International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI:10.55041/ISJEM05831
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Optimisation of Resource Utilisation in Cloud Computing Environments using Machine Learning

Dr. K. Satyam[1], V Harish[2]

[1]Associate Professor, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India.

[2] Post Graduate, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India.

Abstract:

This study demonstrates the significance of time in cloud computing scheduling, allocation, and intelligent resource utilisation. In order to estimate the runtime based on CPU utilisation, memory usage, disc speed, and network delay using MATLAB, this study compares two machine learning models: Support vector machine (SVM) and Decision tree (DT). While the DT (Decision tree) divides execution time into groups or bins (0-10 sec, 10-20 sec, etc.) and uses them to predict time, SVM attempts to anticipate the smallest feasible error between actual life and expected outcome.

There are two common tests to compare which model is superior. The first is the MEAN SQUARED ERROR (MSE), which indicates how accurate your forecasts are. Second, the r-squared approach indicates how well the model captures the variation in the data. The SVM outperforms DT by an astounding 81%, according to the results. As a result, SVM produced significantly superior predictions that were quite accurate, but DT produced predictions that were probably more inaccurate than SVM's. Additionally, it is thought that in the years to come, SVM may be combined with other cloud models to produce even more accurate forecasts and outcomes.

Keywords- Cloud computing; Machine learning; Support Vector Machine; Decision Tree; Prediction; Runtime; Network delay; Mean squared error.

I. Introduction

Cloud computing has become the foundation of modern distributed systems by enabling on-demand access to computational resources such as processing power, memory, storage, and networking. Task scheduling and resource allocation have become crucial issues as cloud platforms accommodate a variety of dynamic workloads. Accurately estimating the time needed to complete a task is one of the main elements impacting scheduling decisions. Performance degradation, service level agreement (SLA) violations, inefficient resource use, and higher operating costs can all result from inaccurate runtime estimation.
Conventional cloud scheduling methods distribute resources using historical averages or static threshold-based algorithms. These methods, however, are not flexible enough to accommodate changing workloads and different system setups.

Execution time cannot be reliably approximated using set analytical models since cloud environments function under changing CPU utilisation, memory consumption, disc throughput, and network delay. As a result, the demand for data-driven and intelligent prediction techniques is rising. Complex correlations between execution time and system performance characteristics can be effectively modelled using machine learning techniques. Predictive models can improve runtime estimation and facilitate proactive resource optimisation by identifying trends in past performance data. Regression and classification-based methods in particular have demonstrated promise in performance prediction and workload modelling.

In order to forecast task execution time based on system-level parameters including CPU usage, memory utilisation, disc speed, and network latency, two machine learning models—Support Vector Machine and Decision Tree—are constructed using MATLAB. While the Decision Tree model divides execution time into predetermined intervals to examine classification-based prediction performance, the SVM model uses regression to minimise prediction error between actual and estimated runtime. To assess the models' predictive power, Mean Squared Error (MSE) and R-squared (R^2) metrics are used. This study's main contribution is a comparison of machine learning techniques for cloud execution time prediction that rely on regression and classification. The results show that precise runtime estimation can greatly improve cloud performance optimisation and facilitate wise scheduling choices.

II. Problem Statement:

Multiple jobs vie for shared computational resources including CPU, memory, disc storage, and network bandwidth in cloud computing settings, which are extremely dynamic. Accurate task execution time estimation is essential for effective scheduling and resource allocation. However, because of workload fluctuation, varied infrastructure, and changeable performance factors, predicting execution time in cloud systems is a hard task. Conventional methods of allocating resources usually depend on fixed guidelines, preset cutoff points, or straightforward historical averages. Nonlinear correlations between runtime behaviour and system performance measures are not captured by these methods. Inaccurate runtime estimation can therefore result in a number of serious problems, such as overprovisioning of resources, underutilisation, longer response times, increased operational expenses, and service level agreement (SLA) violations.

Additionally, a number of interrelated parameters, like disc speed, CPU utilisation, memory utilisation, and network latency, affect execution time. These intricate relationships cannot be adequately described by traditional analytical methods. Cloud systems cannot effectively allocate resources or make proactive scheduling decisions without intelligent prediction tools. As a result, a data-driven strategy that can precisely forecast execution time and identify patterns from system-level performance measurements is required. In order to find the best approach for runtime prediction and performance optimisation in cloud computing settings, this study implements and compares two machine learning models: Support Vector Machine (SVM) and Decision Tree (DT).

III. Objective:

In order to enhance performance optimisation and resource allocation efficiency, the main goal of this project is to create and assess a machine learning-based framework for task execution time prediction in cloud computing environments.

The following are the study's particular goals:

1. To examine how task execution time in cloud systems is affected by system-level performance parameters as CPU utilisation, memory usage, disc throughput, and network latency.

2. Putting into practice a Support Vector Machine (SVM) regression model to forecast execution time with the least amount of prediction error possible.

3. To create a Decision Tree (DT) model for comparison analysis that divides execution time into predetermined intervals.

4. To use common performance metrics, such as Mean Squared Error (MSE) and R-squared (R^2), to assess and contrast the performance of SVM and Decision Tree models.

5. To determine the best machine learning strategy for enhancing the accuracy of runtime predictions and assisting with wise scheduling choices.

6. To illustrate how precise execution time prediction enhances system efficiency, minimises resource waste, and optimises cloud performance.

IV. Dataset Description

System-level performance indicators gathered from cloud computing environments make up the dataset used in this study. The characteristics of resource utilisation that have a direct impact on task execution time are represented by these metrics.

