid method using GA-RF delivers better performance because it reduces power consumption while decreasing execution time and boosting resource utilization. GA-RF adoption presents difficulties for performance because it runs two algorithms and makes scalability determination complex for big dynamic

cloud computing environments. The research presents an intelligent approach to manage VMs efficiently which helps save energy in contemporary cloud infrastructure systems.

TrustFusionNet serves as a cutting-edge framework in cloud computing that uses Random Forest together with Convolutional Neural Networks (CNNs) to evaluate cloud resource trustworthiness according to Bharthi et al. (2025). An investigation has started to tackle basic issues related to safe and secure cloud resource management through assessments that combine historical data and behavioral patterns together with resource attributes. Several components of trust computing become possible through Random Forest implementations due to its multi-factor analysis capabilities yet CNNs automate trust feature discovery beyond traditional assessment methods. The combination between system stability analysis and evaluation across performance and trust assurance and resource use characteristics demonstrated TrustFusionNet delivers better outcomes compared to existing methods in trust assurances and resource optimizations. The model exhibits two key challenges that result from its need for high-quality labeled data and long-term costs stemming from maintaining dual-model system components. TrustFusionNet develops an establish framework of dependable cloud resource management by linking clear machine learning models with deep learning features to construct superior decision systems.

Prediction-enabled feedback Control with Reinforcement learning based resource Allocation (PCRA) is a current method for cloud resource distribution according to Chen et al. (2020) that optimizes QoS and reduces total cost. Real-world installations show that traditional methods using static workload measurements plus expert rules and repeated repetitions are inadequate because they produce poor adaptability and elevate operational expenses. PCRA implements Q-value prediction modeling and reinforcement learning architecture with multiple forecasters establishing various assessments for system states. The system generates accurate rapid answers through decision processes that benefit from additional enhancements brought by the resource allocation mechanism based on feedback control. Lab tests on the RUBiS benchmark achieved 93.7% precise management operation selection with results exceeding traditional and ML-based management technologies by 10–13% and 5–7% respectively based on overall type of performance evaluation. When implemented with holistic cloud environments or extensive setups the method demonstrates restrictions regarding extending to diverse large-scale systems which requires parameter adjustments for peak operational results. The PCRA method demonstrates potential as a real-time system for intelligent resource management of cloud services with high performance.

A metaheuristic-based scheme called Data Format Classification using Support Vector Machine (DFC-SVM) aims to solve the complicated issue of load balancing in cloud-oriented IoT environments as presented by Junaid et al. (2021). The influx of IoT applications generates large and various data volumes which exceed the capabilities of traditional load balancing approaches that use few parameters to achieve suboptimized resource distribution. The proposed method executes pre-classification of IoT raw data which includes audio along with video and text and images before storing information through a Support Vector Machine (SVM)

model that operates with one-to-many classification. The data classification process groups data categories through a modified Particle Swarm Optimization (PSO) algorithm that enables efficient virtual machine assignment of the organized data categories. The joint implementation of SVM-PSO as a framework proves successful in improving load balancing while making significant improvements to VM distribution optimization. The experimental investigation verifies that DFC-SVM obtains a classification precision rate of 94% while decreasing power usage by 11.82% and decreasing response duration by 16% as well as cutting SLA violations by 16.08% relative to existing reference methods. The classification process undertaken offline with this approach creates obstacles for real-time dynamic workload execution because it impacts the system's capability to adapt to fast-changing environments. The proposed method implements an effective system that boosts resource efficiency alongside service quality in cloud-based IoT systems.

Similarly Kumar and Ahmad (2022) have presented a time-efficient multi-class classification-based radio resource management (RRM) method for Cloud-Radio Access Networks (C-RAN) that uses Support Vector Machine (SVM) evolutionary cooperation to optimize spectrum resource distribution between macrocellular users (MUEs), remote-head users (RUEs) and Device-to-Device (D2D) communication pairs. The research investigates sub-channel reuse in 5G networks by developing an efficient solution that ensures top-quality service delivery in densified heterogeneous network environments. The NILP model structures the problem by allowing both cellular and D2D nodes to use sub-channels jointly. Next the proposed system uses a cooperative evolution SVM-based classifier to establish the ideal multi-class sub-channel assignments. The researcher evaluated the method using data generated from a 5G experimental prototype built on Open Air Interface (OAI) which accurately simulated realistic network environments. The proposed method achieves superior outcomes than traditional schemes because it delivers improved network throughput with higher prediction accuracy and better overall system utilization results. The implementation of NILP modeling combined with SVM training for multi-class scenarios introduces scalability challenges in next-generation ultra-dense 5G or beyond-5G networks where real-time deployment optimization needs to be done for better scalability. The research work identifies promising elements for developing adaptive RRM techniques in next-generation wireless networks.

Kamble et al. (2023) introduced the evaluation of deep learning models including CNN, LSTM and Transformer through a comparative study for cloud resource allocation forecasting. The main purpose of this research is to enhance cloud resource administration through accurate resource demand forecasting that improves scalability together with performance and reduced operational costs. The research tackles the important issue of inadequately reactive resource allocation practices because they result in either excessive resource allocation or resource deficiencies. The required evaluation depends on historical workload data from the Google Cluster Data (GCD) database to assess deep learning algorithm efficiency for pattern detection in complex cases. The evaluation demonstrated that LSTM accomplished 0.15 RMSE results because of its

advanced prediction power alongside the Transformer's adaptive strength for managing prolonged dependencies. Predictive analytics enables the development of proactive cloud resource management systems that deliver operational superiority through its investigative findings.

