International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

AI-Based Cost Optimization Model for Cloud Resource Management

Monika KC[1], Dr. Umadevi Ramamoorthy[2]

[1]School of Science and Computer Studies, CMR University Bengaluru, Karnataka, India
Email: *monika_m[at]cmr.edu.in*

[2]Associate Professor, School of Science and Computer Studies, CMR University Bengaluru, Karnataka, India
Email: *umadevi.r[at]cmr.edu.in*

Abstract: ***Cloud computing is really important, for applications because it gives us scalability, flexibility and resources when we need them. The problem is that we often do not use these resources in the best way, which means we waste money and do not use our infrastructure fully. This project is about a way to manage cloud resources using artificial intelligence. We use machine learning and predictive analytics to make sure we use our resources in the way possible while keeping our applications running well. The system looks at how we used resources in the past and how we are using them now to figure out what we will need in the future. Then it automatically adjusts our cloud resources based on how busy our system's. It finds resources that are not being used. Suggests we make them smaller or get rid of them. It also helps us scale up or down when we need to. It tells us how to deploy our applications in a way that saves us money. We have a dashboard that shows us how our resources are being used, like how much of our CPU and memory we're using and how much money we are spending on the cloud. This helps us see what is going on and make decisions. The artificial intelligence system gets better over time because it keeps learning from how our workload changes. We tested this approach and it really works. It saves us a lot of money on cloud infrastructure while keeping our system running and using our resources efficiently. By automating things predicting what we will need and keeping an eye on everything we can stop wasting resources work efficiently and make decisions based on data. This solution helps organizations use their resources in the best way possible even when things are changing fast. Cloud computing is a part of this and our solution makes it scalable, reliable and affordable.***

Keywords: Cloud Computing, Cost Optimization, Artificial Intelligence, Machine Learning, Predictive Analytics, Resource Allocation

1. Introduction

These days a lot of companies are using cloud computing because it allows them to use tools like data storage and computing power whenever they need to. Of buying expensive equipment companies can easily grow their systems through big providers like AWS, Microsoft or Google. With services available when needed teams can adjust quickly without any delays. Paying for what they use is very practical especially for small companies that are just starting out. Over time this approach has become very common across industries. Cloud computing has a lot of advantages. Managing costs can be very difficult. When workloads change suddenly companies can experience spending. Poor setup of tools and lack of tracking can also cause problems. If there are many reserved servers it can be a waste; if there are too few it can hurt speed and break promises to clients. Running systems can also add strain and uneven work distribution across machines can make things worse. It can be hard to see how things are being used, which can lead to overspending. In todays paced world old-school tracking and fixed guidelines are not enough to manage messy cloud setups. When loads change quickly these tactics can fail. There are no real-time adjustments and no solid forecasts for what comes That is why smarter systems are being used, which can run themselves to balance usage against spending without any problems. These systems can get better over time by using reinforcement learning from outcomes. When artificial intelligence looks at usage numbers patterns start to show up. It becomes possible to forecast what resources will be needed next because of these trends. Of guessing decisions are supported by methods like clustering or regression analysis. Timeseries models also help shape choices based on how things unfolded A new approach to managing cloud systems uses intelligence to decide how power is assigned. When needs change adjustments happen in time. There is no waiting. Prediction determines when extra capacity is added or removed. Some machines may be running light while others may be straining too hard. This method can spot these quickly. Scaling happens automatically. It follows clear rules. Money choices guide every move behind the scenes and performance stays steady even as savings are made. Resources are used where they are needed not everywhere at once. The system also includes a monitoring interface that shows live updates on how resources are being used how well they are performing and spending patterns. With this view teams can make choices adjust their setups for better efficiency pick smarter pricing options and shut down unused systems when needed. This project is building a way to cut cloud costs without losing performance. Of just tracking usage it adjusts resources using artificial intelligence. Efficiency grows because the tool learns how systems behave over time. Savings happen as wasted capacity is redirected automatically and sustainability improves when energy-heavy workloads shrink across data centres. Scalability stays strong even as demands shift unexpectedly and cloud computing remains a tool for many companies, like AWS, Microsoft and Google.

