Machine Learning-Based Prediction Models for Efficient Cloud Resource Provisioning

Ms. Himgouri P. Barge[1], Dr. S. K. Yadav [2], Dr. Pravinkumar R. Badadapure[3]

[1]Research Scholar, Computer Engineering Department, Shri Jagdish Prasad Jhabarmal Tibrewala University, Vidyanagari, Jhunjhunu, Rajasthan, India

[2]Professor, Computer Engineering Department, Shri Jagdish Prasad Jhabarmal Tibrewala University, Vidyanagari, Jhunjhunu, Rajasthan, India

[3]Dean Academics and Administration, Pimpri Chinchwad Education Trust, Pune, Maharashtra, India

*** Corresponding Author:** Ms. Himgouri P. Barge,

Abstract -

To satisfy Service Level Agreement (SLA) constraints, cloud Virtual Machine (VM) resources must be allocated in advance due to inherent VM initialization delays. Accurate forecasting of future resource requirements therefore becomes essential for proactive provisioning. In this study, predictive models for cloud clients were designed and experimentally evaluated using the TPC-W benchmark web application. Three machine learning techniques—Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Linear Regression (LR)—were employed to estimate forthcoming resource demands. SLA-related performance indicators, specifically response time and throughput, were integrated into the prediction framework to enable more informed and reliable scaling decisions. Experimental results indicate that the Support Vector Machine–based model consistently delivers superior prediction accuracy compared to the other approaches. Furthermore, the proposed framework demonstrates improved compliance with SLA thresholds while reducing unnecessary resource over-provisioning. The findings highlight the effectiveness of predictive analytics in enhancing cloud resource utilization efficiency. This approach can be readily extended to support dynamic auto-scaling mechanisms in real-world cloud environments.

Keywords: Cloud Computing, Virtual Machine (VM) Provisioning, Service Level Agreement (SLA), Predictive Resource Allocation, Machine Learning, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Linear Regression (LR), TPC-W Benchmark, Response Time, Throughput, Auto-Scaling, Cloud Resource Management.

I. INTRODUCTION

Efficient provisioning of virtual machines (VMs) remains a critical and challenging area of research. VM initialization times have been observed to vary significantly, ranging from approximately 5 to 10 minutes in some studies, and from 5 to 15 minutes in others. This start-up latency can result in penalties for cloud service providers, especially when multiplied across numerous server deployments within a data center, leading to substantial cumulative costs. One effective strategy to address this issue is predictive resource allocation, where anticipating future demands for CPU, memory, network, and disk I/O can guide proactive scaling decisions, thereby mitigating the impact of VM boot-up delays. Various predictive techniques have been proposed, and this study focuses on evaluating three specific approaches: Neural Networks (NN), Linear Regression (LR), and Support Vector Machines (SVM). Unlike traditional resource-focused predictions, we incorporate business-level Service Level Agreement (SLA) metrics—namely throughput and response time—into the prediction models, providing more comprehensive guidance for scaling decisions.

This work is framed within the Infrastructure-as-a-Service (IaaS) model, which offers developers greater flexibility in programming language choice compared to Platform-as-a-Service (PaaS) providers, which typically impose constraints through platform-specific languages such as Java or Ruby on Rails.

