Prediction of Resource Utilization in Cloud Computing using Machine Learning

MSc Research Project
Cloud Computing

Ruksar Shaikh
Student ID: x22174711

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ruksar Shaikh
Student ID:	x22174711
Programme:	Cloud Computing
Year:	2023-24
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	31/01/2024
Project Title:	Prediction of Resource Utilization in Cloud Computing using Machine Learning
Word Count:	XXX
Page Count:	28

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:	
Date:	30th January 2024

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):	

Prediction of Resource Utilization in Cloud Computing using Machine Learning

Ruksar Shaikh
x22174711

Abstract

In today's modern computing infrastructure, cloud computing has emerged as a pivotal paradigm, offering scalability and flexibility to satisfy the demands of a wide variety of specific applications. Maintaining optimal performance and cost-effectiveness inside cloud settings continues to be a significant problem, and one of the most important challenges is efficient resource utilisation. A resource utilisation prediction system is required to aid the resource allocator in providing optimal resource allocation. Accurate prediction is difficult in such a dynamic resource utilisation. The applications of machine learning techniques are the primary emphasis of this research project, which aims to predict resource utilisation in cloud computing systems. The dataset GWA-T-12 bitbrains (from distributed datacenter) have provided the data of timestamp, cpu usage, network transmitted throughput and Microsoft Azure traces has provided the data of cpu usage of cloud server.
To predict VM workloads based on CPU utilisation, we use machine learning models such as Linear Regression, Decision Tree Regression, Gradient Boosting Regression, and Support Vector Regression, as well as deep learning architectures such as Long Short-Term Memory (LSTM) and Bi-directional Long Short-Term Memory (BiLSTM). The Python programming language is used to carry out the implementation within the Google Colab environment. Bi-directional Long Short Term Memory approach is considered more effective as compared to other models in terms of CPU Utilisation and Network Transmitted Throughput as it R2 score is close to 1 hence can produce more accurate results.

Keywords: Cloud Computing, Machine Learning, Deep Learning, Resource Utilization, CPU Utilization, Network Transmission Throughput

1 Introduction

1.1 Background

Cloud service providers often adopt a pay-as-you-go pricing model, which can result in cost savings and increased flexibility for cloud users. The vast variety of improvements in cloud computing technology has resulted in a considerable growth in cloud users and the development of cloud-based applications to access various cloud computing services. Several scientific applications use cloud computing services, resulting in varying utilisation of cloud resources. As a result, efficient resource management is required to handle the shifting demand of users. Efficient resource management in a cloud computing environment can help to optimise resource utilisation, save costs, and improve performance.

Resource utilisation prediction is used to accomplish efficient resource management (Malik et al. (2022)). Predicting the consumption of cloud resources such as CPU, memory, and network throughput is critical for effective resource management (Kaur et al. (2019)). CPU utilisation is one of the most essential metrics for measuring the performance of host machines. It is also a prominent indicator for researchers to evaluate when attempting to anticipate the performance of hosts in the future. The central processing unit (CPU) is typically the resource that is subject to the highest amount of demand in virtualized settings. As a result, it is a significant contributor to resource shortages on cloud host devices (Mason et al. (2018)). Machine learning algorithms have gained a lot of attention and are becoming commonplace in cloud computing applications in recent years. Inspired by the structure of the brain, the Neural Network is one of the most versatile and successful machine learning techniques available. Because neural networks approximate functions, they can be used to solve a wide range of issues, from robotics to regression (Duggan et al. (2017)).

In this paper, we predict virtual machine CPU utilization using ML and DL predictive models. The aims of this research is to investigate the accuracy of a predictive models for predicting CPU utilization when compared to machine learning methods.

1.2 Research Question

How can machine learning and deep learning models be employed to predict CPU usage and Network Transmission Throughput in cloud computing environments, ensuring efficient resource utilization, performance enhancement, and cost-effectiveness?

1.3 Problem Statement

Efficient resource utilisation is still a major difficulty in modern cloud computing settings, affecting cloud systems' cost-effectiveness and performance. It is difficult to forecast resource utilisation in such dynamic contexts, even with cloud models' inherent scalability and flexibility. Inaccurate resource usage forecasting makes it difficult to allocate resources effectively and implement dynamic scaling plans, which can result in inefficient resource provisioning, higher operating expenses, and even performance bottlenecks. This mismatch in resource utilisation prediction creates a fundamental challenge for cloud computing infrastructures, impeding not only cost-efficiency but also the capacity to maintain service quality.

This paper is constructed as follows: Section 2 presents the overview of the existing works related to prediction of resource utilization using machine learning. Section 3 presents the method of research and explained the proposed architecture of the prediction models. Section 4 discussed the design specification. Section 5 discussed the implementation of the proposed solution. Section 6 presents evaluated results of proposed approach and section 7 presents conclusion and future work of the research.

2 Related Work

Resource usage prediction is becoming more and more popular due to recent advancements in the field of resource management(Amiri and Mohammad-Khanli (2017)) . The various prediction techniques based on deep learning and machine learning methodologies are compiled in this section.

2.1 Resource Utilisation using Machine Learning Techniques

Farahnakian et al. (2013) presents a linear regression-based CPU consumption prediction algorithm to maximise cloud computing resource utilisation. In order to minimise power usage and SLA violations, it includes live migration to balance active hosts based on resource demands. In order to avoid overloads and move underloaded hosts into low-power modes, historical CPU data forecasts host utilisation and triggers virtual machine migrations. Its efficacy is confirmed by real workload tracing studies, which show significant drops in energy consumption and SLA breaches while also indicating better dependability and efficiency in cloud environments.

Borkowski et al. (2016) proposes machine learning-based models for predicting resource use at the task and resource levels, which enables Cloud resource provisioning. Evaluations demonstrate significant gains in accuracy, with 20% reduction in prediction errors and up to 89% improvements in certain circumstances. In cloud computing, this method offers optimised resource allocation and customised provisioning.

Conforto et al. (2017) presents a unique machine learning-based resource usage prediction system for IaaS clouds that uses dynamic resource forecasting. It offers major improvements in IaaS infrastructure management and optimisation by combining historical data and real-time monitoring to optimise resource allocation, increase cost-efficiency, and improve overall IaaS performance.

Mehmood et al. (2018) demonstrates how crucial it is to allocate resources precisely on cloud systems to prevent waste or deterioration of service. It suggests an ensemble-based workload prediction system that builds precise predictive models through machine learning approaches. The project attempts to increase prediction accuracy for efficient resource utilisation by utilising multiple learners and stack generalisation.

Morariu et al. (2020) explores ML's role in enhancing scheduling and resource allocation in large-scale manufacturing. It addresses the complexities of production operations by leveraging historical data to create predictive models. These models, encompassing supervised and unsupervised ML algorithms, optimize scheduling decisions by learning from historical patterns. They forecast production schedules and efficiently allocate resources, highlighting their potential to improve scheduling accuracy and resource efficiency. The study emphasizes ML's capacity to streamline and optimize production processes within large-scale manufacturing systems, offering insights into how data-driven approaches can enhance operational efficiency in complex manufacturing environments.

Rohit Daid and Chen (2021) explores data center scheduling, emphasizing CPU utilization optimization and meeting service level agreement (SLA) requirements through machine learning (ML). Investigating challenges in CPU efficiency and SLA fulfillment, the study proposes a hybrid ML approach integrating clustering and regression models for scheduling. Using historical CPU data, predictive models enable proactive resource management, ensuring efficient CPU usage while meeting SLAs. The research showcases ML-driven scheduling's efficacy in optimizing CPU usage, enhancing resource allocation strategies, and upholding service commitments within data center environments. This emphasizes ML's potential to enhance data center operations and ensure adherence to SLAs while optimizing resource usage.

Khan et al. (2022) provides a thorough review of machine learning (ML) applications in cloud resource management. Covering the evolution of ML techniques, it explores their diverse applications in optimization, allocation, and predictive modeling within cloud environments. Ranging from traditional algorithms to advanced deep learning, the paper showcases ML's effectiveness in addressing resource allocation challenges. It also outlines future research directions, highlighting the integration of novel ML paradigms like federated learning and reinforcement learning. This comprehensive review acts as a roadmap, paving the way for innovation in ML-centric resource management, aiming to enhance adaptability and efficiency in cloud computing infrastructures.

Manam et al. (2023) proposes a novel approach in cloud computing, employing a Random Forest algorithm to optimize resource scheduling and reduce costs. The algorithm, known for ensemble learning, constructs decision trees for regression and classification. Benchmarked against XGBoost, Ridge, and Lasso models, it outperforms in predicting CPU and memory usage accuracy. Conducted using the Materna Dataset on Google Colaboratory, the research showcases the algorithm's efficacy in improving resource allocation and minimizing environmental impact.

Estrada et al. (2023) introduces a new method for forecasting CPU usage in virtualized settings. Using a simplified VM clustering technique, it improves prediction accuracy by grouping similar VMs based on resource usage patterns. Employing machine learning algorithms, it analyzes historical CPU data to create clusters of VMs with comparable behavior, enabling more accurate predictions. The paper details the model's methodology, algorithms used, experimental setup, and validation, demonstrating its potential for optimizing resource allocation and improving performance in virtual environments.

Khurana et al. (2023) focuses on refining Gradient Boosting models for forecasting CPU usage in cloud settings. This likely involves extensive parameter optimization, including hyperparameter fine-tuning, cross-validation techniques, and feature engineering. The study aims to develop an advanced predictive tool capable of accurately estimating CPU utilization in cloud environments. Such precision is vital for optimizing resource allocation and managing performance in cloud systems, ultimately aiming for cost-effectiveness and efficient operations by leveraging a sophisticated Gradient Boosting approach fine-tuned to the specific demands of cloud computing environments.

2.2 Resource Utilisation using Deep Learning Techniques

Wang et al. (2016) introduces a proactive VM deployment approach in cloud computing, using CPU utilization predictions via the ARIMA-BP neural network. By foreseeing performance issues, it revolutionizes deployment strategies, ensuring service quality and server efficiency. The method, with four key steps, aims to preemptively manage resources, optimize utilization, and improve overall performance in cloud environments.

Duggan et al. (2017) investigates the use of recurrent neural networks (RNNs) to predict CPU utilization in cloud computing. By analyzing historical CPU and network data, the study employs RNNs to capture temporal dependencies and forecast usage patterns. Demonstrating the effectiveness of RNNs in cloud environments, the research establishes their ability to accurately predict CPU utilization. These findings contribute to refining resource allocation strategies, emphasizing the potential for improved optimization within cloud infrastructures.