2. Literature Review

Cloud computing is really important for digital infrastructure. It lets organizations use computing resources as needed. Cloud environments are getting more complicated. This makes it hard to manage resources well and keep Researchers have been looking into using Artificial Intelligence, Machine Learning, Deep Learning and Reinforcement Learning to make resource management easier. They want to automate

things make predictions about workload and cut operational expenses. Swain and other authors wrote a study on smart cloud resource management. They said that traditional ways of allocating resources often lead to wasting resources or not using them enough. This means costs for operations. Their study showed that using Artificial Intelligence to predict workload helps cloud providers use computing resources efficiently. They can still provide Quality of Service. The study found that Machine Learning really helps use resources and lowers infrastructure costs. Cloud computing and Artificial Intelligence are important, for making this work.[1].

Qu and the other people who worked with them came up with a way to manage cloud costs using math. They wanted to figure out the way to pick the resources that would cost the least amount of money. Their plan looked at all the ways that cloud resources are priced and chose the one that would save the most money. When they tried out their plan they found out that it really did save money on cloud costs. The good thing was that it did not hurt how well the applications worked. The people who wrote the paper said that using this kind of planning can make a big difference in managing cloud costs, over a period of time. They think cloud cost management can be improved a lot by using this method. Cloud cost management is very important. This method can help with cloud cost management.[2].

Sathupadi et al. introduced RAP-Optimizer, an AI-based predictive resource management model specifically designed for Artificial Intelligence as a Service (AIaaS) platforms. Their framework continuously monitored application workloads and dynamically allocated computing resources according to future demand predictions. Experimental results demonstrated significant reductions in infrastructure costs while maintaining Service Level Agreements (SLAs). The authors highlighted the importance of predictive AI models for cloud cost optimization [4].

Rusum and Anasuri proposed an AI-augmented FinOps framework that automates cloud cost management through predictive intelligence. Their research integrated machine learning with financial operations to continuously monitor cloud expenditure, identify cost anomalies, and recommend optimization strategies such as rightsizing and workload scheduling. The study showed that AI-assisted FinOps improves budgeting accuracy while reducing unnecessary cloud spending [5].

Vhatkar et al. presented an intelligent optimization algorithm for virtual machine provisioning, placement, and recycling in cloud computing environments. Their deep learning-based approach dynamically adjusted virtual machine allocation based on workload behaviour. Experimental evaluation showed improvements in execution time, resource utilization, and overall cloud cost efficiency. The proposed algorithm also enhanced energy efficiency by minimizing idle resources [6].

Zhang and the other people who worked with him did a review of how Artificial Intelligence is used in managing cloud resources. They talked about Artificial Intelligence techniques like machine learning and deep learning and reinforcement learning and predictive analytics to make the cloud work better. What they found out is that using Artificial Intelligence to manage resources works better than the way of doing things because it helps the cloud work with more people saves money and makes sure the service is working all the time. Artificial Intelligence is really good, at managing cloud resources. [7].

Kumar et al. proposed a machine learning-based resource allocation framework that utilizes historical workload data to predict future computing demands. Their model dynamically allocates cloud resources according to workload variations, reducing unnecessary virtual machine provisioning. Experimental results demonstrated improved resource utilization, faster response time, and lower cloud infrastructure costs compared to conventional scheduling algorithms [8].

Singh and his team created a way to make cloud computing work better. They made a computer program that can predict when it needs to add or remove resources. This program looks at how work the cloud is doing and makes changes as needed. The people who made this program tested it. Found out that it can save a lot of money. They also found out that it does not hurt how well the applications work or how available they are. The study by Singh and his team showed that this way of adding or removing resources is a good way to make the cloud work better. Cloud computing is what they were trying to improve. They used predictive auto-scaling to do it. Predictive auto-scaling is what made cloud computing work better, in their study. [9].

Adnan et al. introduced a machine learning-based dynamic resource provisioning model that predicts future resource requirements using historical utilization data. Their intelligent provisioning mechanism minimized resource wastage while improving overall cloud performance. Experimental evaluation confirmed that predictive resource allocation significantly reduces cloud operational expenses compared to static provisioning approaches [10].