At last Kaim et al. (2023) proposed a deep learning-based prediction framework aimed at addressing the cloud resource autoscaling which is a key challenge in cloud computing to workload fluctuations. The study's primary goal was to improve resource allocation performance by workload patterns using a proactive model. To achieve this the authors has been designed an ensemble architecture combining an attention mechanism (AM) with bidirectional long short-term memory (BiLSTM) and convolutional neural networks (CNN). This hybrid model captures both temporal dependencies and local patterns in the workload data. The framework was experimentally validated using real-time cloud workload traces by showing good improvements with an 8.93% reduction in RMSE and an 11.43% reduction in MAE over existing models. While the results shows the performance of the proposed method which is a potential limitation is the model’s trust on high-quality, real-time data and computational complexity which may affect scalability and deployment in highly dynamic cloud environments.

Table 1: Comparison Table

Study	Technology Used	Strength	Weakness	Results
Kumar et al., 2021	Genetic Algorithm (GA) + Random Forest (RF)	Efficient VM allocation, reduced power consumption, improved resource utilization	Dual-algorithm complexity, scalability challenges in dynamic environments	Better performance in power reduction, execution time, and resource usage
Bharthi et al., 2025	TrustFusionNet (Random Forest + CNN)	Combines trustworthiness evaluation with deep learning-based feature extraction	Requires high-quality labeled data, costly dual-model maintenance	Outperforms existing methods in trust assurance and resource optimization
Chen et al., 2020	Reinforcement Learning (PCRA with Q-value prediction + feedback control)	High QoS, cost-effective, real-time decision-making	Limited scalability for large-scale systems without parameter tuning	Achieved 93.7% accuracy, surpassed traditional and ML-based approaches by 10–13% and 5–7%
Junaid et al., 2021	SVM with Particle Swarm Optimization (DFC-SVM)	High classification accuracy, reduced power use, improved load balancing	Offline classification limits real-time adaptability	94% classification accuracy, 11.82% lower power use, 16.08% fewer SLA violations
Kumar & Ahmad, 2022	SVM-based Evolutionary Classifier for C-RAN	Effective multi-class classification for spectrum reuse in 5G	Scalability issues in ultra-dense next-gen networks	Improved throughput and accuracy in 5G sub-channel allocation

Kamble et al., 2023	CNN, LSTM, Transformer	Accurate forecasting, reduces operational cost, handles long dependencies	High resource demand for training complex models	LSTM achieved 0.15 RMSE; Transformer showed adaptive strength in long-sequence prediction
Kaim et al., 2023	Attention Mechanism + BiLSTM + CNN (Ensemble DL Model)	Proactive resource autoscaling, captures both temporal and spatial patterns	High dependence on real-time data, computationally complex for dynamic deployment	RMSE reduced by 8.93%, MAE reduced by 11.43% compared to existing models

3. Research Methodology

3.1 Data Collection

The dataset used in this study has been sourced from the GWA-T-13 Materna dataset, which provides good time-series performance metrics collected from a distributed data center operated by Materna. It is a leading European IT service provider. It comprises three separate traces—Materna-trace-1, Materna-trace-2, and Materna-trace-3 where each representing the activity of 520, 527 and 547 virtual machines (VMs) respectively. Collected at 5-minute intervals between January 4, 2016, and February 8, 2016 the dataset records key resource usage parameters critical for cloud infrastructure analysis. Each observation includes timestamp, number of CPU cores, provisioned CPU capacity (MHz), actual CPU usage (MHz and %), memory requested and used (in KB and %), disk write throughput (KB/s), total disk size (GB) and network throughput (received and transmitted in KB/s). This high-resolution resource usage information provides a real-world basis for developing predictive models that can forecast workload demand and use cloud resource allocation by making it especially suitable for research in cloud performance, scalability and intelligent provisioning strategies. The dataset's scale and granularity also allow for realistic simulation of cloud operations and workload behaviors.

3.2 Cloud Components

There are three cloud components which have been used in this study:

- AWS Cloud9 IDE is a cloud-based development environment that allows developers to write, run and debug code using just a browser. It supports multiple programming languages and includes a code editor, terminal and debugger. Cloud9 gives a smooth development experience with no need for local setup and allows real-time collaboration between users.
- EC2 Node refers to a virtual server in Amazon's Elastic Compute Cloud (EC2) used to run applications on the AWS cloud. Each EC2 instance gives scalable computing capacity and can be configured with

specific CPU, memory and storage by making it good for hosting websites, applications or development environments.

- S3 Bucket is a storage container within Amazon Simple Storage Service (S3) used to store and get any amount of data like files, images or backups. Each bucket has a unique name and provides features like versioning, lifecycle policies, and access control. S3 is highly durable and scalable by making it good for storing static content, big data and backup solutions.

