Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 2 (2025)

AI-Driven Resource Provisioning: Enhancing Elasticity and Efficiency with Hybrid RNN-LSTM Models

P. Christopher[1], Dr. R Lawrance[2]

[1]Research Scholar, Department of Computer Science, Bharathiyar University, Coimbatore, Tamilnadu, online.christ@gmail.com

[2]Director, Department of Computer Applications, ANJA College, Sivakasi, Tamilnadu, lawranceraj@gmail.com

Article History:

Received: *25-07-2024*

Revised: *07-09-2024*

Accepted: *15-09-2024*

Abstract

Automatic resource provisioning techniques that dynamically modify resources in response to service demand are essential to implementing elasticity services. For systems with strict latency or reaction time requirements, such as Enterprise Resource Planning (ERP) systems under high traffic loads, this flexibility is crucial for lowering power consumption and guaranteeing quality of service (QoS). Determining the best point at which to scale resources is still a difficult task. In this research, we provide an AI-integrated system that adjusts resources according to anticipated demand. To predict load requests, the system uses an LSTM model and a Hybrid Recurrent Neural Network (RNN). This strategy seeks to minimize overprovisioning, which will lower the cost of infrastructure and energy usage. The deep learning model aims to estimate the resources required to improve service response time and satisfy customer requests, as well as to anticipate with high accuracy the processing load of distributed servers. Proactive provisioning decisions are made for the servers based on the anticipated load. Our tests on a common server request dataset show that the suggested RNN+LSTM model performs better than traditional deep learning models in terms of efficient resource management and prediction accuracy.

Keywords: Elasticity Services, Automatic Resource Provisioning, Hybrid RNN+LSTM, Load Prediction, Quality of Service (QoS).

1. Introduction:

Industry acceptance of cloud computing can be attributed to its pay-as-you-go approach for on-demand resource delivery. Cloud computing typically includes three main service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). In particular, SaaS offers full application access as a service. PaaS offers an application development platform that may be used to create other apps like Azure and Google App Engine (GAE). An environment for deploying managed virtual machines is offered by IaaS. In theory, the providers would supply the resources based on the users' demand when consumers submit their requests. Elasticity is a fundamental approach in cloud computing that allows resources to be acquired and released based on user demand.

1.1 Model of cloud layer:

This section outlines our method's cloud layer model for quickly scaling the resources. While the Delphi method relies heavily on subjective judgment, the cloud layer model places more emphasis on quantitative analysis. The cloud layer model is used to apply the ERP technique. SaaS, PaaS, and IaaS are the three components that make up the layer model. The number of user requests is determined by the SaaS. The broker in a PaaS is in charge of allocating infrastructure resources based on user demand, as seen by the MAPPE loop. The data center in IaaS is made up of a few PMs and VMs. [1] the resources would be supplied by the provider by the requests. Following a detailed description, the major elements of the MAPE are illustrated in Fig 1.1.

Monitor (M): Metrics like CPU and memory usage as well as certain available resource utilization are gathered by the monitoring component. It keeps an eye on the data every five seconds. The performance model, which is thoroughly explained in the following section, gathers, aggregates, and computes the important data.

Analyze (A): The information gathered must be analyzed during the analyzing step. The performance model aggregates and computes the collected data, and we determine whether the expansion action can be initiated by achieving the performance value. Additionally, we shut down the spare machines and used the WMA prediction method to figure out the correct amount of servers.

Plan (P): The central element of a cloud layer structure is this component. It achieves the scaling technique by lowering energy usage and renting prices by customer demand. Furthermore, the resources would be adjusted based on the performance criterion.

Execute (E): By configuring the infrastructure's servers, the Nginx load-balanced server distributes the web requests during the executing phase. Because the VMs are maintained in the PMs, the provider would use the suggested plan to provision the resources based on demand.