Chen et al. proposed a Deep Reinforcement Learning (DRL) framework for cloud resource allocation and cost optimization. Their intelligent agent continuously learned optimal resource management strategies through interaction with dynamic cloud environments. The proposed model effectively balanced cloud cost and system performance while adapting to workload variations. The authors concluded that reinforcement learning offers superior adaptability compared to rule-based resource management [11].

Kumar and Mishra looked into ways to use analytics techniques for managing cloud resources more intelligently. They used intelligence and statistical methods to forecast what the workload demand would be like in the future. This helped them make decisions about scaling the cloud. The framework they proposed made it easier to predict the workload. Saved a lot of money by not wasting cloud resources. At the time Kumar and Mishras framework made sure the system performed well. The predictive analytics techniques used by Kumar and Mishra were really useful, for cloud resource management. [12].

Zhang and the people he worked with looked into using machine learning to make cloud resource scheduling and management. They wanted to see which supervised learning algorithms worked best for guessing workload and making scheduling. What they found out was that smart scheduling algorithms make use of resources and cut down on computational overhead when compared to the old way of scheduling. Zhang and the people he worked with found that machine learning optimization techniques are really good, for cloud resource scheduling and management.[13].

Wang et al. proposed an intelligent machine learning-based resource allocation framework that dynamically optimizes cloud infrastructure according to workload demand. Their predictive model efficiently balanced computational load across distributed cloud resources, resulting in improved system performance and reduced operational costs. The study emphasized the importance of adaptive AI algorithms in future cloud environments [14].

Punniyamoorthy and his team came up with a way to make the most of cloud resources using artificial intelligence. They used machine learning to figure out how to spread work across cloud clusters. When they tried it out they found that it worked well. The cloud resources were used better it cost money and it could handle more work than the old ways of doing things. Punniyamoorthy and his team found that their approach was very good at making the cloud work better. They used machine learning models to make sure the workload was spread out right across multiple cloud clusters. This made the cloud work efficiently. The results showed that this new way was better than the ways. It helped with scalability reduced the amount of money spent on the cloud and made use of the resources. This is what Punniyamoorthy and his team did with their cloud resource optimization framework, for multi-cluster cloud environments. [15].

Gaddipati and Gandikota used machine learning and cloud observability tools together to make resource management smarter. They made a system that always checks how well the cloud infrastructure is working and finds ways to make it better using analytics. The study found out that using intelligence with real-time monitoring helps optimize the cloud before problems happen which makes things run more smoothly and saves money. Gaddipati and Gandikota showed that machine learning and cloud observability tools are very useful, for managing cloud resources. [16]

3. Proposed Method

The proposed system is an Artificial Intelligence and Machine Learning-based framework. It is used for optimizing cloud resource utilization. This is done while minimizing costs. The framework monitors cloud infrastructure. It collects time and historical resource utilization data. Then it predicts workload requirements. After that it automatically allocates cloud resources based on demand. The system does this by integrating analytics and intelligent decision-making. It also uses automated cloud management. This reduces resource wastage. It improves utilization efficiency.. It maintains application performance. The proposed model supports cloud platforms. These include Amazon Web Services, Microsoft Azure and Google Cloud Platform. It provides an adaptive solution for modern cloud environments. The system integrates with cloud platforms. It does this through their APIs and monitoring services. It collects time and historical information. This is related to CPU utilization, memory usage, storage consumption, network traffic, virtual machine uptime and cloud service costs. The collected data is stored in a repository. Then it undergoes preprocessing. This removes inconsistencies handles missing values normalizes features and extracts information.