Nääs Starberg and Rooth (2021) focuses on managing CPU fluctuations in cloud computing by introducing an LSTM model. It forecasts CPU usage up to 30 minutes ahead, aiding in dynamic capacity scaling. Through performance evaluations against RNNs and state-of-the-art models, its accuracy in predicting future utilization is assessed. The LSTM's potential to optimize resource allocation and minimize costs in public cloud-hosted applications is highlighted, offering promising solutions for environmental impact and expense reduction.

B R et al. (2021) proposes a hybrid model for cloud resource utilization forecasting, combining SARIMA for seasonal workloads and LSTM/ARIMA for non-seasonal patterns. It highlights LSTM's accuracy in irregular patterns, SARIMA's effectiveness in forecasting future usage, and its significance in helping providers avoid resource over or under-provisioning.

Bal et al. (2022) addresses the problem of inefficient resource management affecting cloud computing performance. They introduce the RATS-HM technique, which incorporates the Improved Cat Swarm Optimization for task scheduling and the Group-Optimized Deep Neural Network (GO-DNN) for resource allocation. Performance metrics like power consumption, resource utilization, bandwidth utilization, memory utilization, and response time are utilized to evaluate the proposed model. Through simulation in CloudSim, they demonstrate that the RATS-HM strategy optimizes memory resource utilization more efficiently compared to current systems like FCFS, ITSEPM, and round-robin.

In Table 1 Review of works related to Resource Usage Prediction Techniques.

Table 1: Summarized related works of resource utilisation in cloud computing

Author	**Title**	**Dataset**	**Tool**	**Technique**	**Result**
Duggan et al. (2017)	Predicting host CPU utilization in cloud computing using recurrent neural networks	No application/ Dataset of CoMon project	PlanetLab	Recurrent Neural Network	Prediction accuracy is improved.
Rohit Daid and Chen (2021)	An effective scheduling in data centres for efficient CPU usage and service level agreement fulfilment using machine learning	Randomly generated data	Matlab	Linear Regression	Prediction accuracy is improved.
Manam et al. (2023)	A Machine Learning Approach to Resource Management in Cloud Computing Environments	Materna dataset Trace 3	Google Colaboratory platform	Random Forest algorithm	Prediction accuracy is improved.

Continued on next page

Table 1 – continued from previous page

Author	Title	Dataset	Tool	Technique	Result
Mehmood et al. (2018)	Prediction Of Cloud Computing Resource Utilization	Google cluster usage trace data	Cloud system	Ensemble based workload prediction mechanism	Prediction accuracy is improved.
B R et al. (2021)	Resource Utilization Prediction in Cloud Computing using Hybrid Model	Bitbrains dataset	Experiment was conducted using fastStorage, real trace data of Bitbrains data center	SARIMA, LSTM, ARIMA	Prediction accuracy is improved.
Conforto et al. (2017)	Adaptive Resource Utilization Prediction System for Infrastructure as a Service Cloud	Bitbrains dataset	fastStorage of Bitbrains data center	ARIMA and Autoregressive Neural Network (AR-NN)	Prediction accuracy is improved.
Wang et al. (2016)	Research on the Prediction Model of CPU Utilization Based on ARIMA-BP Neural Network	IBM Server	Xen System	ARIMA-BP neural network	Prediction can be improved.
Nääs Starberg and Rooth (2021)	Predicting a business application's cloud server CPU utilization using the machine learning model LSTM	Afry dataset	Python	LSTM	Prediction accuracy is improved.

The table showcases a variety of approaches leveraging different datasets, tools, and machine learning algorithms such as recurrent neural networks, linear regression, random forests, and LSTM among others. Several studies demonstrate improved prediction accuracy when forecasting resource utilization in cloud environments. However, a compelling trend surfaces from the reviewed literature: the utilization of LSTM-based models consistently demonstrates enhanced predictive capabilities across various datasets. The Bidirectional LSTM, with its ability to capture long-term dependencies and process sequential data bidirectionally, presents itself as a robust choice for modeling the complex temporal patterns inherent in cloud resource usage.

The choice of BiLSTM model stems from its capacity to effectively capture both past and future context, which is particularly relevant in resource utilization forecasting where historical trends and future behavior significantly impact predictions. The utilization of this model offers the potential to enhance accuracy, thereby aiding in proactive resource allocation and optimization in cloud environments.

3 Methodology

Steps of Research Methodology for Resource Utilization Prediction in Cloud Computing is as follows and also presented in figure 1:

- **Research Understanding:** This study aims to predict the accuracy of VM CPU Utilisation and Network Transmission Throughput utilizing the ML and DL predictive models in cloud computing environments which focuses on achieving efficient resource utilization, enhanced performance, and cost reduction.

- **Data Collection:** Qualitative and quantitative data from open-source repositories (Bitsbrain dataset from (http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains), Microsoft Azure traces from GitHub) encompassing VM CPU Utilization and Network Transmission Throughput.

- **Data Pre-processing:** Cleaning data, performing feature engineering for model readiness. Addressing missing values and preparing datasets for ML and DL predictive models training.

- **Predictive Model Creation:** Training the predictive ML and DL models using the selected datasets. BiLSTM is chosen for its ability to capture complex temporal dependencies which aims to predict VM CPU Utilisation and Network Transmission Throughput accurately.

- **Evaluation:** Metrics include Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error, and R-squared (R2) value to assess prediction accuracy.

- **Performance Criteria:** Aimed at achieving efficient resource utilization, enhancing performance, and reducing operational costs.

- **Experimentation and Feedback:** Experimental scenarios are meticulously designed with controlled variables to rigorously test the predictive models. These experiments aim to validate the model's performance and reliability in predicting resource utilization.

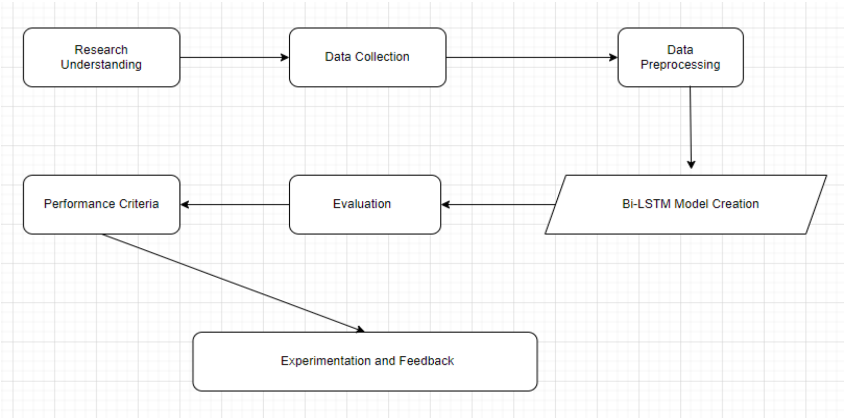


Figure 1: Flowchart of Research Methodology
(Source: Created by learner)

3.1 Dataset Description

This research utilises two key datasets, the Bitbrains dataset obtained from (http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains) and the Microsoft Azure Traces 2017 dataset sourced from GitHub.

The Bitbrains dataset specifically focus on the prediction of CPU utilization and network transmission throughput and Microsoft Azure Traces dataset focuses on CPU utilisation in a time series context. The Bitbrains dataset includes Timestamp, CPU usage, and network transmission throughput data as primary parameter.

Similarly, the Microsoft Azure Traces 2017 dataset contributes CPU usage data to predict CPU utilization and network transmission throughput patterns.
The datasets afford access to key resource metrics including CPU usage which is crucial for the predictive models in this study. To ensure consistency and comparability, the resource utilization metrics across these datasets undergo normalization, scaling these measurements within a range from 0 to 1.

Each resource within the datasets is individually normalized, with a value of 1 signifying the resource's maximum capacity concerning all machines encompassed within the trace. This normalization methodology fosters a standardized approach to resource analysis and prediction across the datasets, ensuring robust and comparable insights into resource utilization patterns.

3.2 Architecture

The resource provisioning model within cloud computing integrates various predictive Machine Learning (ML) and Deep Learning (DL) models, including Linear Regression, Decision Tree Regression, Support Vector Regression, Gradient Boosting Regression, LSTM, and Bi-LSTM. This model acts as a comprehensive framework which guides the allocation and management of critical resources like network bandwidth, storage capacities, and CPU power based on insights derived from these predictive models in figure 2.

In cloud environments, the availability of computing resources such as network capabilities, storage capacities, and CPU power forms the cornerstone of service provision. Predictive ML and DL models play a crucial role by predicting CPU usage and Network Transmission Throughput which significantly impacting these resources. These models enable cloud service providers to anticipate resource requirements more accurately, thus optimizing the allocation of network, storage, and CPU resources to align with projected demand. By harnessing the insights from these models, cloud environments achieve enhanced resource utilization and allocation efficiency.

In both reservation-based and on-demand scenarios, predictive modelling insights have a significant impact on resource allocation techniques. Predictive models are used to accurately predict resource requirements over time, which helps allocation strategies for reservations. This method guarantees ongoing usage in line with expected demands while ensuring efficient resource reservation and minimising waste. Predictive models simultaneously support real-time resource optimisation for on-demand scenarios by dynamically modifying CPU and network resources in response to urgent needs. During fluctuating demands, this dynamic adaptability greatly improves system performance and resource utilisation.

Optimization strategies, predictive ML and DL models play a pivotal role in various aspects of resource management. Real-time on-demand resource optimization leverages these models to swiftly adapt CPU and network resources to immediate needs. This agile adjustment significantly improves resource utilization efficiency, thereby enhancing system performance during fluctuating demands. Additionally, reservation-based optimization harnesses predictive models to minimize unnecessary resource reservations, ensuring system stability while maintaining efficiency. Furthermore, these models facilitate resource expansion during sudden demand surges, allowing proactive scaling to ensure adequate performance during peak times and optimizing cost-effectiveness by dynamically adjusting resource allocation. Overall, predictive ML and DL models form the backbone of efficient resource optimization strategies, driving improved performance and cost-effectiveness within cloud computing environments.