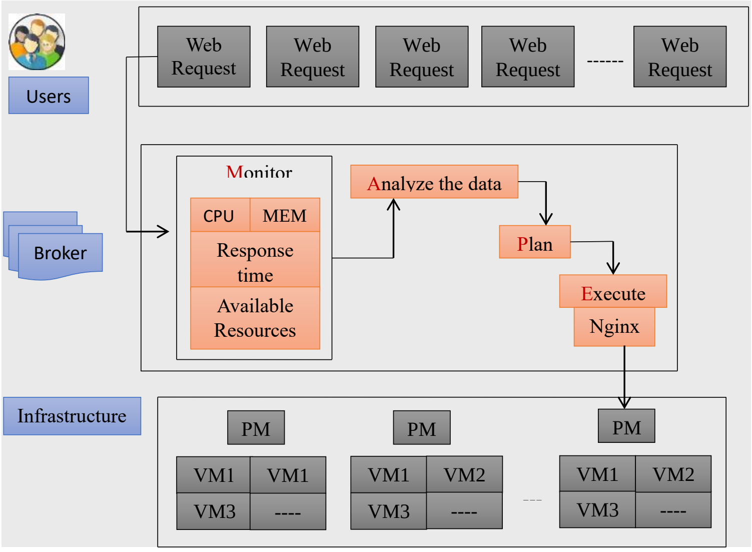

Figure 1.1: Resource Allocation Process

1.2 Elasticity Challenges:

Latency for quasi-immediate scaling: Resizing the resource capacity on-demand takes several minutes in the cloud; in actuality, resource allocation and de-allocation are not instantaneous. Applications requiring real-time and mission-critical functionality that operate at the tiniest timescales of milliseconds or microseconds might not be supported by this non-negligible scaling latency. However, existing elasticity isn't developed enough to provide suitable QoS for these kinds of applications. [2] It is necessary to provide more advanced adaptive scaling techniques to ensure high elasticity for such applications.

Maximum scaling factor limitation: One of the benefits of cloud computing is the ability to obtain almost limitless resource capacity from an elastic service in the cloud whenever needed. However, there is a finite amount that is limited by a default value that represents the maximum feasible capacity of resources that a customer may rent at any one time.

Variable elasticity behaviour: An adaptive cloud system's elasticity is a random variable. Even with an identical experimental setup, the perceived elasticity may fluctuate in an unforeseen way between runs with the same task. The erratic behaviour of cloud elasticity can be attributed to multiple environmental conditions. Better provisioning techniques could also be used by cloud providers to reduce the unpredictable nature of the elasticity of the cloud-based services that are offered.

1.3 Overview of AI-Driven Resource management:

The widespread use of Internet of Things (IoT) devices is one of the main factors enabling AI-driven resource management. The abundance of data produced by these networked gadgets may be used to learn more about how users behave, what's going on around them, and how well apps work. Cloud providers can improve their resource management capabilities and make proactive decisions and predictive resource provisioning through the integration of IoT data streams with their systems. [3] Additionally, by moving processing and data storage farther from the point of use, edge computing has changed the cloud computing landscape. By processing data locally, edge devices such as detectors, pathways, and edge servers reduce latency and bandwidth consumption while improving scalability and dependability. Cloud providers can use edge-to-cloud intelligence to optimize resource allocation across multiple clouds, reducing network congestion and enhancing overall system performance, by expanding AI-driven resource management across the edge. The amalgamation of AI-powered resource management, Internet of Things gadgets, and edge computing signifies a revolutionary development in cloud computing, permitting flexible and astute resource distribution tactics.

1.3.1 Artificial Intelligence for Energy Cloud Management:

Artificial Intelligence is becoming indispensable in the energy field and holds enormous promise for the design of future energy systems. Thus, artificial intelligence (AI) emerges as an emerging technology that has the power to revolutionize any industry. Terabytes or even petabytes of heterogeneous data, whether organized or unstructured, can be handled by AI algorithms. It can use the patterns found in the data to forecast or suggest actions. AI has exploded in the energy business, and every company is embracing it to properly balance supply and demand for energy. Through the use of SM deployed at each user's location, [4] AI algorithms assist in managing the supply and demand for energy, forecasting consumer energy needs, and advising consumers on energy-saving consumption practices. By 2024, it is expected that the global market for AI in energy administration will have grown from $4,439.1 million to $12,200.9 million. AI boosts system efficiency, predicts energy flow, enhances SG stability, and analyses vast volumes of data. Around the world, a large number of firms have been launched that use AI to manage energy. The three main components of AI-Based Load Management are (i) customer classification, (ii) dynamic pricing to match consumer energy demand, and (iii) demand response management.