3.3 Data Preprocessing & Feature Extraction

In this study data preprocessing and feature extraction were important steps to prepare the raw Materna dataset for accurate forecasting within a cloud environment. Initially the dataset was imported from Amazon S3 into the EC2 instance with the help of Boto3 library which is been followed by parsing of the semicolon-tab delimited data. The "Timestamp" column which represents milliseconds was converted into standard datetime format using Pandas for increased temporal analysis. This conversion enabled the extraction of granular temporal features like Hour, Minute, Day, Month and Year which are very important for identifying patterns and trends over time. These features were added as new columns in the DataFrame using commands and similarly for other components. Also irrelevant or redundant columns were removed to improve processing performance. Missing or corrupted values were handled using interpolation and forward-fill techniques to maintain data continuity. The resulting structured dataset provided a richer temporal context, enabling both visual exploration and input preparation for time-series forecasting models. These preprocessing steps ensured that the models received clean, normalized, and temporally rich input data, laying the foundation for accurate and cloud-scalable prediction of VM resource usage.

3.4 Data Visualization

Figure 1 presents a pie chart showing the average CPU usage (measured in MHz) distributed across each hour of the day. Derived from grouped mean values of CPU usage for every hour, the visualization offers insights into system load patterns over a 24-hour cycle. The chart segments each hour by its corresponding CPU utilization percentage, highlighting a fairly uniform distribution throughout the day. The variation across hours remains minimal, with most hourly values ranging between 3.8% to 4.4%, suggesting a consistently balanced workload. A few hours, such as 16 and 0, show slightly higher usage peaks (around 4.39% and 4.36%), whereas hour 2 shows the lowest at approximately 3.82%, implying lighter computational demand during this period. This even spread may reflect automated system processes or background services running with little fluctuation irrespective of the time, potentially indicating server or data center environments operating around the clock. The chart, generated using Plotly Express, was saved as an image and uploaded to AWS S3 for storage and further use. The use of a pie chart mainly focuses on proportional differences and secures a good understanding of hourly resource consumption patterns for performance monitoring or optimization tasks.

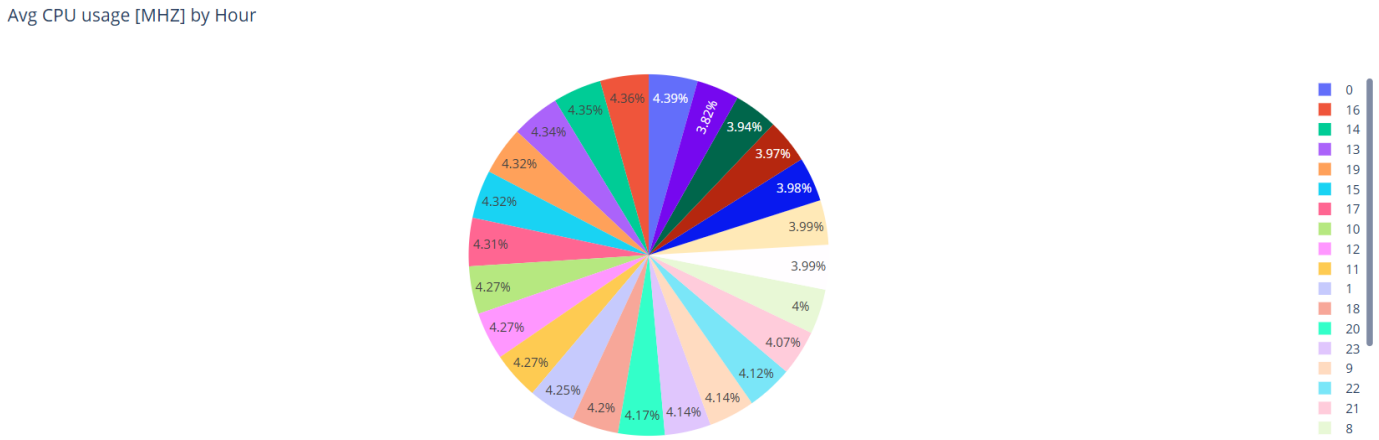

Figure 1: Avg CPU Usage (MhZ) by Hour

Figure 2 is a stacked bar chart shows the average CPU usage (measured in MHz) across different hours of the day for each day of the month. Each bar shows one day with its height showing the total average CPU usage for that day. The color gradient, ranging from deep purple (early hours) to bright yellow (late hours), distinguishes CPU usage by hour by allowing for a detailed type of examination of hourly contributions to daily usage. There are some days like 12th, 27th, and 28th show higher overall CPU activity which peaks close to or above 10,000 MHz by suggesting elevated workloads or processing demands on those dates. In contrast there are some days like the 2nd and 19th shows relatively lower usage by potentially showing off-peak periods.