The processed data is then analysed using machine learning algorithms. These identify workload patterns, detect anomalies, forecast resource requirements and generate intelligent optimization recommendations. Based on these predictions the system automatically adjusts resources. This achieves efficiency while reducing operational expenses. The Data Collection Module is the foundation of the proposed framework. It gathers both time and historical cloud resource information. It connects directly with AWS CloudWatch, Microsoft Azure Monitor and Google Cloud Monitoring APIs. It retrieves performance metrics from environments. The collected information includes CPU utilization, memory usage, disk utilization, network bandwidth, active virtual machines workload statistics and cloud billing data. This comprehensive dataset provides the input for accurate workload prediction and intelligent resource optimization. It ensures monitoring of cloud infrastructure performance. The Data Collection Module is a part of the system. It helps the system make decisions. Before the collected data is used for machine learning it undergoes a preprocessing stage. This improves its quality and consistency. The process removes records handles missing or incomplete values filters noisy observations and normalizes numerical features to a common scale. Feature engineering techniques are applied to extract attributes. These improve prediction accuracy. Proper preprocessing ensures that the machine learning models receive structured data. This results in reliable forecasting and optimization decisions. The preprocessing stage is important. It helps the system make predictions. The Prediction Module estimates cloud resource requirements. It analyzes workload patterns and current resource utilization trends. The proposed system employs forecasting techniques. These improve prediction accuracy. Long Short-Term Memory networks capture term temporal dependencies in workload data. ARIMA models perform time-series forecasting. Random Forest Regression predicts resource utilization based on performance indicators. By combining these models the system accurately forecasts CPU usage, memory requirements, storage demand and network traffic. This enables cloud resource management. The Prediction Module is a part of the system. It helps the system prepare for demands. The Optimization Engine utilizes the predicted workload information. It determines the cost-effective resource allocation strategy. Based on the forecasted demand the engine automatically performs resource scaling, virtual machine resizing, workload migration and resource consolidation. It identifies underutilized instances that can be terminated to reduce unnecessary cloud expenses. The engine directly communicates with Amazon Web Services, Microsoft Azure and Google Cloud Platform through cloud APIs. This enables execution of optimization decisions while maintaining application availability and performance. The Optimization Engine is a part of the system. It helps the

system save money. The Decision-Making Module converts the predictions generated by the Prediction Module and recommendations from the Optimization Engine into actions. It evaluates optimization alternatives. It considers cost, system performance, resource availability and service reliability before selecting the solution.

Finally, the Feedback Loop enables improvement of the proposed Artificial Intelligence-based optimization framework. After each optimization decision is executed the system monitors its impact on resource utilization, application performance and cloud expenditure. The collected performance data is fed back into the machine learning models. This allows them to learn from decisions and adapt to changing workload patterns. This continuous learning process improves prediction accuracy enhances optimization efficiency and ensures that the system remains effective in dynamic cloud computing environments over time. The Feedback Loop is a part of the system. It helps the system get better over time. The Artificial Intelligence and Machine Learning-based framework is a tool. It helps optimize cloud resource utilization. It minimizes costs. The system is designed to support cloud platforms. It provides an adaptive solution for modern cloud environments. The proposed model is a part of the system. It helps the system make decisions.

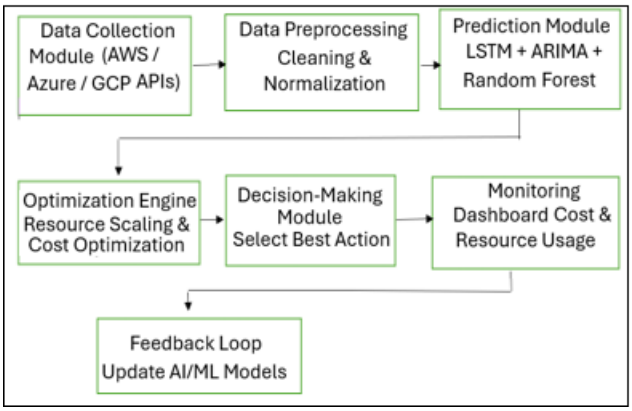

Figure 1: Block diagram of Proposed System

We use Long Short-Term Memory and other things like ARIMA and Random Forest Regression to predict how much we will use resources. This helps us to save money on operations. The Long Short-Term Memory model looks at things that happened in the past like how the CPU was used and how much memory was used. It also looks at when people asked for things to be done and when they were done. This helps it to learn when we will need resources in the future. We clean up the data. Make it nice and simple before we put it into the Long Short-Term Memory network. This network is special because it can remember things that happened a time ago and use that to make good predictions. At the time the ARIMA model tries to figure out how much we will spend on cloud resources in the future. It does this by looking at how much we spent in the past and finding patterns. We are making a system that uses different machine learning things together. This helps companies to guess how much they will spend on resources and make a good budget. We also use Random Forest Regression to predict how much we will use cloud resources and how much it will cost. It does this by looking at different things like how much CPU is used and how much memory is used. It makes different decisions and then combines them to make a good prediction. We take the predictions, from all these models. Put them together to get a good idea of what we will need in the future. This helps our system to automatically add or remove resources as needed and keep costs while still making sure everything works well.