2. Related Works:

Elasticity Thresholds: Static and Dynamic: Two situations include the employment of reactive elasticity: (i) when employing the conventional method with fixed thresholds [8, 11, 17, 21, 23, 25]; (ii) when employing alternative methods to modify the threshold values at runtime [6, 12, 16, 26]. There are minimum two thresholds that control either vertical or horizontal elasticity in both cases: a lower limit t_l and a higher t_u . The 195 writers agree that, even in the second case, the threshold-based technique's performance is strongly influenced by the parameters chosen. [5] Apart from performance, measurements related to energy usage and costs are crucial from the standpoints of cloud administrators and users alike. Additional issues include oscillations on VM allocations, often known as VM 200 thrashing, and responsiveness to initiate elasticity operations. In contrast to the second, which is typically handled with variables that indicate how many times a threshold must be attained to trigger a potential elastic action and a cool-down period after resource reorganisation; the first is actually dependent on t_l and t_u since a load value within each of them implies not initiating resource reorganisation.

RightScale is a management system that offers elasticity and control to many private cloud systems (CloudStack, Eucalyptus, and OpenStack) as well as various public cloud providers (Amazon, Rackspace, GoGrid, and others). The Elasticity Daemon, which powers the solution's automatic-reactive mechanisms, monitors the input queues and launches employee replicas to process jobs that are waiting in the queue. There are various scaling indicators (from hardware and applications) that may be utilized to figure out how many worker instances to start and when to start them. [6] Offering elasticity options to web-based applications that support many clouds, including Rackspace, Amazon, and Cloudstack, is the aim of the open-source Scalr project supports MySQL, PostgreSQL, Redis, MongoDB, Apache, and Nginx at the moment. Similar to RightScale, the operations turn on events based on metrics from hardware and software monitoring.

In distributed environments like clouds, appropriate resource provisioning is critical. Insufficient resource provisioning may result in resource saturation, which may raise response times or, in the case of ARs, increase the number of denied requests. Overprovisioning may result in underutilized resources, which would force the service provider to pay for the wasted processing power. The authors introduce a cost optimization method that minimizes the cost associated with resource supply over a predetermined period to handle the issue of over and under-provisioning. [7] To determine the best course of action for resource provisioning, the algorithm takes into account both the price unpredictability from the supplier of resources and the demand unpredictability from the cloud-based consumer side.

A compilation of the most recent research on autoscaling strategies specifically and handling resources on clouds in general. An auto-scaling feature is offered by VMware's Distributed Resource Scheduler (DRS) evaluating the variations in auto-scaling methods while accommodating several resource management approaches. A cloud resource management middleware called Haizea Lease Manager uses leases to plan requests and manage resources (Virtual Machines). Haizea allows users to use leases to schedule reservations in advance. Haizea launches additional instances if a lease is sought to manage the incoming workload. [8] Amazon CloudWatch is an additional resource management solution that supports scalability based on metrics. These methods, however, are not capable of handling AR scheduling. To handle ARs effectively and increase profit for the gateway cloud provider while decreasing the cost to users for processing their requests, a proactive auto-scaling technique based on future workload projections is described in this article.

The adaptability of host ERP systems is increased by a cloud-based ERP framework, which makes use of cloud infrastructure's capabilities. Service Level Agreements (SLA) guarantee availability, affordability, and scalability, which are the three key advantages of cloud computing. Cloud ERP aims to increase an organization's specified viability and efficiency in data management and storage by combining various ERP framework concepts with cloud services. Cloud computing, with its many technological advancements and massive data sets that contain sensitive and classified information, is a significant feature of the current day. [9] By offering a storage network, cloud technology is associated with dependability in managing massive volumes of data and preventing data failure.