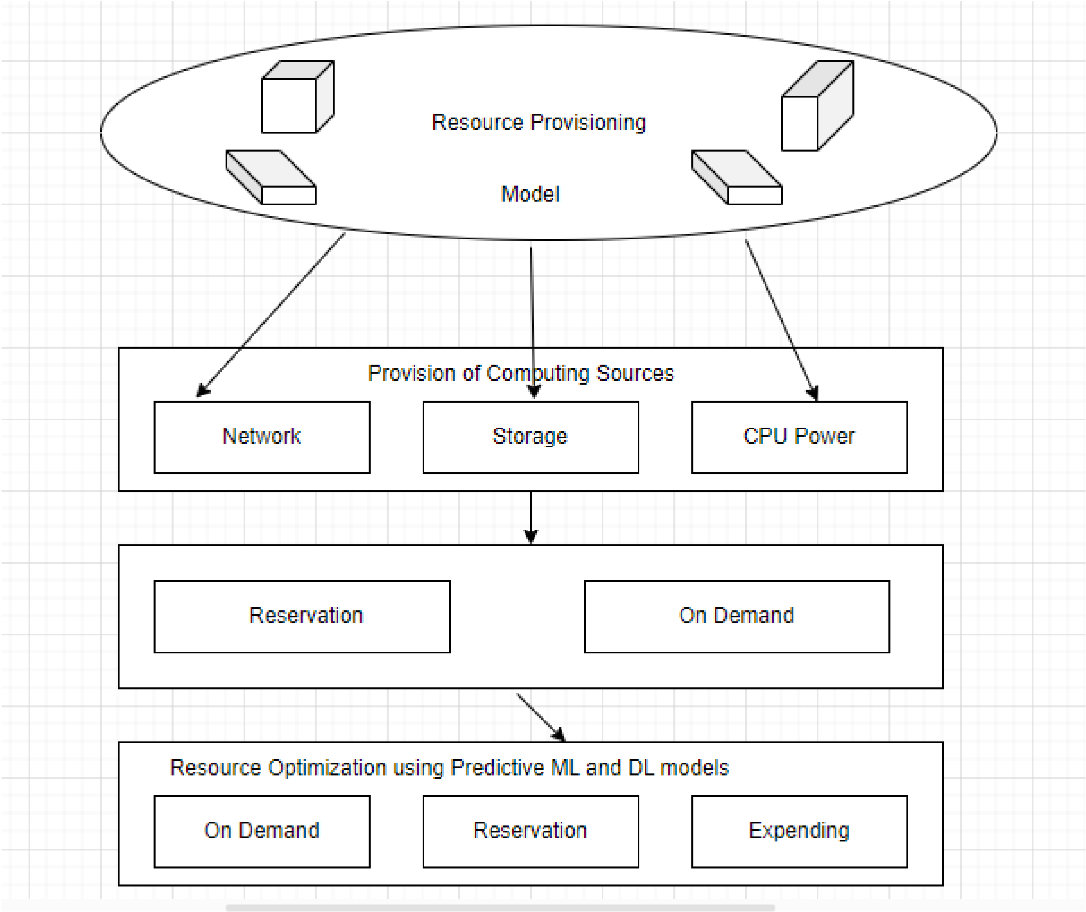


Figure 2: Resource Provisioning Architecture
(Source: Created by learner)

3.3 Proposed Approach

Many studies have delved into harnessing the potential of Machine Learning (ML) and Deep Learning (DL) models to predict resource utilization within cloud computing systems. These models play a pivotal role in predicting and efficiently managing the allocation of critical resources such as CPU utilization and Network-transmitted throughput. This research focused on exploring a comprehensive set of predictive models, encompassing four fundamental ML models: Linear Regression, Decision Tree Regression, Gradient Boosting Regression, Support Vector Regression, alongwith two influential DL models: Long Short-Term Memory (LSTM) and its extension, Bidirectional Long Short-Term Memory (BiLSTM) refer figure 3 .

3.3.1 Linear Regression (LiR):

Linear Regression is a traditional regression model which aims to establish a linear relationship between independent variables (features) and a dependent variable (resource utilization). In predicting CPU utilization and Network-transmitted throughput in cloud computing, Linear Regression would attempt to identify straightforward linear relationships between various factors influencing resource usage and the actual utilization refer figure 3a.

3.3.2 Decision Tree Regression (DTR):

Decision Tree Regression constructs a tree-like structure based on data features, enabling the prediction of resource utilization. In the context of cloud resource prediction, Decision Tree Regression would create decision rules based on attributes such as network transmission throughput, CPU usage, to predict resource utilization levels refer figure 3b.

3.3.3 Gradient Boosting Regression (GBR):

Gradient Boosting Regression builds an ensemble of decision trees, iteratively improving predictions by minimizing errors. In cloud computing resource utilization prediction, this model combines multiple approaches to refine predictions by understanding complex relationships between different factors affecting resource utilization refer figure 3c.

3.3.4 Support Vector Regression (SVR):

Support Vector Regression establishes a hyperplane that best represents the relationship between input features and resource utilization. In the context of cloud computing, it identifies a high-dimensional boundary to predict CPU utilization and Network-transmitted throughput based on various factors refer figure 3d.

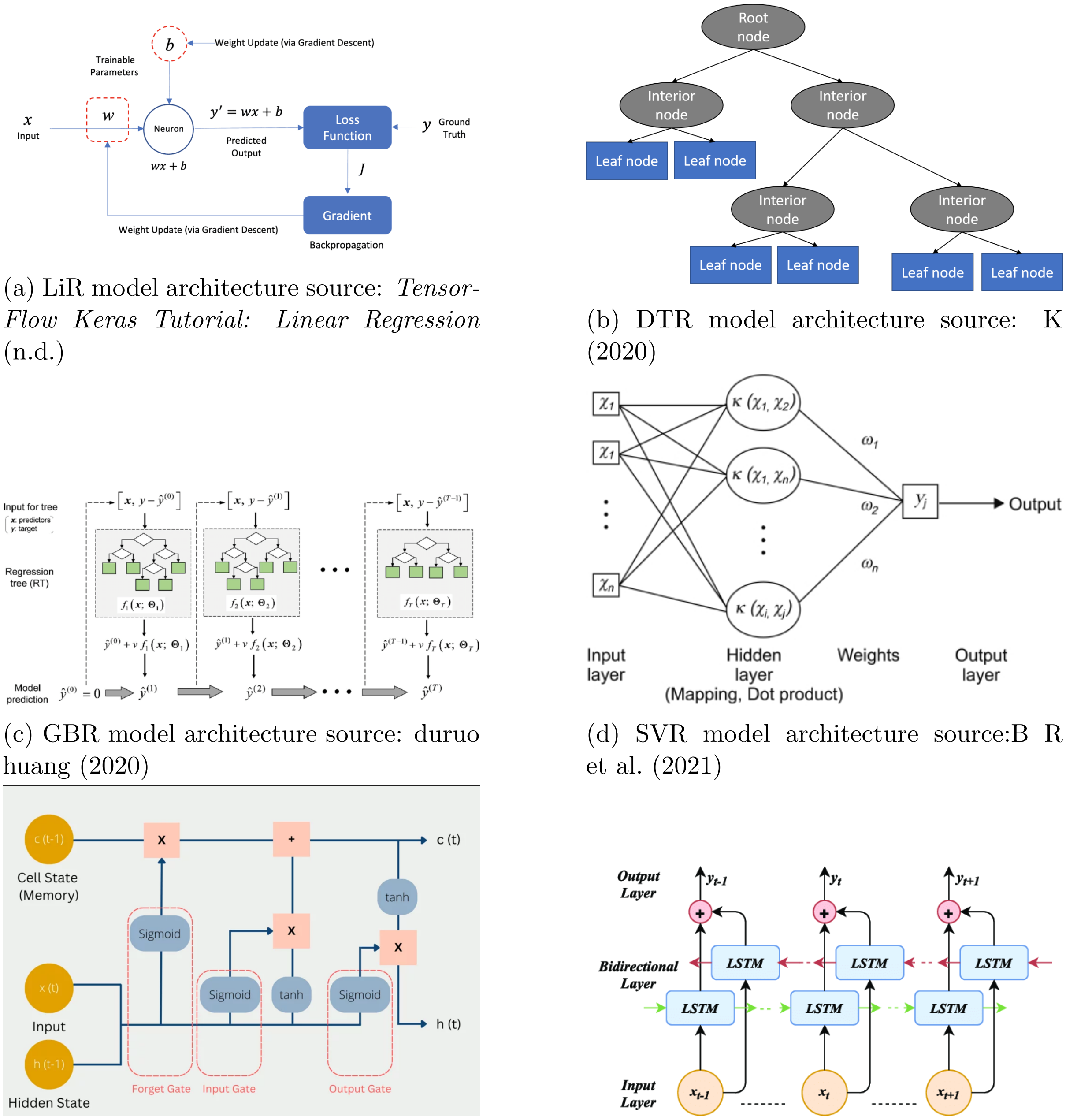
3.3.5 Long Short Term Memory (LSTM):

LSTM is a type of recurrent neural network (RNN) which excels in capturing dependencies in sequential data. In predicting resource utilization in cloud computing, LSTM would focus on identifying temporal patterns in CPU usage and Network-transmitted throughput, recognizing subtle changes and patterns over time refer figure 3e.

3.3.6 Bi-direction Long Short Term Memory (BiLSTM):

BiLSTM extends LSTM by processing data in both forward and backward directions to capture past and future context simultaneously. In cloud computing resource utilization prediction, BiLSTM would comprehensively analyze temporal dependencies in both directions, providing a more holistic understanding of CPU utilization and Network-transmitted throughput patterns refer figure 3f.

The comparison between traditional regression models like Linear Regression, Decision Tree Regression, Gradient Boosting Regression, and Support Vector Regression with deep learning models like LSTM and BiLSTM mirrors the evaluation of simpler, rule-based approaches against complex, memory-enhanced models in their ability to predict resource utilization patterns in cloud computing that particularly focusing on CPU usage and Network-transmitted throughput.



(a) LiR model architecture source: *TensorFlow Keras Tutorial: Linear Regression* (n.d.)

(b) DTR model architecture source: K (2020)

(c) GBR model architecture source: duruo huang (2020)

(d) SVR model architecture source:B R et al. (2021)

(e) LSTM model architecture source:*LSTM* (2022)

(f) BiLSTM model architecture source: *BiLSTM* (2022)

Figure 3: Predictive models architecture

The proposed approach is grounded in utilizing the BiLSTM (Bidirectional Long Short-Term Memory) model to improve the accuracy of predictions for both VM CPU Utilization and Network Transmission Throughput. Firstly, datasets from Bitsbrain (for CPU Utilization and Network Transmission Throughput) and Microsoft Azure Traces 2017 (specifically for CPU Utilization) are collected and meticulously preprocessed to ensure completeness and relevance in the context of the study. After that, the predictive models is implemented and rigorously trained using these datasets. The primary focus lies in assessing key predictive metrics such as Mean Absolute Error(MAE), Mean Squared Error(MSE), Root Mean Squared Error(RMSE) and R2 score to evaluate the model's accuracy in predicting VM resource utilization. The ultimate goal is twofold: to optimize resource utilization and bolster overall performance within cloud computing environments. This is achieved through a comprehensive examination and comparison of the BiLSTM model against established machine learning frameworks which aims to pinpoint the most effective model for precise prediction and efficient resource allocation strategies.