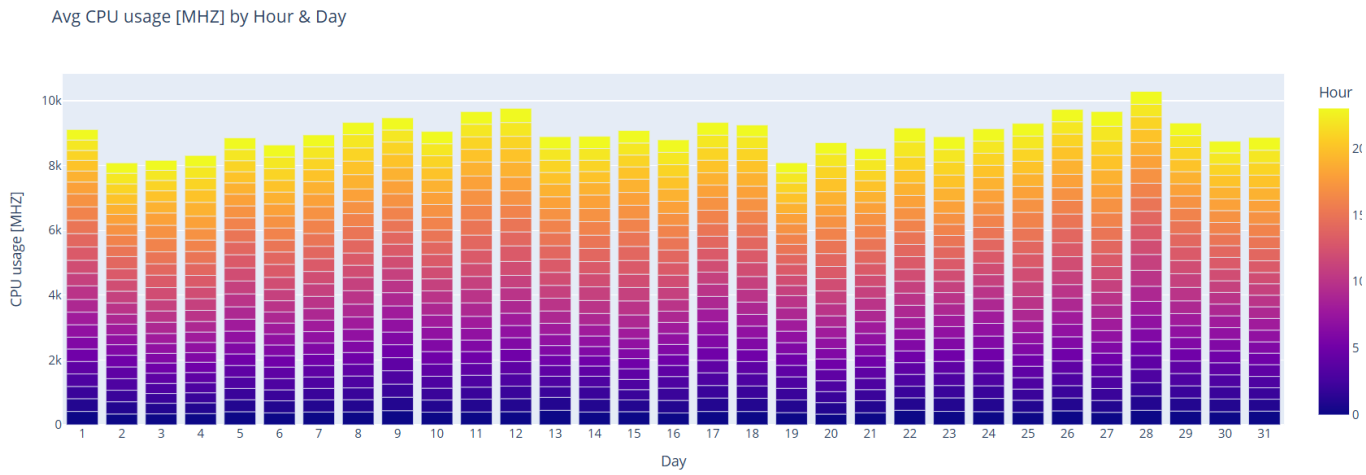

Figure 2: Avg CPU Usage (MhZ) by Hour & Day

3.5 Data Scaling

Data scaling using Min-Max Normalization is a very important preprocessing step implemented to improve model performance by standardizing feature values within a specific range which typically between 0 and 1. In this study the final cleaned dataset was normalized using the MinMaxScaler from Scikit-learn.

3.6 Window Rolling with Timestamp

Window rolling with timestamp is a very important technique in time-series forecasting used to convert sequential data into supervised learning format. In this study there is a window size of 12 which was applied that means the model uses the previous 12 time steps (or 1 hour of data, considering 5-minute intervals) to predict the next value in the sequence. The window_rolling() function has been achieved this by iterating over the dataset and extracting overlapping sub-sequences of 12 consecutive data points as input features (dataX) and the immediate next value as the target (dataY). This sliding window approach captures temporal type of dependencies and enables the model to learn patterns from historical resource usage which is very important for accurate forecasting in cloud environments.

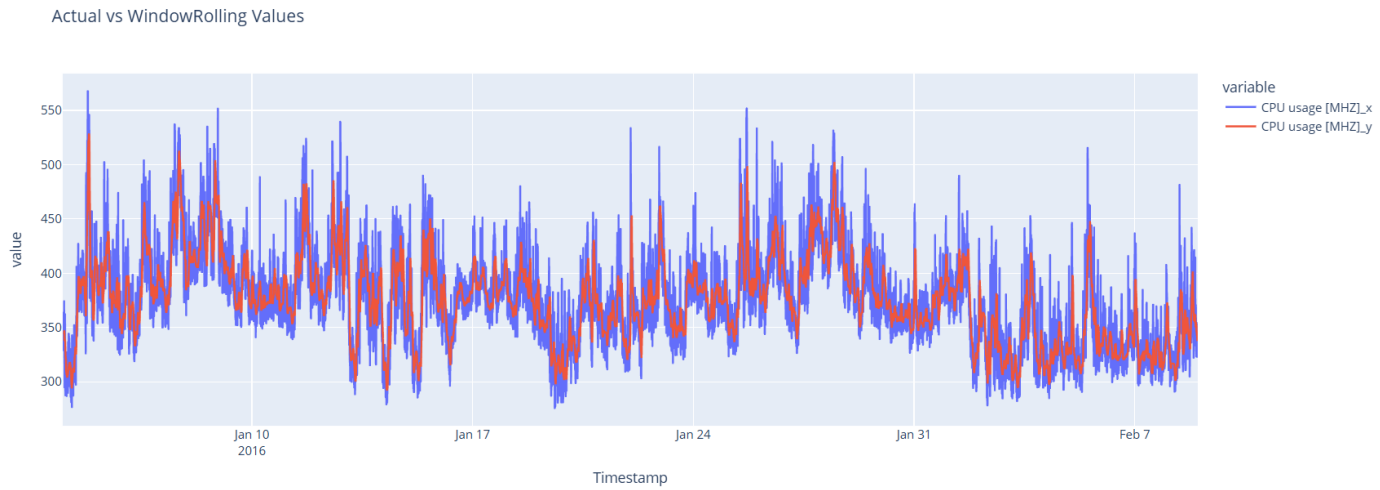

Figure 3: Actual vs Window Rolling Values

Figure 3 presents a time series line chart comparing actual CPU usage (in MHz) against smoothed values obtained through a window rolling average technique by spanning from early January to early February 2016.