4. Experimental Setup

A test is run on a tool that uses intelligence to reduce spending on cloud computing without slowing things down. The goal is to see how resources are used over time and find trends that will matter later. Of just guessing the tool tries to forecast what demand will look like next week or next month. The tool then adjusts resource assignments based on these forecasts to make decisions. Different math methods are used to show results. Each one is compared to the others. The test needs a computer setup that starts with an Intel i5 chip has 8 gigabytes of memory and a solid-state drive with least 256 gigabytes. This setup can handle amounts of data without slowing down which makes training models faster. The software is built on Python. Uses tools like NumPy and Pandas to clean and organize data. It also uses Scikit-learn for methods like Random Forest and basic prediction tasks. For sequence-focused nets, like LSTMs the tool uses TensorFlow or Keras. The tool uses services like AWS, Azure and Google Cloud to get live and past details on how resources used and what they cost. It uses monitoring features and API access to get details like processor load, RAM draw and expenses. When live figures are not available the tool uses made-up sets that act like cloud demands. The test usually uses information collected over three to twelve months. Each entry has details like CPU usage, memory load, disk space, internet traffic and money spent. The tool cleans the data before using it. Fills in missing numbers with average values. It then scales the numbers between zero and one using a method called Min-Max scaling. The tool checks links across variables to find drivers behind spending. The setup involves learning systems tuned for separate jobs. An LSTM handles guessing needs by studying past patterns. The tool also uses ARIMA to forecast money changes and a decision forest to weigh factors when estimating expenses. The tool pulls data from cloud platforms cleans it and picks features. It then feeds the information into training modules for forecast engines. The models output estimates about needs and these estimates are used to adjust resources. The tool tracks performance nonstop using displays and visual interfaces. After everything is done the outcomes from the intelligence method are compared to a standard version, without smart tuning tricks. The differences show up clearly in spending cuts how well gear gets used and overall smoothness of operation. The new approach slashes cloud bills quite a bit even as speed and reliability stay steady.

5. Results and Discussion

Despite older methods falling short, the new AI system cuts costs more effectively by using smarter resource handling. Not only did machine learning analyse past and live cloud activity, but it also learned how demand shifts over time. Instead of guessing needs blindly, the setup actually foresaw spending trends with solid precision. Though basic models struggled, LSTM adapted well to busy and quiet periods alike through pattern tracking. Unlike rigid formulas, ARIMA handled immediate financial movements without much drift.

Because several signals mattered - like processor load, RAM draw, and data flow - the Random Forest method weighed them all carefully. Even when conditions changed fast, those combined inputs led to sharper forecasts than old-style equations ever managed Though it stumbled with fast-changing patterns, ARIMA handled steady, straight-line trends without much trouble. What made LSTM stand out was how it tracked shifts over time, giving it an edge in guessing what demand would do next. Instead of relying on simple links, Random Forest sorted through tangled inputs and still found meaningful paths. Where others faltered under complexity, this mix of methods revealed different strengths across situations. Performance evaluation using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and prediction accuracy indicated that the LSTM model achieved the highest forecasting accuracy, while ARIMA provided reliable baseline predictions for regular workloads. Because reinforcement learning joined the optimization engine, things started working better. Learning from what happened around it, the model began choosing when to add or remove resources depending on expected needs. When demand dropped, extra capacity stayed off; during spikes, enough power got turned on. Shifting like that meant fewer wasted parts and lower running expenses. Efficiency climbed without constant oversight.

