International Journal of Intelligent Data and Machine Learning (IJIDML)
eISSN: 3087-4262
Volume03 Issue10 | Page No: - 33-41

Received: 21 Sep. 2026, Revised: 30 September 2026, Accepted: 12 October 2026, Published: 26 October 2026

DOI: https://doi.org/10.55640/ijidml-v03i10-04

Deep Learning-Based Prediction of SaaS Instance Configurations for Cloud Resource Optimization

Mr. Himanshu Barhaiya
Department of Master Computer Application Lakshmi Narain College of Technology, Bhopal, India

ABSTRACT: There are numerous concerns related to the extensive use of cloud computing, including performance issues with communication- and computation-intensive services on virtual cloud resources. The present study explores the utilization of a Deep Neural Network (DNN) model to predict and optimize SaaS instance configurations on the basis of analyzing a dataset on Kaggle, which harbors Amazon AWS SaaS Sales information. After the data preparation process, such as filling in missing values, handling outliers, and normalizing features, to guarantee the validity of the results, the data is separated into training, validation, and test sets. The DNN model is then trained and tested using a number of different performance measures: accuracy, precision, recall, F-measure, and loss. The model is found to be not only excellent but also very accurate, reaching an accuracy percentage of 96.33%, and a strong balance between precision and recall is maintained. The learning curves suggest that overfitting was not a factor during the training process, which was smooth and stable, while the confusion matrix supports the model's reliability. The DNN has consistently outperformed other common methods, including KNN, LGBM, ARIMA, AdaPCA, and SV, in different datasets. To sum up, the insights obtained from this study point to the robustness and the capability of the model in processing SaaS sales data as well as in optimizing cloud resource configurations.

KEYWORDS: Cloud Computing, SaaS Optimization, Cloud Resource Management, Machine Learning, Deep Neural Network.

1. Introduction

Cloud computing is a system that lets users access computer resources (hardware and software) over the Internet as hosted services that may be flexibly scaled to meet user demands [1][2]. Cloud services are the name given to these services. In the cloud computing environment, the two primary participants are the cloud service provider and the cloud client [3]. Cloud service providers host cloud services in their data centers, which users (cloud customers) can access and utilize on a pay-per-use basis [4].

Resource optimizations are inappropriate for edge computing unless certain restrictions are included. First of all, because edge nodes are not located in the same local physical network as servers in a data center, intra-cluster latency cannot be minimal [5]. Second, some of these models do not account for networking when determining the overall energy requirement, making wireless data transmission more energy-intensive than cable. Finally, in contrast to cloud computing, the ultimate offloading node is determined by user mobility [6]. Consequently, it is necessary to modify both the optimization problem and standard cloud simulations.

These days, cloud computing is the most popular approach for supplying software and technological infrastructures. This approach gives users access to computer resources and apps without requiring them to set up or maintain hardware [7]. The three main cloud service paradigms are infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS). Software suppliers may build successful businesses with the SaaS model. Permitting the simultaneous delivery of software to several users via the cloud [8][9]. Additionally, the subscription-based business model offers chances for consistent revenue streams [10]. Therefore, maintaining a core user base is crucial to a SaaS model's commercial viability. Users utilize software solutions often and for extended periods of time when they see significant benefits from them [11]. SaaS's greatest asset is its engaged customers, who utilize it for extended periods of time with no assistance and may even recommend it to other users.

Cloud computing offers scalable infrastructure for data processing and storage [12]but edge computing

reduces latency and boosts efficiency by bringing computation closer to the devices. The integration of these technologies with artificial intelligence (AI) [13][14] opens up new avenues for developing creative solutions that can learn, adapt, and react instantly to consumer demands. Machine learning (ML) models are frequently used for real-time decision assistance in a SaaS environment. ML models in this context usually look at user behavior and engagement metrics [15]. Recent research has shown that deep learning (DL)-based prediction models, particularly those based on recurrent neural networks (RNNs), are highly capable of directly learning nonlinear temporal correlations from historical data [16]. However, many of these methods don't account for spatial connections between geographically dispersed nodes or subsystems.

A. Motivation and Contributions of the Study

Modern cloud and SaaS infrastructures encounter difficulties in their optimization processes and in predicting user behavior accurately, mainly because they don't consider the edge environments' introduction of mobility, wireless costs, and latency, which are all negative factors for the traditional models. Likewise, the current ML methods do not encompass the important spatiotemporal patterns considered, and thus the resource allocation is poor, and churn prediction is weak. Therefore, the development of next-generation cloud-edge AI frameworks that are capable of modeling real-world dynamics better, thus improving efficiency and increasing user retention in the present-day SaaS ecosystems, is a must. The essential contributions offered by the study are discussed below:

- Utilized the Amazon AWS SaaS Sales dataset sourced from Kaggle.
- Implemented a rigorous preparation procedure that included normalizing the database using the usual scaling method, eliminating outliers, and imputing missing values.
- Implemented a well-organized deep learning method, namely a Deep Neural Network, which efficiently predicts the SaaS configurations for cloud environments.
- Evaluated the model's performance using the loss function, F1-score, accuracy, precision, and recall.

B. Justification of the Study

Improving SaaS instance configurations is significant for elevating cloud performance, preventing wasteful resource usage, and supporting organizations in making wiser decisions. However, traditional ML models tend to be ineffective since they are not capable of grasping the nonlinear and multifaceted aspects and connections of SaaS sales information in their entirety. The study in this way aims at offering a more accurate and powerful predictive tool. This is the advantage of the Deep Neural Network, as it is able to delve into greater details of the data, which is typically not available in the traditional models. The paper supports its argument by referencing other widely used techniques and proving its effectiveness by the outcomes, stating that a DNN-centered approach is the most appropriate and efficient means of cloud resource control.