The contributions of this study are twofold. First, it evaluates the accuracy of resource usage predictions using SVM, NN, and LR across three TPC-W benchmark workloads. Second, it extends these prediction models to integrate business-level SLA metrics, thereby enabling more informed and effective resource scaling strategies for cloud clients. Numerous studies have explored resource usage prediction in cloud environments. For instance, one study proposed an algorithm that leveraged historical usage data to identify recurring patterns and forecast future resource demands. While this approach reported prediction errors ranging from 0.9% to 4.8%, the specific metric used to calculate these errors was not disclosed, and the predictions were limited to a short horizon of only 100 seconds. Another study examined both static and dynamic resource provisioning under three traffic patterns—weekly oscillations, large spikes, and random fluctuations—using a scoring algorithm based on availability and cost. In a different approach, researchers focused on estimating the resource requirements of applications in virtual environments by analyzing utilization traces from their native environments. They applied a trace-based method to predict future CPU usage for capacity planning, assuming a strong correlation between native and virtual environments. Their study incorporated diverse workload mixes reflective of typical enterprise applications and achieved prediction errors below 5% in the 90th percentile. However, the model considered only CPU usage and could not account for response time. As highlighted in prior work, relying solely on CPU utilization for scaling decisions can be misleading, since high CPU usage might result from insufficient memory or disk I/O. Subsequent research addressed this limitation by including memory and I/O metrics alongside CPU usage in predictive models.

Building on these insights, our study examines resource provisioning from the cloud client's perspective, emphasizing the ability of hosted applications to make informed scaling decisions. Unlike previous work, our approach integrates both predicted resource utilization and business-level SLA metrics—specifically response time and throughput—creating a comprehensive three-dimensional auto-scaling decision framework. To achieve this, we employ machine learning techniques, including Neural Networks, Linear Regression, and Support Vector Machines. The performance of these models is evaluated using established metrics: Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and PRED(25).

II. RELATED WORK

Resource provisioning in cloud computing has been widely studied due to its critical role in maintaining Service Level Agreements (SLAs) while optimizing operational costs. One of the main challenges in cloud resource management is the latency associated with virtual machine (VM) initialization, which can range from 5 to 15 minutes depending on the system and workload [6]–[10]. Such start-up delays may lead to SLA violations, especially when multiple VMs are instantiated simultaneously, resulting in substantial cumulative penalties for cloud service providers. Predictive approaches, which forecast future resource demands, have been proposed to mitigate this problem by enabling proactive scaling decisions that compensate for VM boot-up latency [7]. Several studies have explored resource usage prediction using historical utilization data. In [7], a resource prediction algorithm identified patterns from past usage to forecast short-term resource demands. Although the study reported low prediction errors ranging from 0.9% to 4.8%, the evaluation metrics were not explicitly stated, and predictions were limited to a short horizon of 100 seconds. Similarly, the work in [10] investigated static and dynamic provisioning strategies across different traffic patterns, including periodic oscillations, large spikes, and random workloads, using a scoring algorithm based on availability and cost.

Trace-based prediction methods have also been employed, where resource utilization in an application's native environment is analyzed to estimate requirements in a virtualized environment [12]. By leveraging correlations between native and virtual environments, CPU usage forecasts were obtained for capacity planning. These methods considered various workload mixes representative of enterprise applications and achieved prediction errors below 5% in the 90th percentile. However, such models generally focused on CPU utilization alone, excluding other critical metrics such as memory, disk I/O, and response time. As noted in [13], relying solely on CPU usage for scaling decisions can be misleading, since increased CPU utilization may result from memory or I/O bottlenecks. Subsequent work addressed these limitations by including memory and I/O metrics in the predictive models [14]. While prior studies have primarily focused on resource-centric prediction, they often neglect business-level SLA metrics such as throughput and response time. This limitation reduces the effectiveness of auto-scaling decisions from the client's perspective. To address this gap, this study proposes a predictive framework that integrates both resource utilization (CPU, memory, network, disk I/O) and business-level SLA metrics. Three machine learning techniques—Neural Networks (NN), Linear Regression (LR), and Support Vector Machines (SVM)—are employed to generate accurate forecasts. Model performance is evaluated using Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and PRED(25). By combining resource usage prediction with SLA considerations, the proposed framework provides a comprehensive auto-scaling decision matrix. This enables cloud clients to make informed scaling decisions that balance performance, cost, and compliance with SLA requirements, aligning with recent trends in predictive and intelligent cloud resource management.