Compared to the old method of fixed resource sharing, the new AI approach cut cloud costs by a notable amount. Depending on how workloads changed and setup choices, expenses dropped from one fifth up to more than a third in various trials. Instead of running constantly, resources activated just when needed, which lifted overall usage effectiveness. What stands out is that none of these gains came at the expense of service promises, so speed and usability stayed steady throughout. From time to time, visuals built in Tableau revealed how the system behaved - its costs shifting, predictions holding up or slipping, resources ebbing and flowing. When workloads changed, the charts showed what held steady and what cracked under pressure. Optimization choices played out clearly, their effects on spending laid bare across timelines and heatmaps. Clarity came not from complexity but from seeing one piece shift after another. Yet things did not go perfectly in the test run. When data lacks enough volume or comes in messy, models tend to make weaker guesses – this hits hard because results lean needs heavy computing power a poor fit where hardware is basic or tight. On another note, older setups like ARIMA fall short when faced with wild swings in trends that twist sharply off straight lines.

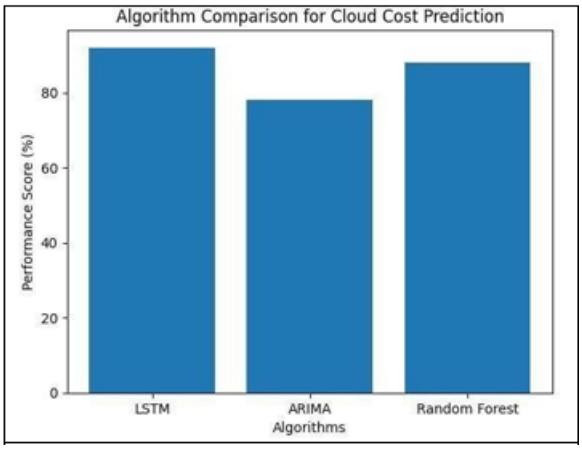

Figure 2: Algorithm comparison

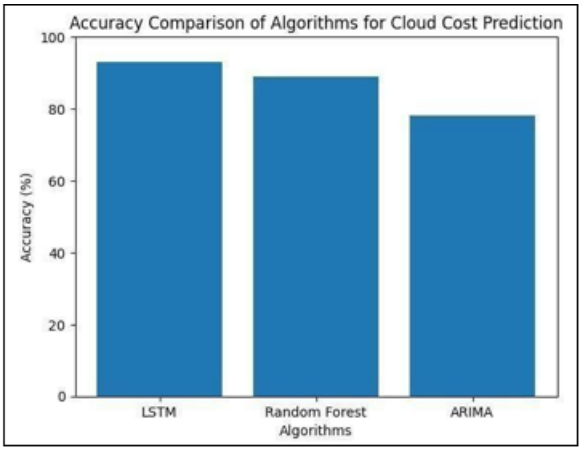

Figure 3: Accuracy comparison

Table 1: Comparison of Accuracy

Algorithm	Performance Score (%)	Accuracy (%)
LSTM	93	93
Random Forest	89	89
ARIMA	78	78

Fig 2 Based on the figure there are three methods that were used to make predictions: LSTM, ARIMA and then Random Forest. These methods were carefully looked at to see how well they could forecast what the cloud would need. The guesses about how much things would cost were also looked at closely. Each of these methods had something to offer when they were actually used. How well they worked was the thing that they were judged on. What was most important was how well each method could change when the demand for things changed. LSTM was the best because it can see patterns that happen over a time like how the demand for CPU changes over hours or days. What makes LSTM stand out is that it can remember things that happened earlier and use that to make predictions on even if a lot of time has passed. It is different from models because it can keep track of information and use it to make better predictions, which is helpful when dealing with workloads that change slowly or happen at odd times. This way of remembering things is very useful when what happened in the past can help predict what will happen in the future. Random Forest also did a good job because it can handle a lot of different kinds of information like CPU and memory and network demands. Even though it is complicated

it can handle types of data without any problems. When it came to predicting cloud demands that changed quickly or were complicated ARIMA did not do well. It was still good at predicting what would happen in the short term. Even though it had trouble with patterns that were not regular it could still make predictions if it only had to look at a small amount of data. Most of the time it could make predictions but sometimes it would make mistakes. When the demand for things changed quickly it had trouble making predictions. Lstm was the best at predicting demand and it did a better job than the other methods. Random Forest was also good at estimating costs. It was very reliable. Together these methods work well for a system that uses intelligence to help balance cloud spending. They are the choices, for this purpose.