4. Design Specification

4.1 Justification for Cloud-Centric Approach

A cloud-centric approach was chosen for this study due to its inherent scalability, flexibility and resource performance which are very important for handling large-scale, time-series datasets like the GWA-T-13 Materna traces. AWS services such as EC2, S3, and Cloud9 provided a robust, secure, and cost-effective environment for real-time data processing, storage, and model execution without the limitations of local hardware. The ability to dynamically allocate resources, automate workflows, and integrate with other cloud-native tools ensured a seamless, end-to-end pipeline for resource forecasting and optimization, aligning perfectly with the project's goal of enhancing cloud infrastructure performance and decision-making.

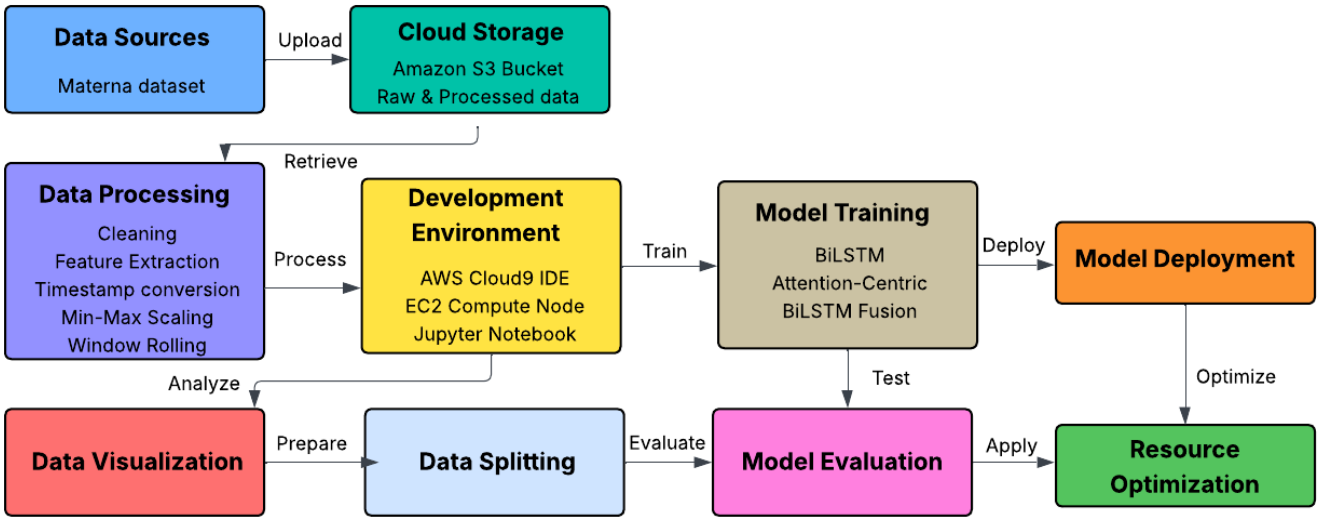

Figure 4: Proposed Workflow Diagram

Figure 4 is showing proposed workflow diagram of this study. The figure shows the complete workflow architecture of the proposed cloud-based workload prediction and optimization system. It begins with the Materna dataset as the primary data source which is been uploaded to Amazon S3 Bucket for cloud storage by maintaining both raw and processed data. The stored data is been retrieved and sent for data processing which does includes cleaning, feature extraction, timestamp conversion, min-max scaling and window rolling to prepare it for modeling. The processed data is then handled within the development environment hosted on AWS Cloud9 IDE, EC2 compute node and Jupyter Notebook. Model training includes BiLSTM and Attention-Centric BiLSTM Fusion models which have been designed for accurate forecasting of workloads.

5. Implementation

5.1 Cloud Infrastructure Setup

5.1.1 Amazon EC2 Instance

The AWS Cloud9 IDE, hosted on an EC2 instance, served as the central development environment for the project, providing a fully-featured, browser-based code editor directly integrated with AWS services. This environment eliminated the need for complex local setup and allowed for seamless collaboration, code execution, and debugging in the cloud. Running on a dedicated EC2 instance, Cloud9 offered native support for Python, Jupyter notebooks, AWS CLI, and Git, making it ideal for data processing, model training, and interacting with cloud resources. The integrated terminal and pre-installed development tools accelerated project setup, while persistent storage ensured that scripts, logs, and notebooks remained accessible across sessions. With IAM role-based access control, Cloud9 securely communicated with S3 buckets and other

AWS resources, enabling end-to-end data pipeline development within a single interface. Additionally, as Cloud9 was deployed within a Virtual Private Cloud (VPC), network security and access restrictions were enforced to prevent unauthorized access. The real-time coding and testing capabilities of Cloud9 significantly enhanced productivity and flexibility during model development and evaluation, making it a vital component of the cloud infrastructure for this project.