3. Methods and Materials:

3.1 Artificial Intelligence (AI):

The field of artificial intelligence (AI) is generating a lot of interest and investigation. Researchers are working to improve AI by creating intelligent systems and developing sophisticated machine learning algorithms. The goal of AI research is to increase machines' capacity for understanding, reasoning, and decision-making. Researchers in artificial intelligence are investigating ways to improve the accuracy and efficiency of machine learning systems. They are experimenting with novel techniques and models that perform well with large datasets, enhance the adaptability of AI systems, and facilitate comprehension of these systems' inner workings. [10] Additionally, scientists are developing AI algorithms that can adapt to changing circumstances or learn from small amounts of data, enabling robots to become increasingly proficient at whatever they do over time. Artificial intelligence architecture is shown in Figure 3.1.

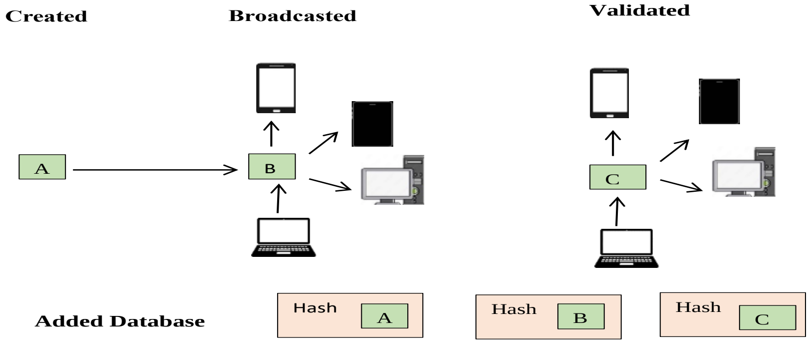

Figure 3.1: Architecture of Artificial Intelligence

3.2 Proposed model of LSTM:

An input of a_t and an output of b_t are used to illustrate a portion of the network of neurones in Figure 3.2. Data can be sent from a single network phase to the LSTM with the help of a loop. One kind of RNN that can recognise long-term dependencies is the LSTM method. A chain of neural network modules is the shape of every recurrent neural network. This recurring module in ordinary RNNs will have a straightforward structure. It is possible to characterise Figure 3.3(a) as having activation functions for the concealed, setting, results, and intake layers, respectively, as well as c_t, d_t, and e_t. [11]These are all sigmoid functions, with t being a time instance, b_t and a_t being the output and input, respectively, and bias values (O_b for output bias, I_b for input bias, and H_b for bias values) T stands for the activation function, Crosses(X) for the multiplication operation, and hidden layer bias.

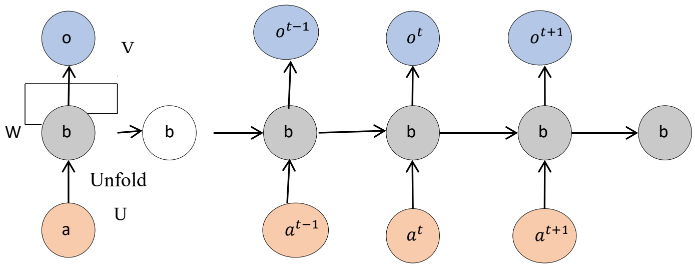

Figure 3.2: Recurrent neural network architecture (RNN)