III. RESEARCH METHODOLOFY

The methodology adopted in this study is designed to evaluate and compare the effectiveness of three machine learning techniques—Linear Regression (LR), Neural Networks (NN), and Support Vector Machines (SVM)—for predicting cloud resource usage and supporting SLA-driven auto-scaling decisions. The overall workflow of the research methodology is depicted in Figure 1.

Data Collection and Preprocessing

The study utilizes performance traces from a multi-tier web application benchmarked using TPC-W. Resource usage metrics including CPU, memory, network bandwidth, and disk I/O are collected over different workload patterns to capture the dynamic behavior of cloud applications. Business-level SLA metrics, specifically response time and throughput, are also recorded. The collected data is cleaned to remove anomalies and missing values and normalized to ensure comparability across different scales. Finally, the dataset is split into training (70%) and testing (30%) sets for model evaluation.

Linear Regression (LR)

Linear Regression is employed as a baseline predictive model due to its simplicity and interpretability. In LR, the target resource utilization metric is modeled as a linear combination of input features (CPU, memory, network, disk I/O, and SLA parameters) with weights learned from the training data. The predicted value $\hat{y}$is obtained as:

$$\hat{y} = w_0 + w_1 a_1 + w_2 a_2 + \cdots + w_k a_k$$

where $a_1, a_2, \ldots, a_k$are input attributes, $w_0, w_1, \ldots, w_k$are the corresponding weights, and $\hat{y}$represents the predicted output. The model's parameters are estimated using the least-squares method to minimize the prediction error over the training set.

Neural Networks (NN)

Neural Networks are utilized to capture complex nonlinear relationships and temporal dependencies in resource usage data. Specifically, a Multilayer Perceptron (MLP) is implemented, which consists of an input layer, one or more hidden layers, and an output layer. Each neuron applies a nonlinear activation function to its weighted inputs and propagates the result to subsequent layers. The network is trained using backpropagation to minimize prediction error. NN is particularly suitable for handling noisy or partially observed data and can model interactions between resource metrics and SLA parameters that may not be captured by linear methods.

Support Vector Machines (SVM)

Support Vector Machines are employed to perform regression using the Support Vector Regression (SVR) framework. SVR aims to identify a function that approximates the target variable within a specified margin of error while maintaining maximum flatness to avoid overfitting. The model relies on structural risk minimization, making it robust against local minima and ensuring generalization to unseen workloads. SVM is trained on the input features (resource and SLA metrics) and evaluated for its ability to predict future resource usage accurately.

Model Evaluation

The performance of the three predictive models is evaluated using standard metrics:

- Mean Absolute Percentage Error (MAPE): Measures the average absolute percentage deviation between predicted and actual values.

- Root Mean Square Error (RMSE): Provides a measure of the magnitude of prediction errors.
- PRED(25): Indicates the percentage of predictions within 25% of the actual value.

These metrics provide a comprehensive assessment of prediction accuracy and reliability.

IV. EXPERIMENTAL SETUP AND DATA COLLECTION

The experimental evaluation was conducted using a virtualized testbed comprising a web server and a database server. The web server was hosted on a Linux-based virtual machine configured with a single processor, 1 GB RAM, a 100 MB/s network interface card, and a 10 GB disk. The database server operated on a Windows platform running MySQL and was provisioned with a dual-core AMD Athlon 64 processor (5000+), 4 GB RAM, a 100 MB/s network interface, and a 230 GB storage disk. The overall experimental methodology was organized into multiple phases, including feature selection, historical data acquisition, feature reduction, data preprocessing through normalization and scaling, followed by training and testing of the predictive models using the processed dataset. Initially, a comprehensive set of system-level and application-level performance metrics was collected to serve as input features. These included CPU utilization, response time, throughput measured as write transactions per second, total packets received and transmitted per second, available free memory, used memory expressed in kilobytes, system load averages over one-minute and five-minute intervals, page availability, context switches per second, run queue length, page-in and page-out disk operations, number of active tasks, read transactions per second, and the proportion of utilized swap space.