4 Design Specification

The design specification explains the underlying architecture and requirements for the implementation of an accurate prediction of VM resource utilization which focuses on CPU Utilization and Network Transmission Throughput within cloud computing environments. It outlines the techniques, framework, architecture and essential requirements. Post-acquisition of datasets from open-source repositories, an integral phase involves meticulous data preprocessing. This stage involves checking for missing values, ensuring data completeness, and applying appropriate feature engineering techniques to prepare the datasets for subsequent analysis.

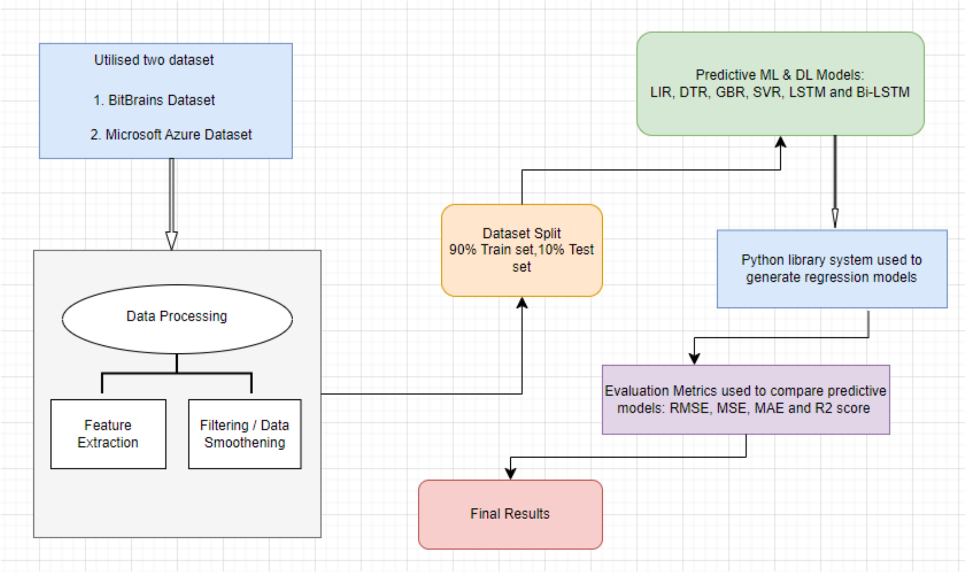

Utilised two dataset
1. BitBrains Dataset
2. Microsoft Azure Dataset
Data Processing
Feature Extraction
Filtering / Data Smoothening
Dataset Split 90% Train set,10% Test set
Predictive ML & DL Models: LIR, DTR, GBR, SVR, LSTM and Bi-LSTM
Python library system used to generate regression models
Evaluation Metrics used to compare predictive models: RMSE, MSE, MAE and R2 score
Final Results

Figure 4: Roadmap of Design Specification for Implementation
(Source: Created by learner)

The proposed architecture involves predictive ML and DL models leveraging Linear

Regression, Decision Tree Regression, Gradient Boosting Regression, Support Vector Regression, Long Short Term Memory, Bi-directional Long Short Term Memory (BiLSTM) networks for predicting the accuracy of VM CPU Utilization and Network Transmission Throughput in cloud computing environments. The model's architecture comprises bi-directional recurrent neural networks, which enable comprehensive analysis of temporal dependencies within the input data. BiLSTM networks incorporate both forward and backward information flow, allowing the model to capture intricate patterns in time series data. The implementation environment serves as the foundation for developing, training, and evaluating the predictive models, facilitating efficient experimentation and iterative model refinement. To ascertain the predictive accuracy and effectiveness of the BiLSTM model, evaluation metrics quantify the model's performance in predicting resource utilization, aiding in assessing the accuracy and reliability of the predictions. An essential facet involves conducting a comparative analysis between the BiLSTM model, Long Short Term Memory and Linear Regression, Support Vector Regression, Decision Tree Regression, and Gradient Boosting Regression. This comparative study aims to highlight the superior predictive performance of the BiLSTM model in predicting VM CPU Utilization and Network Transmission Throughput within cloud computing environments. Controlled experimental scenarios are meticulously designed to facilitate a comprehensive comparison of predictive capabilities among various models. These controlled experiments contribute to showcasing the efficacy of the BiLSTM model in forecasting resource utilization, thereby substantiating its applicability in real-world cloud computing environments.

5 Implementation

This research project employs Python programming language within Google Colab to predict resource utilization. Figure 4 shows the implementation steps.

- This research utilizes two datasets for resource utilization, the Bitbrains dataset selecting parameters such as CPU usage, network transmission throughput and timestamp. Microsoft Azure Traces dataset selecting parameters such as min cpu, max cpu, avg cpu usage and timestamp. For the both the datasets, implementation is carried out separately Bitbrains dataset is used to predict the CPU Utilization and Network Transmission Throughput and Microsoft Azure Traces is used to predict CPU Utilization.

- Before starting the modeling process, thorough checks for missing values and handling of such instances are diligently executed to ensure the integrity and completeness of the datasets. Feature scaling techniques, particularly MinMaxScaler, are employed to normalize the data within both the Bitbrains and Microsoft Azure Traces datasets. This crucial preprocessing step standardizes the attributes, ensuring uniformity and optimal performance during machine learning model training.

- The implementation phase begins with an extensive Exploratory Data Analysis (EDA) aimed at comprehensively understanding the datasets. This in-depth examination includes scrutinizing critical details such as dataset shape, size, data types, mean values, column names, counts, standard deviations, and the range between minimum and maximum values. These statistical insights provide a holistic view of the datasets, essential for subsequent modeling.

- The machine learning process commences by identifying the target column, denoted as 'y', which will be predicted by the models. To facilitate model training and evaluation, the dataset is split into four subsets: X-train, X-test, y-train, and y-test. This split is conducted with a 90-10 ratio, where 90% of the data is allocated for training and 10% for testing purposes.
- This facilitates the effective implementation of some machine learning and deep learning procedures like Linear Regression, Decision Tree Regression, Gradient Boosting Regression, and Support Vector Regression, Long Short Term Memory and Bi-directional Long Short Term Memory.
- A robust library system is harnessed to execute these algorithms, ensuring precise outcomes. Moreover, this process adeptly configures the algorithms by leveraging the capabilities offered by the library system.
- Evaluation metrics such as Mean Square Error, Mean Absolute Error, R Square error, and Root Mean Square Error are calculated to predict the accuracy of each regression model.
- Real-time testing is conducted to validate the models effectiveness in practical scenarios using test data, ensuring their viability and accuracy in a live cloud environment. Continuous monitoring and optimization of these models remain pivotal, allowing for adjustments based on evolving cloud infrastructure dynamics and patterns within the datasets. Ultimately, this implementation aims to provide a robust predictive system facilitating efficient resource allocation, improved performance, and cost reduction within cloud computing infrastructures.

5.1 Experimental Setup

Data Collection:

- **Bitbrains Dataset:**
    - Obtain the Bitbrains dataset, ensuring it includes relevant metrics like CPU usage, Network Transmission Throughput and Timestamp.
- **Microsoft Azure Dataset:**
    - Collect the Microsoft Azure dataset, specifically focusing on CPU utilization data, ensuring it includes a similar set of features for consistency and comparison.

Data Preprocessing:

- **Data Cleaning and Integration:**
    - Cleanse the datasets to remove missing values, outliers or inconsistencies.
    - Using both Bitbrains and Microsoft Azure datasets, ensuring compatibility in terms of features and timeframes for CPU utilization.
- **Feature Engineering:**

- Extracted relevant features for modeling such as historical CPU usage, network traffic patterns, time of day, etc.
- Transform categorical variables, normalize/standardize numerical variables if required.

Experimental Setup:

- **Model Selection:**
    - Chosen four traditional Machine Learning models: Linear Regression (LIR), Decision Tree Regression (DTR), Support Vector Regression (SVR), Gradient Boosting Regression (GBR).
    - Selected two Deep Learning models: Long Short-Term Memory (LSTM) and Bi-directional Long Short-Term Memory (BiLSTM).
- **Data Splitting:**
    - Divided the datasets into training, validation, and test sets preserving temporal order if applicable.
- **Model Training:**
    - Trained each of the selected models (LIR, DTR, SVR, GBR, LSTM, BiLSTM) separately using the training data.
- **Model Evaluation:**
    - Evaluated the models performance using the validation set to tune hyperparameters and ensure generalization.
    - Compared the models based on metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R-squared.

Specific Model Testing:

- **Bitbrains Dataset Testing:**
    - Tested the ML and DL models (LIR, DTR, SVR, GBR, LSTM, BiLSTM) on the Bitbrains dataset to predict CPU usage and Network Transmission Throughput.
- **Azure Dataset Testing:**
    - Tested the ML and DL models (LIR, DTR, SVR, GBR, LSTM, BiLSTM) on the Microsoft Azure dataset to predict CPU utilization.

Analysis and Conclusion:

- **Performance Analysis:**
    - Analyzed and compared the performance of each model in predicting CPU usage and Network Transmission Throughput.
- **Efficiency and Cost-Effectiveness Evaluation:**
    - Considered the computational efficiency and cost-effectiveness of models concerning resource utilization.

5.2 Tools and Technology Stack:

Programming Languages:

- **Python:** It is widely used for data preprocessing, model implementation (using libraries like scikit-learn, TensorFlow, Keras) and analysis.

Libraries and Frameworks:

- **scikit-learn:** It is used for implementing traditional Machine Learning models (Linear Regression, Decision Tree Regression, Support Vector Regression, Gradient Boosting Regression).
- **TensorFlow/Keras:** It is used for developing and training Deep Learning models (LSTM, BiLSTM).
- **Pandas:** It is used for data manipulation and preprocessing.
- **NumPy:** It is used for numerical computations.
- **Matplotlib/Seaborn:** It is used for data visualization.

Cloud Platforms:

- **Google Colab:** It is used for implementation, experimentation, and collaboration.
- **Google Drive:** It is used for storing and accessing datasets.

Data Collection and Management:

- **Bitbrains Dataset:** It is managed and accessed through Google Drive.
- **Microsoft Azure Dataset** It is managed and accessed through Google Drive.