SVM_MSE	SVM_R2	DT_MSE	DT_R2
4866.732	-0.00151	10060.29	-1.07027

The input features considered in this study include:

- CPU utilization
- Memory usage
- Disk throughput
- Network latency

The target variable is:

Task execution time (in seconds)

Prior to model training, the dataset underwent preprocessing. Missing or inconsistent values were handled using data cleansing techniques. To evaluate the model, the dataset was split into training and testing subsets, and the feature variables were isolated from the target execution time variable. In order to facilitate predictive optimisation in cloud environments, the dataset aims to characterise the link between execution time and resource utilisation characteristics.

V. Performance Evaluation and Results

To evaluate the effectiveness of the proposed models, two standard performance metrics were used:

- Mean Squared Error (MSE)
- R-squared (R^2)

These metrics measure prediction accuracy and model explanatory power.

Support Vector Machine (SVM) Results

- MSE: 4866.732
- R^2: -0.00151

The SVM model outperformed the Decision Tree model in terms of prediction accuracy when calculating execution time, as evidenced by its lower Mean Squared Error. The model performs comparably better than the Decision Tree method, despite the R2 value being nearly zero.

Decision Tree (DT) Results

- MSE: 10060.29
- R^2: -1.07027

The Decision Tree model produced a significantly higher Mean Squared Error, indicating greater deviation between predicted and actual execution times. The negative R2 value indicates that the model performs worse than a straightforward baseline mean predictor and is unable to sufficiently explain the variance in execution time.

Comparative Analysis

When comparing both models:

SVM reduced prediction error by more than 50% compared to Decision Tree.

Decision Tree showed poor generalization capability.

SVM demonstrated better stability in runtime prediction.

The experimental findings show that Support Vector Machines, as opposed to Decision Tree classification-based modelling, are more suited for execution time prediction in cloud computing environments.

VI. Algorithms Used

Support Vector Machine (SVM) and Decision Tree (DT) are two machine learning methods used in this work to forecast task execution time in cloud computing settings. System-level performance data including CPU utilisation, memory usage, disc throughput, and network latency are used to train both models.

- **Support Vector Machine (SVM) for Regression**

A supervised learning technique for classification and regression issues is the Support Vector Machine. SVM is used in regression mode in this study to forecast values for continuous execution times. The basic idea behind SVM regression is to choose the best hyperplane for minimising prediction error while preserving model generalisation. In order to minimise total error, the algorithm attempts to fit a function that deviates from real target values by no more than a predetermined margin (ε).

The regression function can be expressed as:

f(x)=wTx+bf(x) = w^T x + bf(x)=wTx+b

Where:

xxx = input feature vector (CPU, memory, disk, network)

www = weight vector

bbb = bias term

The objective is to minimize:

12||w||2+C∑i=1nξi\frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i21||w||2+Ci=1∑nξi

Where:

CCC = regularization parameter

ξi\xi_iξi = error tolerance

||w||2||w||^2||w||2 ensures margin maximization

Why SVM Was Selected

• Effectively manages nonlinear interactions

• By using margin optimisation, overfitting is decreased.

• Does nicely in feature space with high dimensions.

• Fit for ongoing runtime forecasting

In this study, SVM outperformed the Decision Tree model in terms of execution time estimation due to its lower Mean Squared Error

Decision Tree (DT)

A supervised learning algorithm called Decision Tree uses decision rules that are built from input information to build a hierarchical tree structure.

Principle of Operation

Using feature thresholds, the Decision Tree divides the data into subsets. A decision rule is represented by each internal node, and a predicted output class or value is represented by each leaf node. The Decision Tree model in this study categorised activities into specified intervals based on execution duration, such as 0–10 and 10–20 seconds.

Usually, the splitting criteria reduces impurity by employing techniques like: In the case of regression trees, the mean squared error

• Gini Index (in the case of classification trees)

Reasons for Choosing a Decision Tree

• Simple to understand

• Minimal data preprocessing is needed.

Nonlinear relationships are captured.

• Quick training

Nonetheless, the Decision Tree model displayed negative R2 values and a larger prediction error, suggesting a limited capacity for generalisation in runtime prediction.

Algorithm Comparison

Model	Type	Target Output	Strength	Limitation
SVM	Regression	Continuous execution time	Lower error, better stability	Requires parameter tuning
Decision Tree	Classification	Time intervals	Simple & interpretable	Higher error, prone to overfitting

The experimental comparison reveals that SVM is more suitable for predicting execution time in dynamic cloud environments due to its better error minimization capability.

VII. Conclusion

In order to facilitate intelligent scheduling and performance optimisation, this study introduced a machine learning-based method for forecasting task execution times in cloud computing systems. System performance measures such as CPU utilisation, memory consumption, disc throughput, and network latency were used to develop and assess two models: Support Vector Machine (SVM) and Decision Tree (DT). In terms of prediction accuracy, the SVM model fared better than the Decision Tree model, according to experimental evaluation using Mean Squared Error (MSE) and R-squared (R^2). Significantly lower error levels were obtained with the SVM model, suggesting improved execution time estimation ability.

The comparison analysis unequivocally demonstrates that SVM is superior for runtime prediction in the provided dataset, despite the fact that both models showed poor explanatory ability as demonstrated by negative R2 values. Reducing performance degradation, enhancing resource allocation, and facilitating effective cloud scheduling all depend on accurate execution time estimation. Future research might concentrate on enhancing prediction accuracy via feature engineering,

hybrid machine learning models, sophisticated hyperparameter tweaking, or interaction with real-time cloud auto-scaling frameworks.

References

[1] M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
[2] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-Scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, vol. 12, no. 4, pp. 559–592, 2014.
[3] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, pp. 273–297, 1995.
[4] L. Breiman, "Classification and Regression Trees," Wadsworth International Group, 1984.
[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-Art and Research Challenges," Journal of Internet Services and Applications, 2010.
[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, 2008.