Data Collection Using the TPC-W Benchmark

To generate representative web application workloads, the TPC-W benchmark was employed. TPC-W is a standardized benchmark designed to model the behavior of e-commerce applications by simulating a web server and backend database environment under realistic operational conditions. Due to its widespread adoption in cloud resource provisioning and capacity planning studies, TPC-W provides a reliable foundation for performance evaluation. In this study, a Java-based implementation of the TPC-W benchmark was deployed to emulate an online bookstore application within a three-tier architecture consisting of client emulators, a web server, and a database server. System-level resource utilization metrics such as CPU usage and memory consumption were collected from the server side, while business-level SLA metrics including response time and throughput were measured from the client perspective. The benchmark's remote browser emulator (RBE) was used to simulate multiple concurrent clients, with the client side chosen as the reference point for performance evaluation.TPC-W supports three distinct workload profiles—Browsing, Shopping, and Ordering—each reflecting different user interaction behaviors. By varying the number of emulated clients between 100 and 1000 and combining linear and random request generation patterns, a dynamically changing workload was produced. This approach ensured continuous and realistic request traffic to the web server throughout the experiment.

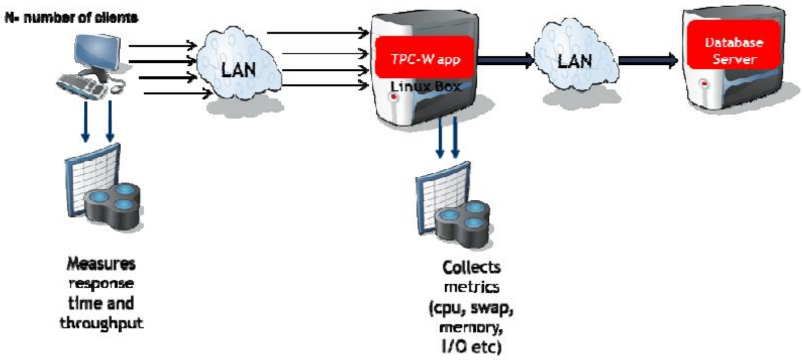

Fig.1 Model implementation architecture

System performance data were collected at 60-second intervals using the sysstat monitoring utilities available in Ubuntu. The total duration of the experimental run was approximately 170 minutes, yielding a sufficiently large dataset for training and validating the prediction models. The collected data were subsequently used to construct predictive models capable of forecasting future resource requirements and SLA performance of the web server.

To reduce dimensionality and improve model efficiency, feature selection was performed using the WEKA machine learning toolkit. Attribute relevance was assessed with respect to the target variables—CPU utilization, response time, and throughput. Features exhibiting minimal correlation, such as read transactions per second, run queue length, page-in operations, swap space usage, and the number of active tasks, were excluded from the final dataset. During preprocessing, the remaining input features were normalized to a uniform range between 0 and 1. This normalization step was essential to prevent attributes with larger numeric scales from disproportionately influencing the learning process, thereby enhancing prediction stability and accuracy across all machine learning models.

Training of the Dataset

The training phase was conducted by applying three machine learning techniques—Neural Networks (NN), Linear Regression (LR), and Support Vector Regression (SVR)—to the preprocessed dataset. Initially, CPU utilization was selected as the target variable, and a predictive model was trained using the selected input features. Subsequently, a separate predictive model was developed in which response time and throughput were jointly considered as target variables.

To evaluate the effectiveness of the learning process, three standard performance metrics were employed: Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and PRED(25), which represents the percentage of predictions whose error falls within 25% of the actual observed value. These metrics were used to assess both training accuracy and generalization capability during testing.

Accordingly, two distinct models were constructed. **Model 1** focused on predicting CPU utilization, while **Model 2** targeted business-level SLA metrics, namely response time and throughput. The WEKA machine learning environment was used to implement NN, LR, and SVR for both models. For consistency and fair comparison, identical parameter configurations were applied across response time and throughput predictions within Model 2.