Experimentation and Analysis:

- **Model Training and Validation:** Used Python-based machine learning and deep learning libraries to train and validate models within Google Colab.
- **Model Evaluation Metrics:** Utilized Python libraries to evaluate models based on performance metrics (MSE, RMSE, R-squared, MAE) within Google Colab.
- **Data Visualization:** Matplotlib and Seaborn for visualizations within Colab notebooks.

6 Evaluation

In this section, the effectiveness of conventional machine learning algorithms as described in literature is assessed against the proposed approach. Employing the Scikit-Learn library, the experiments are conducted on the Google Colaboratory platform, serving as the environment for training and testing. Five distinct machine learning algorithms are evaluated: Linear Regression, Decision Tree Regression, Gradient Boosting Regression, Support Vector Regression, and the LSTM deep learning model. These approaches are compared to gauge their performance in contrast to the proposed BiLSTM model.

6.1 Performance Metrics

6.1.1 Root Mean Squared Error (RMSE):

RMSE is a measure of the differences between predicted values and observed values. It represents the square root of the average of the squared differences between the predicted and actual values. The formula for RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

Here, y_i represents the actual value, y_i^2 represents the predicted value, and n is the number of samples. (Sourced from Chugh (2020))

6.1.2 R-squared (R2) Score:

R2 score represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It is calculated as the ratio of the explained variation to the total variation. The formula for R2 score is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

Here, y_i represents the actual value, y_i^2 represents the predicted value, and n is the number of samples. (Sourced from Chugh (2020))

6.1.3 Mean Squared Error (MSE):

MSE measures the average of the squares of errors or deviations. It's calculated by taking the average of the squared differences between predicted and actual values. The formula for MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

Here, y_i represents the actual value, y_i^2 represents the predicted value, and n is the number of samples. (Sourced from Chugh (2020))

6.1.4 Mean Absolute Error (MAE):

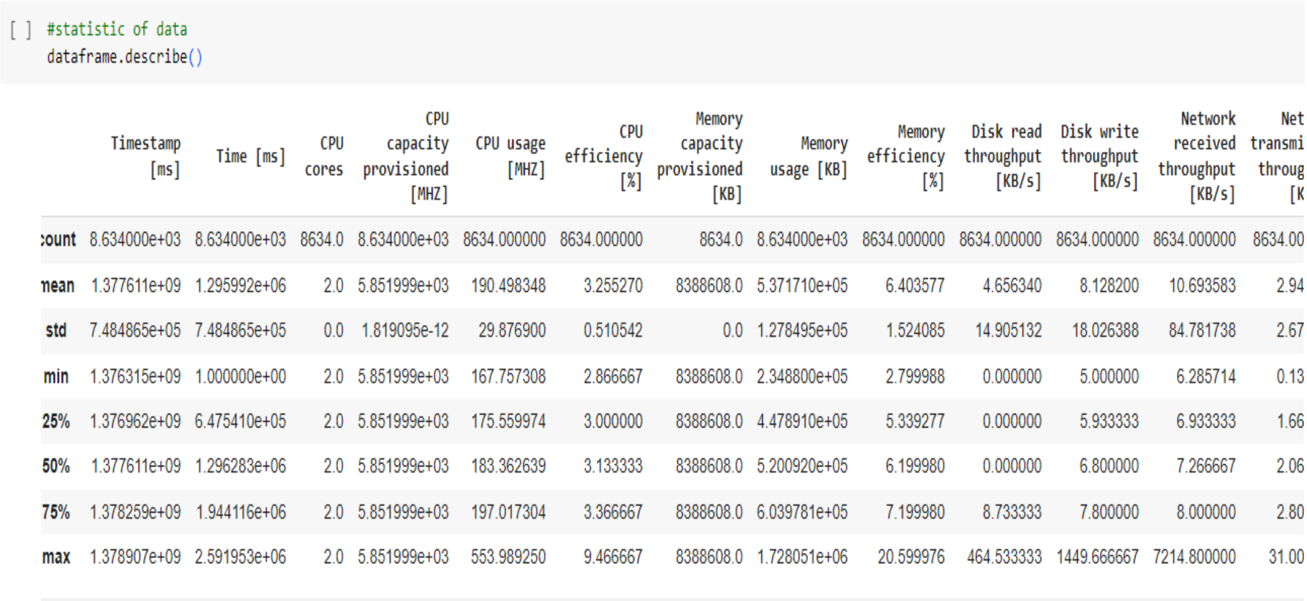
MAE is the average of the absolute differences between predicted and actual values. It measures the average magnitude of errors without considering their direction. The formula for MAE is:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

Here, y_i represents the actual value, y_i^2 represents the predicted value, and n is the number of samples. (Sourced from Chugh (2020))

6.2 Dataset Analysis

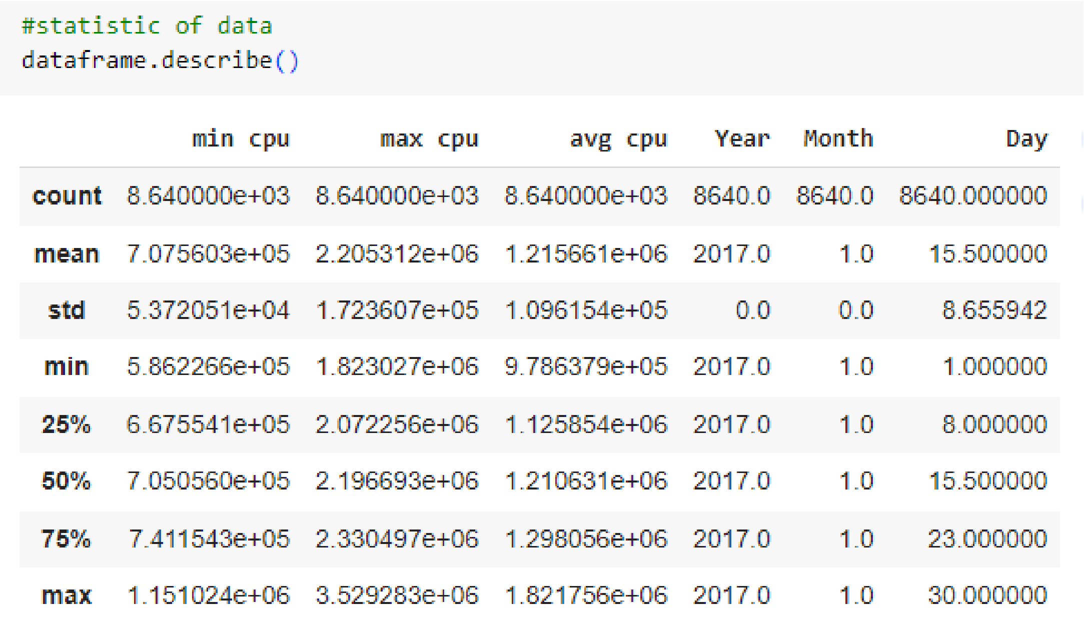
The figure 5 and figure 6 presented below illustrates the descriptive statistics extracted from the dataset. It comprehensively displays key statistical measures including counts, means, standard deviations, minimum and maximum values for all columns within the dataset.



```
[ ] #statistic of data
    dataframe.describe()
```

	Timestamp [ms]	Time [ms]	CPU cores	CPU capacity provisioned [MHZ]	CPU usage [MHZ]	CPU efficiency [%]	Memory capacity provisioned [KB]	Memory usage [KB]	Memory efficiency [%]	Disk read throughput [KB/s]	Disk write throughput [KB/s]	Network received throughput [KB/s]	Net transmi throug [K
ount	8.634000e+03	8.634000e+03	8634.0	8.634000e+03	8634.000000	8634.000000	8634.0	8.634000e+03	8634.000000	8634.000000	8634.000000	8634.000000	8634.00
nean	1.377611e+09	1.295992e+06	2.0	5.851999e+03	190.498348	3.255270	8388608.0	5.371710e+05	6.403577	4.656340	8.128200	10.693583	2.94
std	7.484865e+05	7.484865e+05	0.0	1.819095e-12	29.876900	0.510542	0.0	1.278495e+05	1.524085	14.905132	18.026388	84.781738	2.67
min	1.376315e+09	1.000000e+00	2.0	5.851999e+03	167.757308	2.866667	8388608.0	2.348800e+05	2.799988	0.000000	5.000000	6.285714	0.13
25%	1.376962e+09	6.475410e+05	2.0	5.851999e+03	175.559974	3.000000	8388608.0	4.478910e+05	5.339277	0.000000	5.933333	6.933333	1.66
50%	1.377611e+09	1.296283e+06	2.0	5.851999e+03	183.362639	3.133333	8388608.0	5.200920e+05	6.199980	0.000000	6.800000	7.266667	2.06
75%	1.378259e+09	1.944116e+06	2.0	5.851999e+03	197.017304	3.366667	8388608.0	6.039781e+05	7.199980	8.733333	7.800000	8.000000	2.80
max	1.378907e+09	2.591953e+06	2.0	5.851999e+03	553.989250	9.466667	8388608.0	1.728051e+06	20.599976	464.533333	1449.666667	7214.800000	31.00

Figure 5: BitBrains Dataset Analysis
(Source: Generated using the Google Colab Notebook)



```
#statistic of data
dataframe.describe()
```

	min cpu	max cpu	avg cpu	Year	Month	Day
count	8.640000e+03	8.640000e+03	8.640000e+03	8640.0	8640.0	8640.000000
mean	7.075603e+05	2.205312e+06	1.215661e+06	2017.0	1.0	15.500000
std	5.372051e+04	1.723607e+05	1.096154e+05	0.0	0.0	8.655942
min	5.862266e+05	1.823027e+06	9.786379e+05	2017.0	1.0	1.000000
25%	6.675541e+05	2.072256e+06	1.125854e+06	2017.0	1.0	8.000000
50%	7.050560e+05	2.196693e+06	1.210631e+06	2017.0	1.0	15.500000
75%	7.411543e+05	2.330497e+06	1.298056e+06	2017.0	1.0	23.000000
max	1.151024e+06	3.529283e+06	1.821756e+06	2017.0	1.0	30.000000

Figure 6: Microsoft Azure Traces Dataset Analysis
(Source: Generated using the Google Colab Notebook)

6.3 Data Visualization

In this section, Data Visualization simplifies complex information by turning it into visual pictures, helping us to understand data better. Figure 7, the CPU Utilization graph based on BitBrains dataset presents a time-series depiction of CPU utilization over a specific period. The graph exhibits temporal variations, portraying the percentage of CPU usage across distinct intervals.