5.1.2 Amazon S3 Bucket

The EC2 Node functioned as the primary compute engine for running data-heavy operations and training deep learning models. Launched with a customizable instance type (such as t3.large), it provided scalable virtual hardware resources necessary for high-performance processing of the large-scale Materna dataset. This node executed the core logic of the project, including parsing large volumes of time-series VM trace files, transforming data into supervised learning format using sliding window techniques, and training BiLSTM and attention-based models. The instance was configured with additional swap space to handle memory-intensive operations and was equipped with a Python virtual environment for dependency isolation. Crucially, the EC2 node was securely connected to the rest of the AWS infrastructure using IAM roles and security groups, which ensured authorized access and protected it from unwanted intrusion. Logging, error tracking, and runtime performance metrics were captured and monitored using AWS CloudWatch to ensure smooth and consistent operations. The flexibility to scale the instance vertically or horizontally allowed for experimentation with different model complexities and training durations without compromising performance, making EC2 an essential component in deploying and executing the project efficiently in the cloud.

5.1.3 AWS Cloud9 IDE

The Amazon S3 bucket served as the central data repository for the project, offering highly durable, scalable, and cost-effective storage for both raw and processed datasets. Acting as the backbone of the data pipeline, the S3 bucket securely hosted the GWA-T-13 Materna trace files uploaded from local storage and made them readily accessible to EC2 and Cloud9 environments through IAM role-based access. This seamless integration allowed Python scripts running on EC2 to fetch input files directly from S3, perform processing, and then store cleaned data, trained model weights, and prediction outputs back into designated folders within the bucket. The bucket was configured with versioning enabled to track file changes and prevent accidental loss of data, while server-side encryption (SSE-S3) ensured that all stored files were encrypted at rest. To enforce access control, bucket policies were implemented to allow read/write permissions only to specific trusted AWS services. Multipart upload support facilitated smooth handling of large CSV files, ensuring reliability even under network disruptions. The use of S3 eliminated the need for traditional file servers, offering a cloud-native, always-available data storage solution that significantly enhanced the project's efficiency, security, and scalability.

6. Evaluation

6.1 Experiment 1: BiLSTM Model

Figure 5 presents a time-series line graph comparing actual versus predicted CPU usage (in MHz) from February 2 to February 9, 2016, using a BiLSTM (Bidirectional Long Short-Term Memory) model for forecasting. The blue line represents the real CPU usage values, while the red line shows the predicted values generated by the model. The data is sampled at 5-minute intervals, and early entries include real values such as 323.36 MHz at 20:15 on February 1 and 312.83 MHz at 20:20, with corresponding predictions of 347.01 MHz and 341.87 MHz respectively. These early examples illustrate a moderate prediction overshoot, common in initial time steps as the model stabilizes. Throughout the week, both curves exhibit a consistent pattern, capturing periodic spikes and drops in CPU load—particularly visible around February 5 and February 8, where values exceed 500 MHz. The model closely tracks the cyclical nature and temporal variability of the actual data, showing that the BiLSTM is effective in modeling non-linear sequences with time-dependent patterns. The figure was plotted using Plotly Express and saved as cpuusageforecatsing_loadbalancing1.png, then uploaded to an S3 bucket for remote access. This visualization confirms the model's strong predictive capability in dynamic cloud environments for load balancing applications.

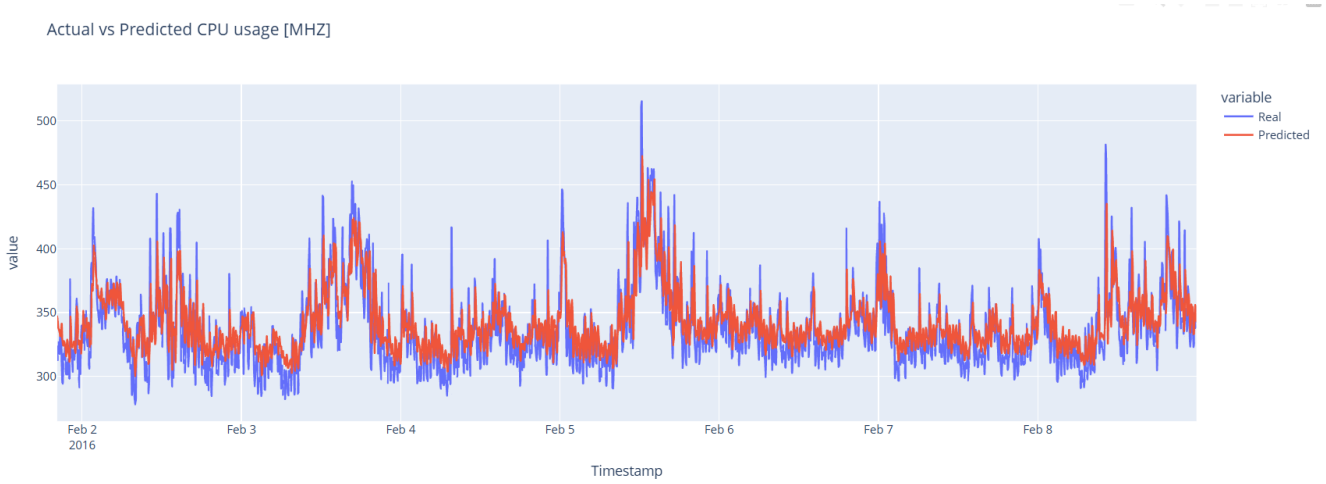

Figure 5: Actual vs Predicted CPU Usage (MhZ)