Testing of the Trained Model

To validate the predictive performance of the trained models, the dataset was divided into training and testing subsets using a 60:40 ratio. This partitioning strategy was selected based on experimental trials, as it yielded the most reliable prediction accuracy among the tested configurations.

The trained models were evaluated using a prediction horizon of 12 minutes, which was chosen in accordance with findings reported in earlier studies related to virtual machine initialization delays. This prediction interval aligns with documented VM boot-up times and provides sufficient lead time for proactive scaling decisions. By adopting this forecasting window, the models were tested for their ability to anticipate future resource demands and SLA performance metrics in advance, thereby supporting timely and effective auto-scaling actions.

V. RESULTS AND DISCUSSION

The primary objectives of this research are to assess the prediction accuracy of selected machine learning techniques for future resource utilization and to incorporate business-level Service Level Agreement (SLA) metrics into the forecasting framework. To achieve these objectives, two predictive models were developed: one focusing on CPU utilization (Model 1) and the other targeting SLA-related performance indicators, namely response time and throughput (Model 2).

The performance results obtained during training and testing for the CPU utilization model are presented in Tables 3 and 4, respectively. Additionally, the Mean Absolute Percentage Error (MAPE) for step-ahead predictions covering the 9–12 minute interval is reported to evaluate short-term forecasting accuracy. For the SLA prediction model, Tables 6 and 8 summarize the evaluation metrics for response time and throughput on the test dataset, while Table 7 presents the corresponding multi-step prediction results for the same interval.

The observed CPU utilization patterns, illustrated in Figures 3, 4, and 5, exhibit highly variable and non-deterministic behavior. This variation reflects a realistic operational scenario in which user requests arrive and terminate randomly, resulting in frequent fluctuations in system load. Such dynamic user interactions lead to sudden increases and decreases in CPU usage, making accurate prediction particularly challenging.

Analysis of the training results indicates that Support Vector Regression (SVR) consistently outperforms Neural Networks (NN) and Linear Regression (LR) in terms of MAPE and PRED(25) metrics, as shown in Table 3. Furthermore, the close similarity between training and testing MAPE values demonstrates the strong generalization capability of the SVR model. The test dataset results further confirm the superiority of SVR, which achieves the most stable and accurate predictions among the evaluated techniques. In contrast, the NN model exhibits comparatively poor performance, with prediction errors exceeding those of LR in several instances.

Table 1: CPU Utilization Training Performance Metric

Model	MAPE	RMSE	PRED(25)
SVR	16.15	6.75	0.77
NN	26.18	8.68	0.60
LR	18.07	6.72	0.74

Table 2: CPU Utilization Test Performance Metric

Model	MAPE	RMSE	PRED(25)
SVR	16.84	12.21	0.84
NN	40.86	28.45	0.25
LR	22.01	16.18	0.68

Table 3: CPU Utilization Step Prediction for MAPE

Model	9-min	10-min	11-min	12-min
SVR	16.41	16.86	16.72	16.84
NN	42.67	30.20	33.02	40.86
LR	19.42	21.16	21.45	22.01

Table 4: Response Time Test Dataset Performance Metric

Model	MAPE	RMSE	PRED(25)
SVR	14.17	1.923	0.893
NN	13.35	1.742	0.911
LR	14.30	2.000	0.911

Table 5: Response Time Step Prediction for MAPE

Model	9-min	10-min	11-min	12-min
SVR	14.17	14.32	14.33	14.17
NN	16.02	16.30	16.09	13.35

LR	13.48	13.87	13.56	14.30

Figures 2, 3, and 4 illustrate the actual versus predicted CPU utilization for LR, NN, and SVR, respectively. SVR demonstrates superior responsiveness to nonlinear and random request patterns, capturing rapid changes in system load more effectively than the other models. This characteristic contributes to its consistent prediction performance across varying workloads. The NN model, however, displays notable instability in multi-step forecasting. As observed in Table 3, NN predictions fluctuate significantly between successive time intervals, including a sharp variation between the 11th and 12th minute, corresponding to a prediction difference of approximately 24%. This erratic behavior can be attributed to the sensitivity of NN models to noisy and highly dynamic input patterns.