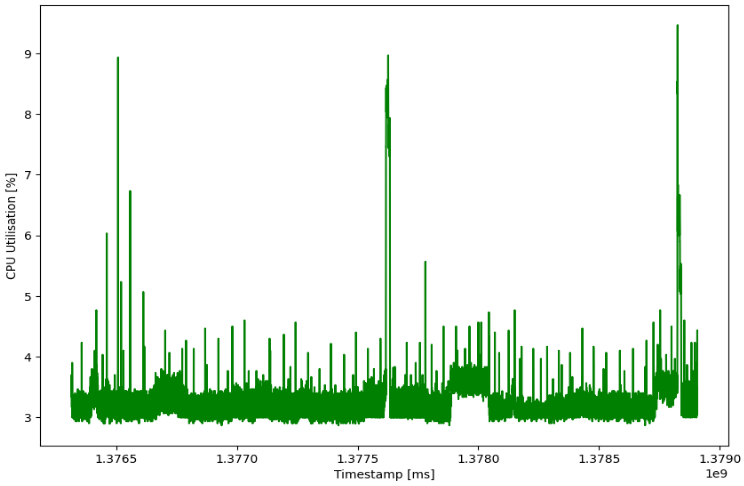


Figure 7: BitBrains Dataset CPU Utilisation
(Source: Generated using the Google Colab Notebook)

Figure 8, shows the network-transmitted throughput within the cloud environment, exhibiting the dynamic changes and variations in network data transmission rates over a given amount of time. Insights into overall network performance and data transmission patterns are provided by the graph, which is a visual representation of actual network throughput statistics. These insights are crucial for comprehending network efficiency and capacity utilisation.

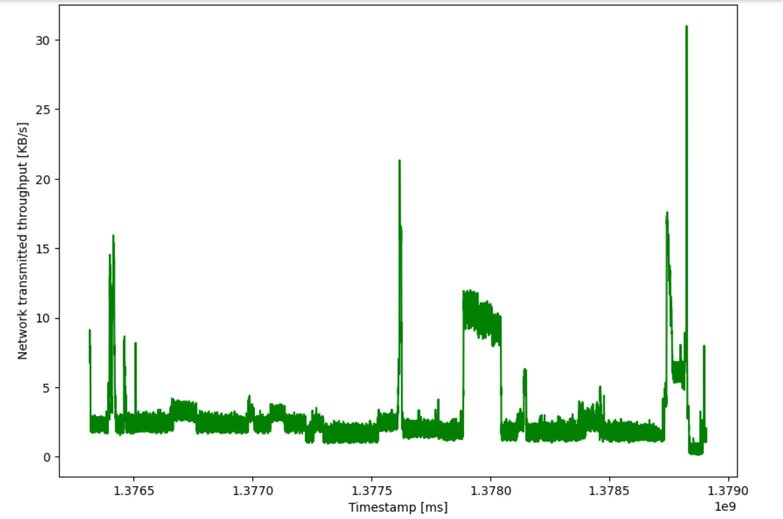


Figure 8: BitBrains Dataset Network Transmission Throughput
(Source: Generated using the Google Colab Notebook)

Figure 9, presents a comprehensive depiction of CPU Utilization trends over a specified duration, showcasing the highest peaks denoting maximum CPU usage, lowest troughs representing minimum CPU usage, and the trajectory of average CPU utilization. These metrics collectively offer a comprehensive overview of CPU performance within the Azure environment.

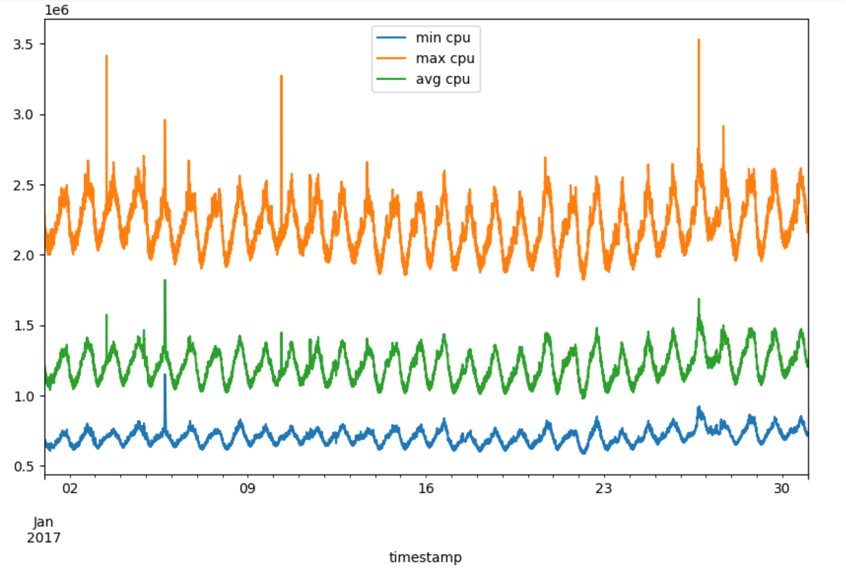


Figure 9: Microsoft Azure Dataset CPU Utilisation
(Source: Generated using the Google Colab Notebook)

6.4 Evaluation of Resource Utilisation for Machine Learning and Deep Learning Models

In this research project, we have conducted implementation using two publicly available datasets i.e BitBrains and Microsoft Azure Traces 2017. We have evaluated the CPU Utilisation and Network Transmitted Throughput using BitBrains dataset, alongwith CPU Utilisation evaluated for Microsoft Azure dataset.

6.4.1 Evaluation of CPU Utilisation

The evaluated results of CPU Utilisation using BitBrains dataset are presented in details in Table 2 and Figure 10. Also for MicroSoft Azure dataset, evaluated results of CPU Utilization are presented in details in Table 3 and Figure 11. The MSE, MAE, RMSE and R2 metrics of the ML and DL algorithms compared in this paper are shown in Table 2 and Table 3. The results show that for RMSE, Decision Tree Regression, Gradient Boosting Regression algorithms had higher error values when compared to BiLSTM model which performed better than the compared models shown in Table 2 and Table 3. The evaluated results of the CPU utilization for the prediction and actual values of the machine learning models are presented in Figure 10 and Figure 11. Hence, BiLSTM performed better than the compared approaches followed by Linear Regression and LSTM.

Table 2: For BitBrains dataset - Comparison of machine learning and deep learning algorithms for CPU Utilization prediction

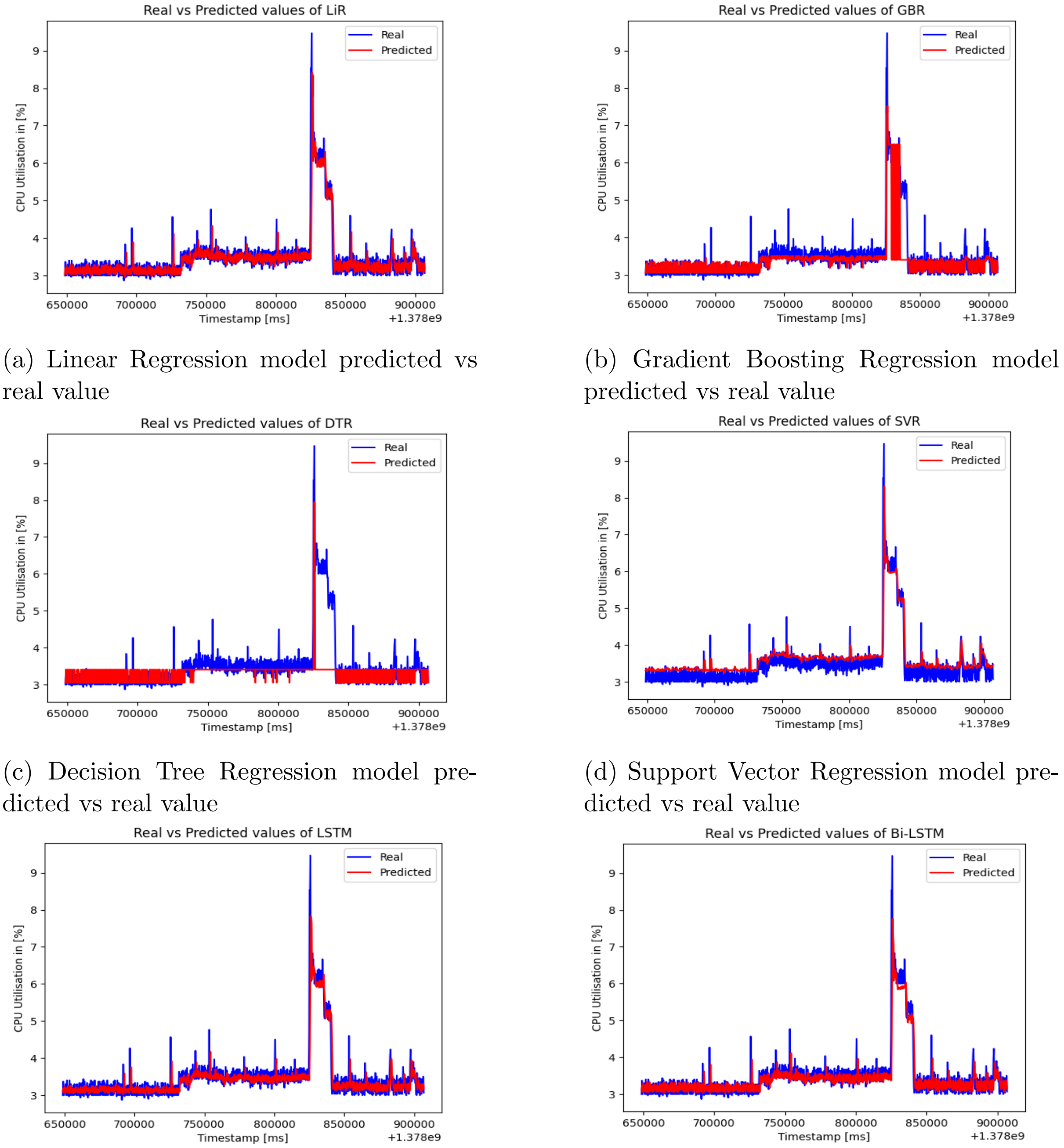
Model	MSE	MAE	RMSE	R2 score
LiR	0.0027	0.0215	0.0521	0.7794
DTR	0.0099	0.0402	0.0995	0.1951
GBR	0.0055	0.0294	0.0747	0.5465
SVR	0.0030	0.0389	0.0556	0.7488
LSTM	0.0026	0.0233	0.0515	0.7843
BiLSTM	0.0024	0.0224	0.0490	0.8042

Table 3: For Microsoft Azure dataset - Comparison of machine learning and deep learning algorithms for CPU Utilization prediction

Model	MSE	MAE	RMSE	R2 score
LiR	0.0002	0.0131	0.0169	0.9833
DTR	0.0023	0.0390	0.0480	0.8661
GBR	0.0010	0.0255	0.0321	0.9399
SVR	0.0015	0.0337	0.0388	0.9127
LSTM	0.0009	0.0239	0.0304	0.9462
BiLSTM	0.0004	0.0169	0.0214	0.9732

Figure 10 and figure 11 illustrates the predictions for CPU utilization using a range of machine learning and deep learning techniques, including Linear Regression(LIR), Gradient Boosting Regression(GBR), Decision Tree Regression(DTR), Support Vector Regression(SVR), Long Short Term Memory(LSTM) and Bi-directional Long Short Term Memory (BiLSTM). These predictive models enable accurate forcasting of the CPU utilization, providing valuable insights into the resource demands and usage patterns within the cloud environment.