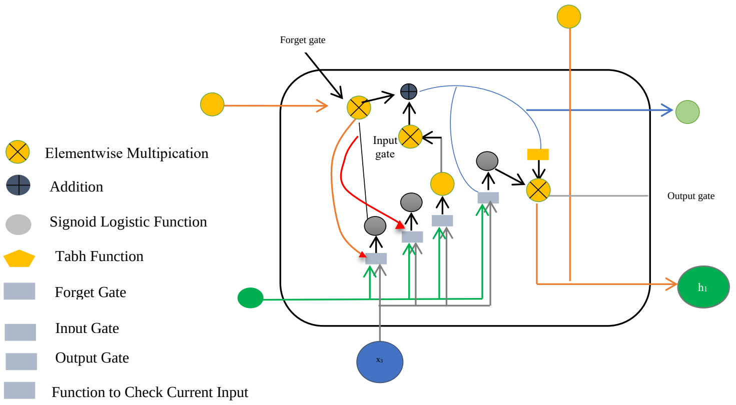

Figure 3.3 (a): LSTM Architecture

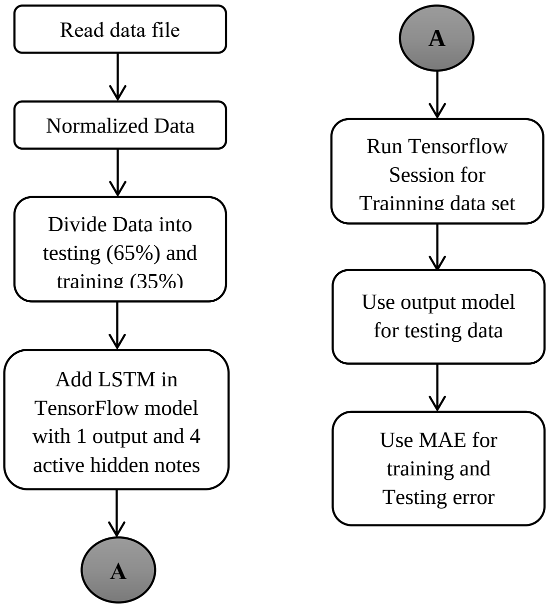
Read data file
Normalized Data
Divide Data into testing (65%) and training (35%)
Add LSTM in TensorFlow model with 1 output and 4 active hidden notes
A
A
Run Tensorflow Session for Trainning data set
Use output model for testing data
Use MAE for training and Testing error

Figure 3.3(b): LSTM algorithm flow diagram

The provided data set is forecasted using LSTM, and the error is propagated using the root mean square propagation optimiser. In this case, $\in = 1,$ e−10, p=0.0, $\beta^{-} = 0.9$ and $\alpha' = 0.1$. Step size represents the learning rate, whereas the decay represents the history/coming gradient discounting factor. Because momentum is a floating-point quantity, it helps to prevent being trapped in the minimum at the local level. A tiny integer called epsilon is used to prevent zero denominators. One layer is utilized to operate three effective context nodes and four active hidden nodes. An instrument for deep learning The LSTM model is added via Tensor Flow. Bias values and activation functions are determined by the tensor flow. Figure 3.3(b) depicts the LSTM flow.

$$MAE = \frac{1}{m}\sum_{m} \left| Y_p - Y \right| \qquad (1)$$

Using eq. (1) the mean absolute error (MAE) for the accuracy measure is calculated.

3.3 LSTM Model:

When compared to more conventional machine learning techniques, recurrent neural networks have become more prevalent and efficient. RNN handles arbitrary-length sequences in addition to stationary input and output patterns. Because RNNs have short-term memory issues, LSTM was developed as a solution to the problems of gradient explosion and vanishing gradient. This study uses long short-term memory (LSTM) experiments to predict user-requested CPU and memory utilization. [12]When contrasted with other artificial intelligence algorithms, particularly about time series data, LSTM has demonstrated superior prediction accuracy in the majority of recent publications. The RNN mechanism is the foundation for LSTM operation.

Important features can be captured by the LSTM model, which can then retain that knowledge for a considerable amount of time. The LSTM model has a unique feature called the Memory cell, which is an intermediate kind of storage, as seen in Figure 3.4. Because it decides to either ignore or preserve the memory information, the memory cell is also known as a gated cell. The sigmoid layer in gated cells produces numbers between zero and one. Information can be preserved with a value of zero or removed with a value of one. According to the weighing values provided during the conditioning phase, decisions are made. Thus, the model picks up the skill of keeping the data it needs and discarding the rest.