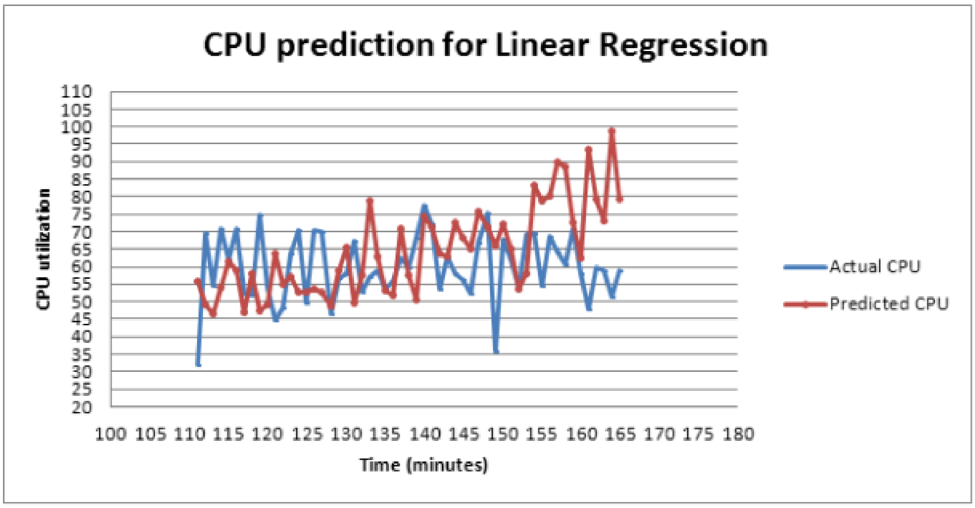

Fig. 2 Actual and Predicted CPU utilization - Linear Regression

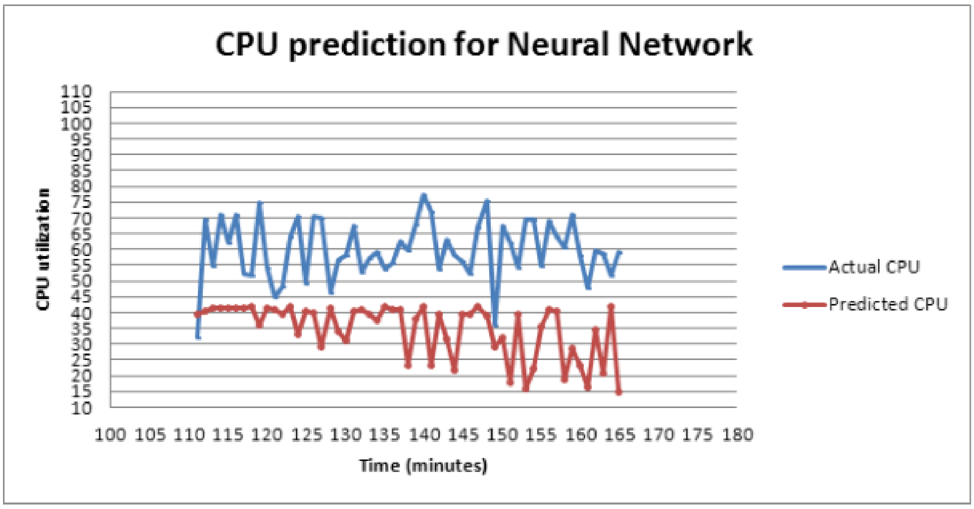

Fig. 3 Actual and Predicted CPU utilization – Neural Network

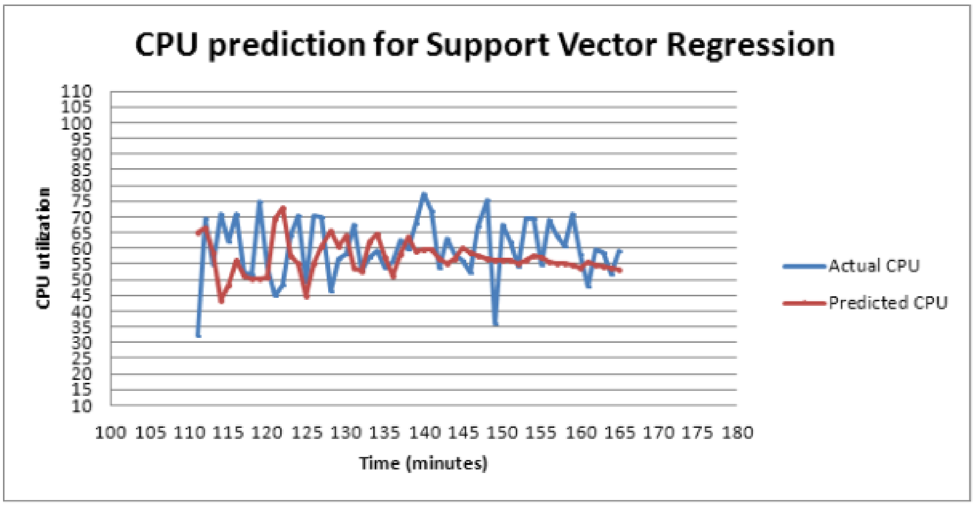

Fig. 4 Actual and Predicted CPU utilization – Support Vector Regression

When compared with earlier studies that assumed linear user request patterns, the present work highlights the challenges posed by random and realistic workload conditions. Unlike those studies, which reported smoother prediction trends, the random request generation adopted here exposes the limitations of NN-based approaches under highly variable conditions. LR demonstrates moderate stability and ranks second after SVR in terms of prediction consistency for CPU utilization.

For the response time prediction model, NN achieves the lowest overall prediction error, with SVR performing closely behind, as shown in Table 4. However, a detailed examination of the step-ahead predictions between the 9th and 12th minute (Table 5) reveals that NN continues to suffer from inconsistency, particularly between the 11th and 12th minute, where a deviation of approximately 21% is observed. In contrast, LR exhibits comparatively stable behavior across the 9th to 11th minute prediction horizon and, in some cases, marginally outperforms SVR in short-term prediction accuracy.