6.2 Experiment 2: Attention-Centric BiLSTM Fusion Model

Figure 6 illustrates the performance of an Attention-Centric BiLSTM Fusion Model in forecasting CPU usage (in MHz) over a week-long interval from February 2 to February 9, 2016. The blue curve denotes actual CPU utilization, while the red line represents the model's predictions. The graph is based on data collected at 5-minute intervals and plotted using Plotly Express. The model exhibits strong temporal tracking, aligning closely with real usage patterns, including periodic peaks and troughs. Notably, CPU usage spikes above 500 MHz on February 5 and February 8, where the model successfully captures the surge, indicating its robustness in modeling short-term variability and complex time-dependent behavior. Initial predictions such as at 20:15 and 20:20 on February 1 show real values of 323.36 MHz and 312.83 MHz, respectively, with corresponding predictions of 347.33 MHz and 347.75 MHz, slightly overestimating actual usage. As the timeline progresses, the model adjusts and mirrors fluctuations with impressive accuracy, even during dense oscillations and usage valleys around February 4 and February 6. This performance demonstrates the effectiveness of combining BiLSTM with attention mechanisms to enhance sequence learning and contextual memory, making it highly suitable for intelligent load balancing strategies in dynamic cloud computing environments. The graph was saved as cpuusageforecatsing_loadbalancing2.png and uploaded to an S3 bucket.

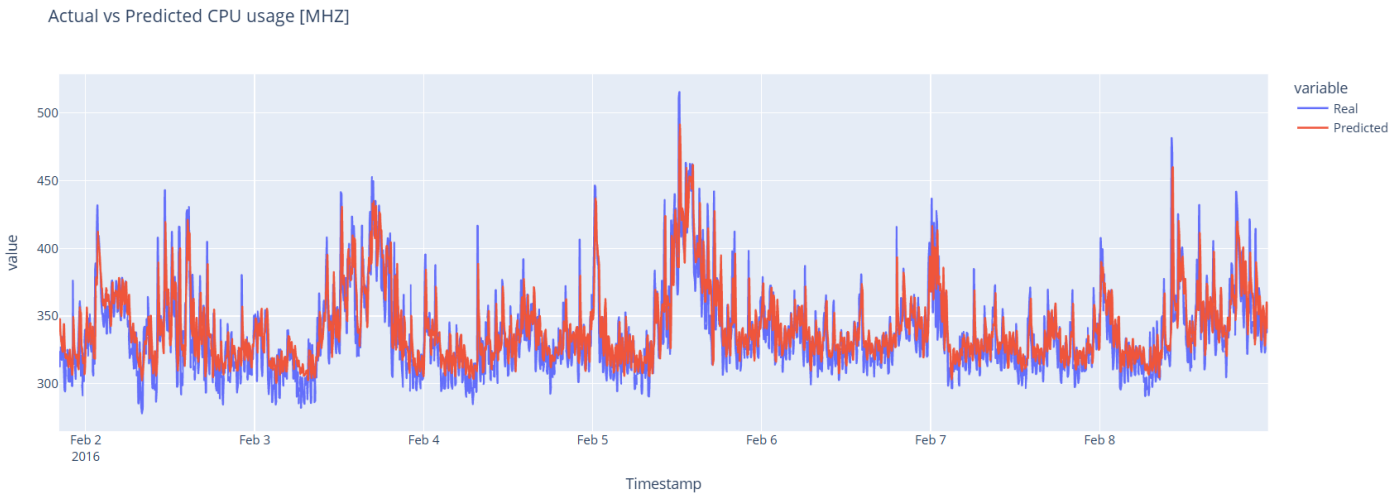

Figure 6: Actual vs Predicted CPU Usage (MhZ)

Table 2: Model Performance

Metric	BiLSTM Model	Attention-Centric BiLSTM Fusion Model
Mean Squared Error (MSE)	0.004404578569906176	0.004309236404342292
Root Mean Squared Error (RMSE)	0.06636699910276324	0.0656447743871688

6.3 Comparison of Model Performance: Previous Study vs. Current Study

The current study has outperformed the previous work by (Zhu et al., 2019) by achieving a much lower RMSE of 0.0656 compared to 6.903. This improvement is attributed to the use of a BiLSTM architecture fused with a lightweight attention mechanism by allowing for better temporal learning without incurring high computational costs. Also the model was implemented and optimized within a real cloud environment (AWS) by using practical scalability and integration unlike Zhu's study, which lacks deployment specifics. The rolling window strategy also helped maintain forecasting accuracy across time-series sequences while reducing complexity. This study is also mentioned in the literature review chapter.