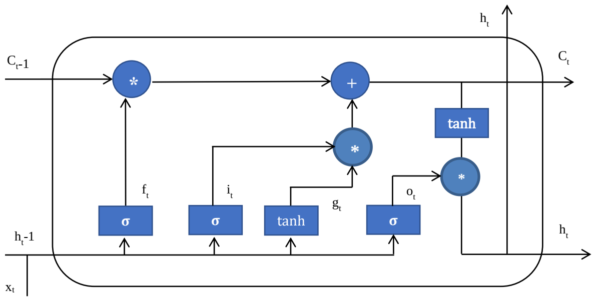

Figure 3.4: Internal LSTM block structure

3.4 Dataset:

The preceding $n{-}1$ request vector must be supplied into the network in order to train the workload prediction model. The n_{th} request scalar is the real label for the corresponding request vector sequence. The output of the workload model represents the n_{th} request, and it's the same as the framework for language creation due to the usage of the many-to-one LSTM model.

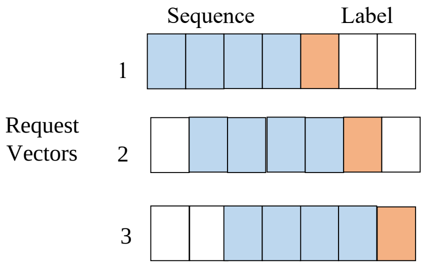

Figure 3.5 (a): Creation of the data collection for the workload prediction model

The idea remains the same even though the label that the network produces has a new meaning. This sets the first sequence to be 1 to $n{-}1$ query vector, the second sequence to be 2 to n, and so on, emulating the effect of a sliding window with a length of $n{-}1$. The method used to create the data set for this model is depicted in Figure 3.5(a).

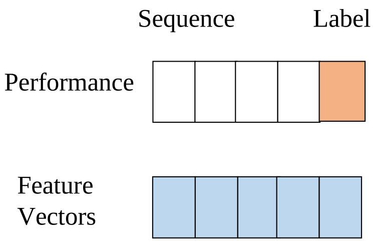

Figure 3.5(b): Creation of Data Sets for the Performance Prediction Model

Regarding the performance prediction model, input vectors with n features should be fed through the network, and the network's output must be the performance at moment tn. Figure 3.5(b) depicts the steps involved in creating the data set for this model. The process of creating the data set differs slightly from that of Figure 3.5(a) since both the old and the new feature vectors need to be added to the network. [13] Before being fed into the network, all performance results are normalized by dividing the theoretical maximum value, which can enhance both the network's performance and training effectiveness.

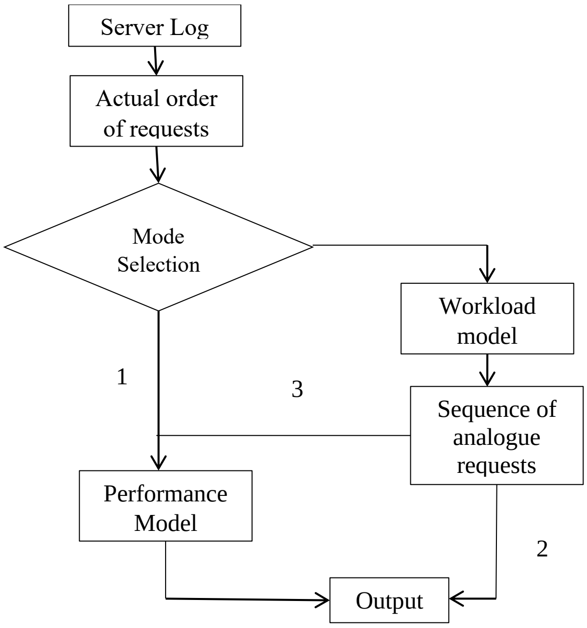

Figure 3.6: Framework of applications for two models.

Framework for Applications: Concerning the model's application, we suggest application architecture (Figure 3.6) that includes three types of model selection. By examining fresh log data, we may utilise the models to forecast the server's performance and workload. The fundamental method of using our simulations (1, 2 in Figure 3.6) is to employ the two models independently. Different seeds can be fed into the network to construct request sequences under various load situations after the model has been trained for workload prediction. This model can be trained and used in a manner that is quite similar to how RNN-LSTM is used in the field of natural language generation (NLG). The model for predicting performance has the ability to forecast the server's performance under various workload scenarios. Additionally, two models can be combined: the workload prediction model can send the analogue request sequence to the performance prediction model (3 in Figure 3.6) after first producing the sequence.