Corresponding plots for NN and LR are provided in Appendix A (Figures A1 and A2), offering further insight into model behavior. Overall, the results indicate that while NN can achieve high accuracy under certain conditions, its instability in multi-step forecasting limits its reliability. SVR offers a more balanced trade-off between accuracy and robustness, making it particularly suitable for SLA-aware predictive auto-scaling in dynamic cloud environments.

VI. CONCLUSION AND FUTURE WORK

This study presented a comparative evaluation of three predictive modeling techniques—Linear Regression, Neural Networks, and Support Vector Regression—applied to a two-tier TPC-W web application. Unlike conventional approaches that rely solely on CPU utilization for forecasting, the proposed framework integrates business-level Service Level Agreement (SLA) indicators, namely response time and throughput, into the prediction process. This integration enables a more comprehensive assessment of application performance and resource requirements.

Experimental results demonstrate that the Support Vector Regression (SVR) model consistently achieves higher prediction accuracy than both Neural Networks and Linear Regression within a forecasting horizon of 9 to 12 minutes. Among the evaluated techniques, SVR exhibits superior generalization capability and robustness when handling nonlinear and highly dynamic workload patterns. As a result, SVR emerges as the most effective approach for proactive resource demand forecasting, making it particularly suitable for cloud clients seeking reliable predictive models.