Fig 3 Looking at how well the algorithms performed, LSTM came out on top with roughly 93% precision. Because it handles extended sequences so well, spotting trends in cloud workloads over time becomes easier. Close behind, Random Forest reached nearly 89%, managing varied inputs like CPU load, memory use, while also tracking network flow without trouble. Then there is ARIMA, landing near 78%; fine when patterns follow a straight line, yet struggles once things get complex or shift rapidly. While each method has its place, changes in behaviour across systems reveal clear differences in real-world fit. LSTM stands out with the highest prediction accuracy when tested against real data. Right after comes Random Forest, holding up well but not quite matching LSTM's precision. For tricky cloud workload behaviours, ARIMA trails behind both in performance.

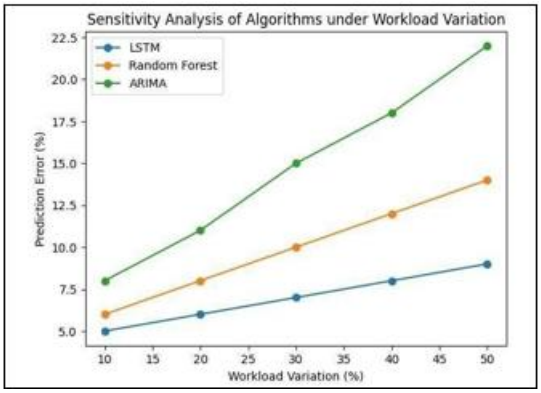

Figure 4: Sensitivity analysis

Table 2: Sensitivity Analysis of Algorithms under Workload Variation

Workload Variation (%)	LSTM Prediction Error (%)	Random Forest Prediction Error (%)	ARIMA Prediction Error (%)
10	5	6	8
20	6	8	11
30	7	10	15
40	8	12	18
50	9	14	22

Fig 4 In the figure when workloads shift more in cloud systems, one way to check prediction accuracy is through sensitivity tests. Workload swings between ten percent and fifty percent show heavier demands on resources like CPU load or memory draw, also request counts rising across time. Looking at the graph, you can see how much each method's guess changes as workloads shift. Error rates climb differently depending on the approach used. Some stay steady while others jump up fast. When demand swings grow bigger, mistakes go higher for most. A few handle changes better than the rest. Each line tracks one technique's performance under pressure. The steeper the rise, the worse it copes. Stability matters when conditions keep shifting. Not every system reacts the same way. Bigger workload jumps reveal who struggles and who holds firm.

When workloads shift more, LSTM's mistakes rise just a bit - starting at 5%, ending near 9%. Over time, its accuracy dips only slightly despite bigger changes in load When workloads shift, LSTM adjusts without losing balance. Since it picks up on extended time-based patterns, stability sticks around through change. Midway through, error climbs a bit - jumping from 6% up to nearly 14%. Though it manages many inputs without trouble, time-based trends slip past its logic. Because of that, shifts in demand affect it more than they would an LSTM, yet performance stays within reasonable limits Rising fast, ARIMA's mistake jumps from 8% up to nearly 22%. Linear trends are what this method expects in data over time. As things shift quickly, the forecast starts missing marks. One thing stands clear: the AI-driven method cuts cloud spending while sharpening how resources are used. Because it leans on forecasting tools paired with smart adjustments, shifts in demand get met smoothly. This flexibility grows naturally with usage, fitting today's shifting cloud needs without extra effort.

6. Conclusion

The AI-based cost optimization model for cloud resource management provides an efficient and intelligent approach to managing cloud infrastructure. The proposed system integrates machine learning models and optimization techniques to analyse cloud resource usage patterns, predict future workload demands, and make intelligent resource allocation decisions. By utilizing LSTM, ARIMA, and Random Forest algorithms, the system accurately forecasts both workload requirements and cloud cost trends, enabling proactive and cost-effective resource management. The integration of reinforcement learning further improves the system by enabling dynamic and automated decision-making. Unlike traditional static resource allocation methods, the proposed model continuously learns from real-time cloud data and automatically adjusts resource provisioning according to changing workloads. This helps reduce both over-provisioning and under-utilization, resulting in significant cost savings while maintaining system performance, availability, and reliability. Experimental results demonstrate that the proposed model achieves higher prediction accuracy, better resource utilization, and lower operational costs compared to conventional cloud management approaches. Furthermore, the system supports efficient cloud resource allocation while maintaining Service Level Agreement (SLA) compliance, making it suitable for real-world cloud environments. Although challenges such as data dependency and computational complexity remain, the proposed AI-based framework provides a scalable and practical solution. Overall, it effectively combines predictive analytics, intelligent automation, and optimization techniques to improve cloud resource management and minimize operational costs.