4. Results and Discussion:

The workload prediction experiment involved the generation of 42677 URL requests with identical request IDs by the model. The number of vectors of the top 5 most often requests are compared, as the amount of distinct IDs is excessive and most queries only occur periodically. Experiment results are displayed in Table 4.1 and Figure 4.1. The data indicates that less than 1% of requests are of the majority of request types, with requests with ID: 2 accounting for almost half of all requests.

Table 4.1: Actual versus predicted resource request - workload prediction

Proportion					
Requests (ID)	2	10	20	35	52
Actual Resource Requests	0.6421	0.0723	0.0542	0.0482	0.0142
Predicted Resource Requests	0.5175	0.0747	0.0527	0.0452	0.0187

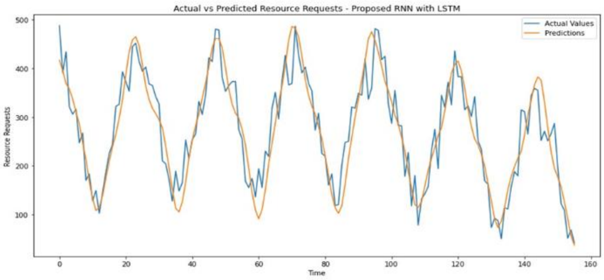

Figure 4.1: Actual versus predicted resource request - workload prediction

4.1 Performance Comparison:

We compared it to more conventional deep learning models, like feedforward neural networks (DNN) and simple RNN models, to assess how well the suggested RNN+LSTM hybrid model performed in resource provisioning. [14] Several indicators related to resource management and forecast accuracy were included in the common server request dataset used for the study.

Overprovisioning Reduction: Because the RNN+LSTM model effectively predicts and adjusts for resource needs, it considerably reduces overprovisioning, which lowers infrastructure and energy costs.

Power Consumption: Reduced power usage was a direct consequence of the hybrid models improved demand forecasting, which led to higher resource efficiency.

Load Prediction Accuracy: The RNN+LSTM model predicted server load requests more accurately. This is essential to guaranteeing effective resource allocation and minimizing service response times.

Response Time: The system more reliably satisfies quality of service (QoS) criteria and efficiently increases service response times with precise load projections.

Table 4.2: Overprovisioning Rate Comparison

Model	**Overprovisioning Rate**
RNN+LSTM	7.5
Traditional DNN	12.3

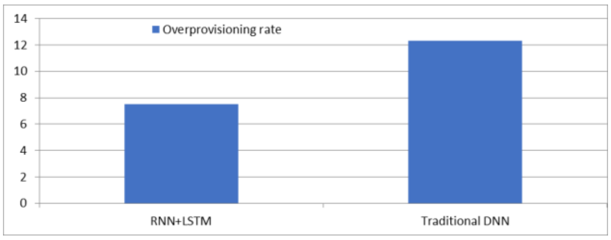

Figure 4.2: Comparing RNN+LSTM WITH Traditional DNN of Overprovisioning rate

Table 4.3: Load Prediction Accuracy

Model	Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)
RNN+LSTM	0.20	0.36
Traditional DNN	0.35	0.47