The inclusion of SLA metrics alongside resource utilization data enables a three-dimensional decision framework for adaptive virtual machine (VM) scaling. This approach is especially valuable because performance degradation in response time and throughput can occur well before CPU utilization reaches predefined thresholds. By accounting for these early warning indicators, cloud clients can initiate timely scaling actions and reduce the likelihood of SLA violations.

Future work will focus on deploying and validating the proposed prediction framework within a public cloud environment to assess its scalability and real-world applicability. Additionally, extending the model to incorporate database server behavior represents a promising direction for further research. Such an extension would allow the prediction of scenarios involving unsaturated web servers with overloaded databases, as well as simultaneous saturation of both tiers, enabling proactive provisioning strategies before SLA penalties are incurred.

REFERENCES

1. D. Hiley, "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings", in "Scholarly Materials and Research at Tech", Georgia Tech Library. April, 2009
2. "Amazon elastic compute cloud (amazon ec2)", 2012. [Online]. Available: http://aws.amazon.com/ec2
3. "Google AppEngine", 2012. [Online]. Available: https://developers.google.com/appengine
4. "Salesforce", 2012. [Online]. Available: http://www.salesforce.com
5. W. Iqbal et al., "Adaptive resource provisioning for read intensive multi-tier applications in the cloud" in Future Generation Computer Systems, vol. 27, pp. 871 – 879,
6. S. Islam et al., "Empirical prediction models for adaptive resource provisioning in the cloud", Future Generation Computer Systems, vol. 28, no. 1, pp 155 – 165, January, 2012
7. E. Caron, F. Desprez and A. Muresan, "Forecasting for Grid and Cloud Computing On-Demand Resources Based on Pattern Matching" in Cloud Computing Technology and Science (CloudCom), in 2010 IEEE Second International Conference , vol., no., pp.456-463, November, 30 2010
8. A. Quiroz et al., "Towards autonomic workload provisioning for enterprise Grids and clouds" in Grid Computing, 2009 10th IEEE/ACM International Conference pp 50-57, October, 2009
9. J. Kupferman et al., Scaling Into the Cloud TPC, TPC-W Benchmark, Transaction Processing Performance Council (TPC), San Francisco, CA, USA, 2003
10. T. Wood et al., "Profiling and Modeling Resource Usage of Virtualized Applications" in Proc. Of ACM/IFIP/USENIX Middleware, 2008
11. Kundu, S.; Rangaswami, R.; Dutta, K.; Zhao, M.; , "Application performance modeling in a virtualized environment," High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on , vol., no., pp.1-10, 9-14 Jan. 2010
12. S. Kundu et al., "Modeling virtualized applications using machine learning techniques", in Proc. Of 8th ACM SIGPLAN/Sigpos conference on Virtual Execution Environments, pp3 – 14, London, UK 2012
13. A. Khashman and N.I Nwulu, "Intelligent prediction of crude oil price using Support Vector Machines," Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium, vol., no., pp.165-169, 27-29 Jan. 2011
14. G.E Sakr et al., "Artificial intelligence for forest fire prediction," in Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference, vol., no., pp.1311-1316, 6-9 July 2010
15. P. SangitaB, and S.R Deshmukh, "Use of support vector machine for wind speed prediction," in Power and Energy Systems (ICPS), 2011 International Conference, pp.1-8, 22-24 Dec. 2011
16. I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Academic Press, USA, 2000
17. S. Samarasinghe, Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition. Auerbach Publications, USA, 2007
18. H. Guosheng et al., "Grid Resources Prediction with Support Vector Regression and Particle Swarm Optimization," Computational Science and Optimization (CSO), 2010 Third International Joint Conference, vol.1, pp.417-422, 28-31 May 2010
19. A.J Smola and B. Scholkopf, "A tutorial on support vector regression" in Statistics and Computing vol 14, pp. 199 – 222, 2004
20. "Comp-activ dataset" 2012. [Online]. Available: http://www.cs.toronto.edu/~delve/data/comp- activ/compActivDetail.html