References

[1] S. R. Swain, A. K. Singh, and C. N. Lee, "Efficient Resource Management in Cloud Environment," arXiv preprint arXiv:2207.12085, 2022.
[2] Z. Qu, M. Dawande, and G. Janakiraman, "Cloud Cost Optimization: Model, Bounds, and Asymptotics," Operations Research, vol. 72, no. 1, pp. 132–150, 2024.
[3] P. Nawrocki and M. Smendowski, "Optimization of the Use of Cloud Computing Resources Using Exploratory Data Analysis and Machine Learning," Journal of Artificial Intelligence and Soft Computing Research, vol. 14, no. 4, pp. 275–289, 2024.
[4] K. Sathupadi, R. Avula, A. Velayutham, and S. Achar, "RAP-Optimizer: Resource-Aware Predictive Model for Cost Optimization of Cloud AIaaS Applications," Electronics, vol. 13, no. 22, Art. no. 4462, 2024.
[5] G. P. Rusum and S. Anasuri, "AI-Augmented Cloud Cost Optimization: Automating FinOps with Predictive Intelligence," International Journal of Artificial Intelligence, Data Science and Machine Learning, vol. 5, no. 2, pp. 82–94, 2024.
[6] K. Vhatkar, A. B. Kathole, A. P. Kshirsagar, and J. Katti, "Improved Optimization Algorithm for Resource Management in Cloud Applications with Performance Monitor of VM Provisioning, Placement and Recycling," Journal of Healthcare Engineering, 2024.
[7] Y. Zhang, X. Li, and H. Wang, "Artificial Intelligence for Cloud Resource Management and Cost Optimization: A Review," IEEE Access, vol. 12, pp. 45678–45701, 2024.
[8] Y. Zhang, B. Liu, Y. Gong, J. Huang, J. Xu, and W. Wan, "Application of Machine Learning Optimization in Cloud Computing Resource Scheduling and Management," arXiv preprint arXiv:2402.17216, 2024.
[9] A. Jayanetti, S. Halgamuge, and R. Buyya, "A Deep Reinforcement Learning Approach for Cost Optimized Workflow Scheduling in Cloud Computing Environments," arXiv preprint arXiv:2408.02926, 2024.
[10] H. Wang, S. S. Gill, et al., "Ainet0: AI Forecasting Based Carbon Neutral Cloud Resource Management for Net Zero Targets," Transactions on Emerging Telecommunications Technologies, 2025.
[11] Y. Wang and X. Yang, "Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning," arXiv preprint, 2025.
[12] V. Punniyamoorthy, A. K. Agarwal, B. Kumar, A. Mazumder, and K. Kannan, "AI-Driven Cloud Resource Optimization for Multi-Cluster Environments," arXiv preprint, 2025.
[13] S. S. Gaddipati and S. Gandikota, "Intelligent Cloud Resource Management Integrating Machine Learning with Observability Tools for Cost and Performance Optimization," International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1, pp. 469–479, 2022.
[14] S. Deochake, "Cloud Cost Optimization: A Comprehensive Review of Strategies and Case Studies," arXiv preprint arXiv:2307.12479, 2023.
[15] S. Chen, J. Lei, and K. Moinzadeh, "Cost Optimization in Cloud Computing: Capacity Reservation for Intermittent Random Demand Surges," Production and Operations Management, vol. 33, no. 6, pp. 1265–1284, 2024.
[16] H. Nerella, P. S. Puvvada, and S. Gadiparthi, "AI-Driven Cloud Optimization: A Comprehensive Literature Review," International Journal of Computer Trends and Technology, vol. 72, no. 5, pp. 177–181, 2024.