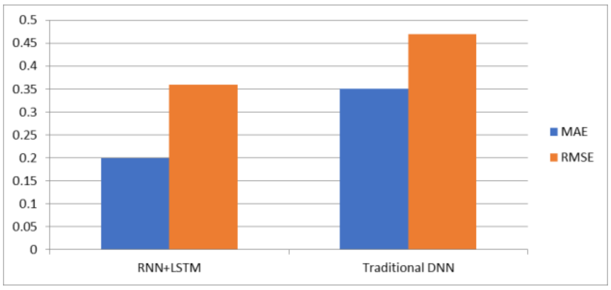

Figure 4.2: Comparing RNN+LSTM WITH Traditional DNN of MAE and RMSE

5. Conclusion:

Using a hybrid recurrent neural network (RNN) and long short-term memory (LSTM) models, the research investigates a cutting-edge AI-driven method of resource provisioning that enhances system adaptability and efficiency. In order to meet the rigorous latency and reaction time requirements of systems like high-traffic Enterprise Resource Planning (ERP) systems, this system is built to dynamically modify resources in response to changing service needs. The primary objective of this strategy is to reduce overprovisioning, which lowers energy and infrastructure expenses while preserving excellent service quality (QoS). The system makes optimal use of resource allocation by employing deep learning algorithms to forecast future load requirements with high accuracy. The AI model makes proactive provisioning decisions that improve service response times and better satisfy customer needs possible by accurately predicting server processing loads. Based on a common server request dataset, the experimental findings show that the Hybrid RNN+LSTM model performs better than conventional deep learning models in terms of prediction accuracy and resource management efficiency. By reducing overprovisioning and power consumption while improving load prediction accuracy and service response times, the RNN+LSTM hybrid model proves to be a valuable tool for dynamic resource provisioning in server environments.

6. Reference:

[1] Feng, D., Wu, Z., Zuo, D., & Zhang, Z. (2019). ERP: an elastic resource provisioning approach for cloud applications. Plos one, 14(4), e0216067.

[2] Barnawi, A., Sakr, S., Xiao, W., & Al-Barakati, A. (2020). The views, measurements and challenges of elasticity in the cloud: A review. Computer Communications, 154, 111-117.

[3] Iqbal, S., & Heng, A. (2023). AI-Driven Resource Management in Cloud Computing: Leveraging Machine Learning, IoT Devices, and Edge-to-Cloud Intelligence.

[4] Kumari, A., Gupta, R., Tanwar, S., & Kumar, N. (2020). Blockchain and AI amalgamation for energy cloud management: Challenges, solutions, and future directions. Journal of Parallel and Distributed Computing, 143, 148-166.

[5] da Rosa Righi, R., Rodrigues, V. F., Rostirolla, G., da Costa, C. A., Roloff, E., & Navaux, P. O. A. (2018). A lightweight plug-and-play elasticity service for self-organizing resource provisioning on parallel applications. Future Generation Computer Systems, 78, 176-190.

[6] Galante, G., & de Bona, L. C. E. (2012, November). A survey on cloud computing elasticity. In 2012 IEEE fifth international conference on utility and cloud computing (pp. 263-270). IEEE.

[7] Melendez, J. O., Biswas, A., Majumdar, S., Nandy, B., Zaman, M., Srivastava, P., & Goel, N. (2013, May). A framework for automatic resource provisioning for private clouds. In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (pp. 610-617). IEEE.

[8] Biswas, A., Majumdar, S., Nandy, B., & El-Haraki, A. (2014, December). Automatic resource provisioning: a machine learning based proactive approach. In 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (pp. 168-173). IEEE.

[9] Yathiraju, N. (2022). Investigating the use of an artificial intelligence model in an ERP cloud-based system. International Journal of Electrical, Electronics and Computers, 7(2), 1-26.

[10] Gutierrez-Estevez, D. M., Gramaglia, M., De Domenico, A., Dandachi, G., Khatibi, S., Tsolkas, D., ... & Wang, Y. (2019). Artificial intelligence for elastic management and orchestration of 5G networks. IEEE wireless communications, 26(5), 134-141.

[11] Muneer, A., Ali, R. F., Almaghthawi, A., Taib, S. M., Alghamdi, A., & Ghaleb, E. A. A. (2022). Short term residential load forecasting using long short-term memory recurrent neural network. International Journal of Electrical & Computer Engineering (2088-8708), 12(5).

[12] Anupama, K. C., Shivakumar, B. R., & Nagaraja, R. (2021). Resource utilization prediction in cloud computing using hybrid model. International Journal of Advanced Computer Science and Applications, 12(4).

[13] Huang, Z., Peng, J., Lian, H., Guo, J., & Qiu, W. (2017). Deep recurrent model for server load and performance prediction in data center. Complexity, 2017(1), 8584252.

[14] Xu, M., Song, C., Wu, H., Gill, S. S., Ye, K., & Xu, C. (2022). esDNN: deep neural network based multivariate workload prediction in cloud computing environments. ACM Transactions on Internet Technology (TOIT), 22(3), 1-24.