Figure 13 presents the predictions for the network transmission throughput, utilizing machine learning and deep learning models, including Linear Regression(LIR), Gradient Boosting Regression(GBR), Decision Tree Regression(DTR), Support Vector Regression(SVR), Long Short Term Memory(LSTM) and Bi-directional Long Short Term Memory (BiLSTM). These predictions offer valuable insights into the anticipated network throughput trends and patterns within the cloud environments, aiding in the proactive management and optimization of network resources.



(a) Linear Regression model predicted vs real value

(b) Gradient Boosting Regression model predicted vs real value

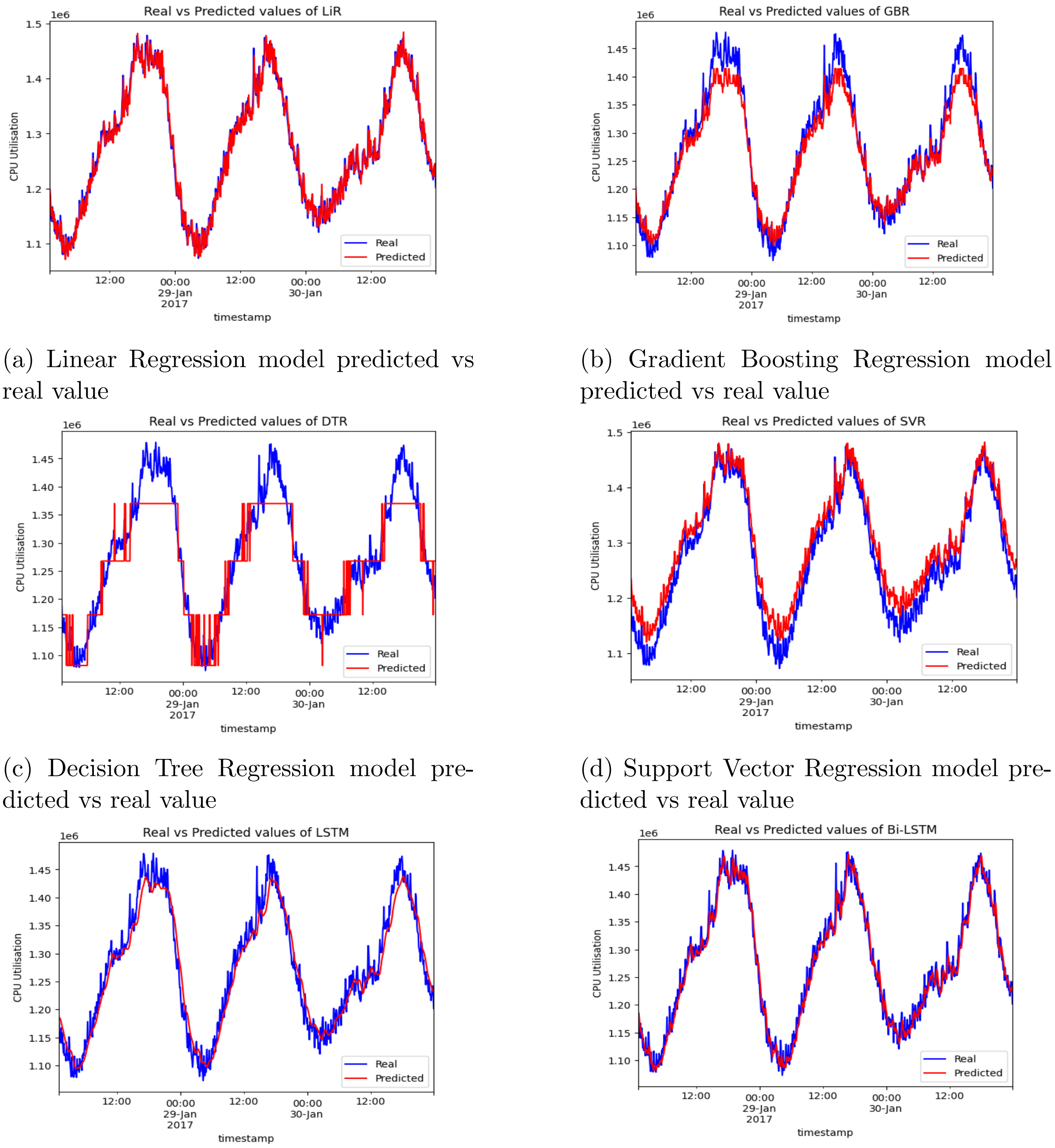
(c) Decision Tree Regression model predicted vs real value

(d) Support Vector Regression model predicted vs real value

(e) Long Short Term Memory model predicted vs real value

(f) Bi-directional Long Short Term Memory model predicted vs real value

Figure 10: Prediction of CPU Utilisation using BitBrains dataset
(Source: Generated using the Google Colab Notebook)



(a) Linear Regression model predicted vs real value

(b) Gradient Boosting Regression model predicted vs real value

(c) Decision Tree Regression model predicted vs real value

(d) Support Vector Regression model predicted vs real value

(e) Long Short Term Memory model predicted vs real value

(f) Bi-directional Long Short Term Memory model predicted vs real value

Figure 11: Prediction of CPU Utilisation using Microsoft Azure dataset (Source: Generated using the Google Colab Notebook)

6.4.2 Evaluation of Network Transmission Throughput

The evaluated results for network transmitted throughput are presented in Table 4 and Figure 13.From the results, it can be seen that BiLSTM has very close values when compared to the actual value. BiLSTM achieved higher network transmission throughput prediction accuracy than the compared models with 0.9 for R2 metrics and lower error rates for RMSE and MAE. Figure 13a shows same real and predicted values because the accuracy of linear regression with 0.92 for R2 metrics is higher than all the algorithms in Network Transmission Throughput. Both BiLSTM and Linear Regression are giving almost similar results. BiLSTM has advantages over Linear Regression as it can work with complex temporal patterns in the future. Also, it works better where data might exhibit non-linear patterns.

Table 4: For BitBrains dataset - Comparison of machine learning and deep learning algorithms for Network Transmission Throughput prediction

Model	MSE	MAE	RMSE	R2 score
LiR	0.0012	0.0114	0.0359	0.9256
DTR	0.0043	0.0473	0.0656	0.752
GBR	0.00183	0.0259	0.0428	0.894
SVR	0.0037	0.0495	0.0610	0.786
LSTM	0.0026	0.0232	0.0513	0.848
BiLSTM	0.00172	0.0203	0.0415	0.901

6.4.3 Comparative Analysis of Datasets Used for Predicting CPU Utilization

In comparing models for CPU Utilization predictions using BitBrains and Microsoft Azure datasets, the BiLSTM model consistently stood out as the most accurate in Figure 12. Compared to Linear Regression, LSTM, SVR, GBR, and DTR models, BiLSTM consistently demonstrated superior performance across both datasets. Its strength in capturing complex temporal dependencies allowed for more precise predictions of CPU Utilization dynamics. CPU Utilization for the BitBrains dataset, the performance value is quite different GBR: 0.5465 and LSTM: 0.7843. Same for the Azure dataset, GBR: 0.9399 and LSTM: 0.9462. There is a marginal difference in the R2 score. Both GBR and LSTM predict a R2 score for CPU utilization that is quite similar. Even yet, there is only about a 1% difference between the two. There are no intricate patterns in the collection, such as (Timestamp, CPU utilization). As a result, both algorithms may end up performing similarly by capturing the patterns in a similar way. It's also important to note that the outcomes for the same algorithms on the BitBrains dataset differ widely. While other models showed promise to varying degrees, none matched the robustness of BiLSTM in handling the intricacies within these datasets. This underlines the pivotal role of model architecture in effectively predicting CPU Utilization across diverse datasets.