Table 3: Comparative Performance of Workload Prediction Models — (Zhu et al., 2019) vs. Current Study

Aspect	(Zhu et al., 2019)	Current Study (Attention-Centric BiLSTM Fusion) (Best)
Model Used	LSTM Encoder-Decoder with Attention	Attention-Centric BiLSTM Fusion
Application Domain	Workload prediction in cloud-end clusters	Resource utilization forecasting in cloud infrastructure
RMSE Achieved	6.903	**0.0656**
Prediction Strategy	Scroll-based multi-step forecasting	Time-series forecasting using rolling window strategy
Scalability Challenges	High due to heavy attention and multi-step computation	Moderate, optimized for AWS EC2-based deployment
Cloud Integration	Limited	Fully integrated with AWS (EC2, S3, Cloud9)

7. Conclusion and Future Works

This study has been successfully showed the potential of leveraging intelligent forecasting models within a cloud-native environment to optimize resource allocation in cloud computing. By deploying a predictive analytics pipeline using AWS services such as EC2, Cloud9, and S3, the study not only maintained a strong cloud focus but also integrated machine learning models as supporting tools for decision-making. Two deep learning models—BiLSTM and an enhanced Attention-Centric BiLSTM Fusion Model—were implemented to forecast key resource usage metrics like CPU and memory consumption based on real-world time-series data from over 1500 virtual machines. The results showed that the attention-based model marginally outperformed the standard BiLSTM in terms of both MSE and RMSE, validating the effectiveness of attention mechanisms in improving model accuracy for cloud workload prediction.

From a cloud perspective, the architecture ensured scalability, modularity, and security, all while using cost-effective infrastructure. Cloud services were not only utilized for storage and compute but also played a vital role in automating tasks, handling data pipelines, and monitoring performance. The modular implementation using AWS components demonstrates how predictive intelligence can be seamlessly embedded into real-world cloud operations.

Looking ahead, several enhancements can be incorporated to elevate this system further. Firstly, integrating AWS SageMaker would allow for managed training, tuning, and deployment of models in a production-grade MLOps workflow. Additionally, real-time data ingestion from live VM monitoring tools like CloudWatch Logs and AWS Kinesis could enable dynamic, real-time scaling. The system could also be extended to trigger autoscaling actions automatically via Lambda functions based on predictive thresholds. Lastly, containerizing the entire pipeline using Docker and orchestrating with Kubernetes would facilitate multi-cloud deployment and horizontal scaling. These future directions not only promise improved performance but also align the solution with modern, serverless, and edge-ready cloud architectures.

References

1. Kumar T, S., Mustapha, S.D.S., Gupta, P. and Tripathi, R.P., 2021. Hybrid approach for resource allocation in cloud infrastructure using random forest and genetic algorithm. *Scientific Programming*, *2021*.
2. Bharathi, S.T., Balasubramanian, C. and Shanmugapriya, S., 2025. Enhancing Cloud Resource Allocation with TrustFusionNet Using Random Forests and Convolutional Neural Networks. *Tehnički vjesnik*, *32*(1), pp.116-122.
3. Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X. and Rong, C., 2020. Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Transactions on Cloud Computing*, *10*(2), pp.1117-1129.
4. Junaid, M., Sohail, A., Turjman, F.A. and Ali, R., 2021. Agile support vector machine for energy-efficient resource allocation in IoT-oriented cloud using PSO. *ACM Transactions on Internet Technology (TOIT)*, *22*(1), pp.1-35.
5. Kumar, N. and Ahmad, A., 2022. Cooperative evolution of support vector machine empowered knowledge-based radio resource management for 5G C-RAN. *Ad Hoc Networks*, *136*, p.102960.
6. Shojaei, B., Naserabadi, H.D. and Amiri, M.J.T., 2024. Optimizing Competency-Based Human Resource Allocation in Construction Project Scheduling: A Multi-Objective Meta-Heuristic Approach. *Qubahan Academic Journal*, *4*(3), pp.861-881.
7. Al-Khanak, E.N., Lee, S.P., Khan, S.U.R., Behboodian, N., Khalaf, O.I., Verbraeck, A. and van Lint, H., 2021. A heuristics-based cost model for scientific workflow scheduling in cloud. *Computers, Materials & Continua*, *67*(3), pp.3265-3282.

8. Shirvani, M.H. and Talouki, R.N., 2021. A novel hybrid heuristic-based list scheduling algorithm in heterogeneous cloud computing environment for makespan optimization. *Parallel Computing*, *108*, p.102828.
9. NoorianTalouki, R., Shirvani, M.H. and Motameni, H., 2022. A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), pp.4902-4913.
10. Zhou, G., Tian, W., Buyya, R. and Wu, K., 2023. Growable genetic algorithm with heuristic-based local search for multi-dimensional resources scheduling of cloud computing. *Applied Soft Computing*, *136*, p.110027.
11. Kamble, T., Deokar, S., Wadne, V.S., Gadekar, D.P., Vanjari, H.B. and Mange, P., 2023. Predictive Resource Allocation Strategies for Cloud Computing Environments Using Machine Learning. *Journal of Electrical Systems*, *19*(2).
12. Kaim, A., Singh, S. and Patel, Y.S., 2023, January. Ensemble cnn attention-based bilstm deep learning architecture for multivariate cloud workload prediction. In *Proceedings of the 24th International Conference on Distributed Computing and Networking* (pp. 342-348).
13. Shukur, H., Zeebaree, S., Zebari, R., Zeebaree, D., Ahmed, O. and Salih, A., 2020. Cloud computing virtualization of resources allocation for distributed systems. *Journal of Applied Science and Technology Trends*, *1*(2), pp.98-105.
14. Catillo, M., Villano, U. and Rak, M., 2023. A survey on auto-scaling: how to exploit cloud elasticity. *International Journal of Grid and Utility Computing*, *14*(1), pp.37-50.