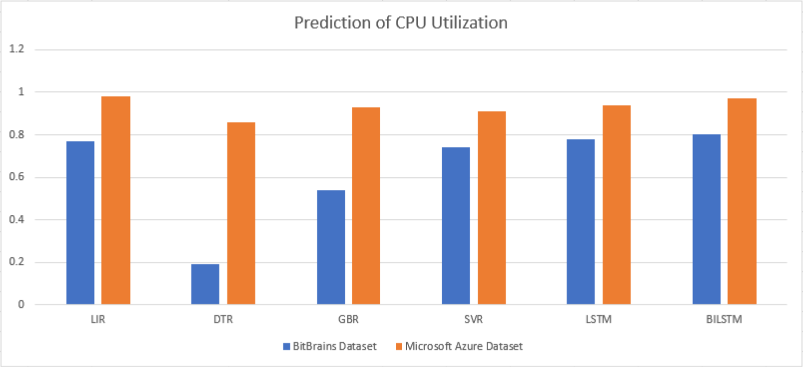
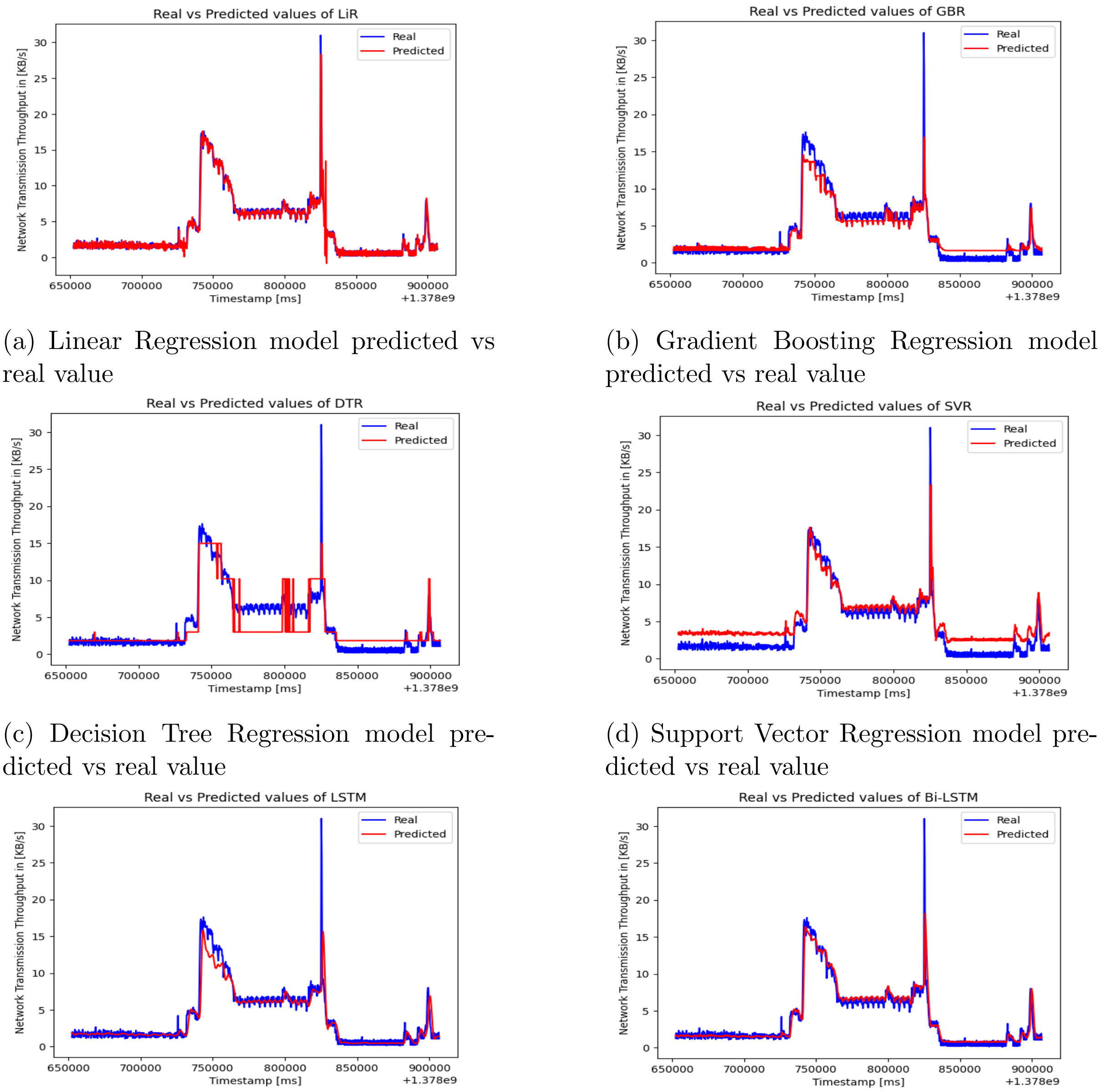


Figure 12: Comparison of Prediction of CPU Utilisation
(Source: Created by learner)



(a) Linear Regression model predicted vs real value

(b) Gradient Boosting Regression model predicted vs real value

(c) Decision Tree Regression model predicted vs real value

(d) Support Vector Regression model predicted vs real value

(e) Long Short Term Memory model predicted vs real value

(f) Bi-directional Long Short Term Memory model predicted vs real value

Figure 13: Prediction of Network Transmission Throughput using BitBrains dataset (Source: Generated using the Google Colab Notebook)

6.5 Discussion

This section encapsulates the methodologies employed in deploying four machine learning and two deep learning algorithms while showcasing their respective outcomes. The comprehensive analysis underscores the precision and robustness of the entire process, particularly highlighting the Bi-directional Long Short Term Memory (BiLSTM) model. Notably, the BiLSTM model displayed superior performance, as evidenced by its nearest-to-unity R-square value and the lowest Root Mean Square Error (RMSE) among all evaluated machine learning algorithms. This approach entails Python programming executed within Google Colab Notebook to predict resource utilization. The utilization of

BitBrains and Microsoft Azure datasets, encompassing CPU usage, network transmission throughput, and corresponding timestamps, enables the application of machine learning techniques such as Linear Regression, Decision Tree Regression, Gradient Boosting Regression, Support Vector Regression, and deep learning models including Long Short Term Memory and Bi-directional Long Short Term Memory.

7 Conclusion and Future Work

This study conducted a comprehensive exploration into predicting resource utilization within cloud computing frameworks through a diverse range of machine learning and deep learning models. Python programming within Google Colab was utilized alongside BitBrains and Microsoft Azure datasets, encompassing critical metrics such as CPU usage, network transmission throughput, and timestamps. The findings strongly emphasized the efficacy of the Bi-directional Long Short-Term Memory (BiLSTM) model, surpassing other machine learning algorithms in accuracy and performance. The achieved R-square values and Root Mean Square Error (RMSE) metrics highlight the BiLSTM model's exceptional predictive abilities in anticipating resource utilization, offering pivotal insights for optimizing cloud computing efficiency.

Based on this research, there are a number of interesting directions for further study. Prediction accuracy might be increased even more by investigating ensemble learning strategies to integrate different models. A more thorough grasp of resource usage patterns may be obtained by extending the dataset's reach outside BitBrains and Microsoft Azure. Further research into other real-time data aspects may improve prediction accuracy; nevertheless, improving the models' interpretability is still a crucial step towards gaining more profound understanding. Additionally, a critical first step towards the model's practical application in the real world is to assess its scalability and practical implementation in real-world cloud infrastructures under various workloads.

8 Video Presentation Demo

The URL is: `https://studentncirl-my.sharepoint.com/:v:/g/personal/x22174711_student_ncirl_ie/EXhxYSaF7QxIidJLwQ0xPE8Ba9yA409887h4zkvHOd29Lg?e=3vWvyZ`

References

Amiri, M. and Mohammad-Khanli, L. (2017). Survey on prediction models of applications for resources provisioning in cloud, *Journal of Network and Computer Applications* **82**: 93–113.

B R, S., K C, A. and Ramaiah, N. (2021). Resource utilization prediction in cloud computing using hybrid model, *International Journal of Advanced Computer Science and Applications* **12**: 2021.

Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K. and Hu, Y.-C. (2022). A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques, *Sensors* **22**(3).
URL: *https://www.mdpi.com/1424-8220/22/3/1242*

BiLSTM (2022). `https://stackoverflow.com/questions/76446615/can-bilstm-be-applied-to-timeseries`. Accessed: Month Day, Year.

Borkowski, M., Schulte, S. and Hochreiner, C. (2016). Predicting cloud resource utilization, *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, pp. 37–42.

Chugh, A. (2020). Mae, mse, rmse, coefficient of determination, adjusted r-squared: Which metric is better?, *Medium* . `http://tiny.cc/137ivz`.

Conforto, S., Zia Ullah, Q., Hassan, S. and Khan, G. M. (2017). Adaptive resource utilization prediction system for infrastructure as a service cloud, *Computational Intelligence and Neuroscience* **2017**: 4873459.

Duggan, M., Mason, K., Duggan, J., Howley, E. and Barrett, E. (2017). Predicting host cpu utilization in cloud computing using recurrent neural networks, *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 67–72.

duruo huang (2020). Schematic diagram of the gradient boosted regression tree. Image retrieved from ResearchGate.
URL: *https://www.researchgate.net/figure/Schematic-diagram-of-the-gradient-boosted-regression-tree*$_{f}ig2_{3}42270212$

Estrada, R., Valeriano, I. and Aizaga, X. (2023). Cpu usage prediction model: A simplified vm clustering approach, *Conference on Complex, Intelligent, and Software Intensive Systems*, Springer, pp. 210–221.

Farahnakian, F., Liljeberg, P. and Plosila, J. (2013). Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers, *2013 39th Euromicro conference on software engineering and advanced applications*, IEEE, pp. 357–364.

K, G. M. (2020). Machine learning basics: Decision tree regression. Web article retrieved from Towards Data Science.
URL: *https://towardsdatascience.com/machine-learning-basics-decision-tree-regression-1d73ea003fda*

Kaur, G., Bala, A. and Chana, I. (2019). An intelligent regressive ensemble approach for predicting resource usage in cloud computing, *Journal of Parallel and Distributed Computing* **123**: 1–12.

Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M. and Buyya, R. (2022). Machine learning (ml)-centric resource management in cloud computing: A review and future directions, *Journal of Network and Computer Applications* **204**: 103405.
URL: *https://www.sciencedirect.com/science/article/pii/S1084804522000649*

Khurana, S., Sharma, G. and Sharma, B. (2023). A fine tune hyper parameter gradient boosting model for cpu utilization prediction in cloud.

LSTM (2022). `https://databasecamp.de/en/ml/lstms`. Accessed: Month Day, Year.

Malik, S., Tahir, M., Sardaraz, M. and Alourani, A. (2022). A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques, *Applied Sciences* **12**(4): 2160.

Manam, S., Moessner, K. and Asuquo, P. (2023). A machine learning approach to resource management in cloud computing environments, *2023 IEEE AFRICON*, pp. 1–6.

Mason, K., Duggan, M., Barrett, E., Duggan, J. and Howley, E. (2018). Predicting host cpu utilization in the cloud using evolutionary neural networks, *Future Generation Computer Systems* **86**: 162–173.

Mehmood, T., Latif, S. and Malik, S. (2018). Prediction of cloud computing resource utilization, *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT (HONET-ICT)*, pp. 38–42.

Morariu, C., Morariu, O., Răileanu, S. and Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems, *Computers in Industry* **120**: 103244.
URL: *https://www.sciencedirect.com/science/article/pii/S0166361519311595*

Nääs Starberg, F. and Rooth, A. (2021). Predicting a business application's cloud server cpu utilization using the machine learning model lstm.

Rohit Daid, Yogesh Kumar, Y.-C. H. and Chen, W.-L. (2021). An effective scheduling in data centres for efficient cpu usage and service level agreement fulfilment using machine learning, *Connection Science* **33**(4): 954–974.
URL: *https://doi.org/10.1080/09540091.2021.1926929*

TensorFlow Keras Tutorial: Linear Regression (n.d.). Webpage/tutorial retrieved from LearnOpenCV.
URL: *https://learnopencv.com/tensorflow-keras-tutorial-linear-regression/*

Wang, J., Yan, Y. and Guo, J. (2016). Research on the prediction model of cpu utilization based on arima-bp neural network, *MATEC Web of Conferences*, Vol. 65, EDP Sciences, p. 03009.