C. Organization of the Study

The study's structure is as follows: Section II explores the existing literature on SaaS Instance Configurations for Cloud Resource Optimization. Further, Section III outlines the methodology, including the dataset, model implementation, and performance. Next, Section IV gives the results analysis. Lastly, Section V concludes the study and gives future directions.

2. Literature Review

The primary sources that have contributed significantly to narrowing the focus and providing a rough direction of the study were a review and critical analysis of existing research on SaaS Instance Configurations for Cloud Resource Optimization.

L. Zhang and J. Bai (2026) propose a two-phase cooperative IaaS resource design technique to optimize SaaS providers' anticipated income under three pricing models. Before dynamic resource reallocation among providers with resource surpluses and deficits, each provider first examines past user data to establish optimal resource demand. Demand uncertainty is addressed, Quality of Service (QoS) breaches are decreased, and resource utilization is improved. For the fixed-price (OCCS_FI), segmented-price (OCCS_SI), and dynamic-price (OCCS_DI) models, three approaches are created. Its efficiency in preserving high resource utilization while optimizing income is demonstrated by theoretical analysis and experimental [17].

Mohapatra et al. (2025). The system adopts the Cloud-Odc framework that categorizes defects by severity, components involved, SaaS, PaaS, and IaaS layers. Experimental results on a dataset containing over 8,000 annotated bug reports show that the accuracy of detection has been 40% improved and time to resolve has also been reduced significantly. The suggested system enhances completeness of the reports, decreases the overhead of interactions between developers and users, and increases the scalability and efficiency of defect management in contemporary cloud-based processes [18].

Burle et al. (2024) compare a collection of ML models based on recall, F1 score, accuracy, and precision. Results show that the Hybrid SVM + Decision Tree model had a

higher accuracy rate of 92.71% as compared to the rest of the models and provides more accurate results of customer churn in SaaS applications. Future studies are recommended to examine more advanced feature engineering methods, take into account time dynamics to better predict changes, and involve real-time streams of data to modify models dynamically.[19].

Zhao et al. (2024) checked the effectiveness, and an approximate 24s retrieval time of 100-1000 information groups is obtained. The optimized level of information trust gives security coefficients of 98.83% and 96.86% to the human resources information to send and receive endpoints, respectively. SaaS cloud computing is also useful in improving the efficiency of the HR management system, and information trust can optimize the HR information security during transmission [20].

Alzahrani and Tang (2024) propose a new method. Specifically, the microservice-based SaaS deployment issue may be resolved with Hybrid Particle Swarm Optimization (HPSO). Throughout the optimization process, the HPSO incorporates social, cognitive, and adaptive inertia weight to balance the trade-off between exploration and exploitation. The HPSO also employs a local optimizer to improve the swarm's optimal global solution, with a focus on increasing energy efficiency. Through experimentation, they have used the HPSO and compared its performance with a GA approach. The experimental findings indicate that, in comparison to GA, the HPSO may further lower the rise in energy consumption by 3.68% [21].

Alzahrani and Tang (2023) This article proposes a novel SaaS deployment mechanism based on microservices. The data center determines where to deploy microservices under the new approach. The increasing energy consumption of the compute servers and communication network is taken into consideration by the new microservice-based SaaS deployment methodology. The deployment of microservices-based SaaS is a combinatorial optimization challenge. To address the issue, a genetic algorithm with a mending mechanism is suggested. The suggested strategy can reduce the energy usage related to microservice-based SaaS deployment by 37.55% as compared to the conventional deployment approach [22].

Sharma and Bhardwaj (2022) The purpose of this study is to use the method for the improved Emperor Penguin optimization approach to give a unique way of distributing virtual machines to fulfill the demands of individual consumers. This relevance is further demonstrated by the fact that the proposed solution relies on well-established optimization techniques for effective virtual machine allocation. The NetBeans IDE 12.4 was used to install this Java-based application [23].

Table I summarizes recent studies on SaaS Instance Configurations for Cloud Resource Optimization along with the datasets, focus, methods, findings, and limitations used in those studies.

TABLE 1. Recent Studies on Saas Instance Configurations for Cloud Resource Optimization

Author & Year	Purpose / Focus	Method / Model Used	Key Findings	Limitations / Future Work
L. Zhang and J. Bai (2026)	To optimize SaaS resource configuration, improve resource utilization, and maximize providers' expected revenue.	Two-stage collaborative IaaS resource allocation using fixed-price, segmented-price, and dynamic-price models.	Improved resource utilization and revenue while reducing Quality of Service (QoS) violations under uncertain user demand.	Deep learning-based prediction of optimal SaaS instance configurations is not explored.
Mohapatra et al., (2025)	To classify cloud-based defect reports by severity, affected component, and service layer (SaaS/PaaS/IaaS) using the Cloud-Odc framework.	Cloud-Odc classification system using >8,000 annotated bug reports.	40% improvement in defect detection accuracy; reduced resolution time; improved report completeness and scalability.	Not specified; potential for broader datasets and real-time integration.
Burle et al. (2024)	To evaluate ML models for predicting customer churn in SaaS applications.	Multiple ML models; best-performing hybrid SVM + Decision Tree.	Hybrid model achieved 92.71% accuracy, outperforming other models.	Explore advanced feature engineering, temporal dynamics, real-time data streams.
Zhao et al. (2024)	To assess SaaS-based HR information management efficiency and transmission security.	Information trust optimization in SaaS cloud environment.	Retrieval time ~24s; security coefficients 98.83% (sending) and 96.86% (receiving); better HR management efficiency.	Retrieval speed may need optimization; scalability across diverse HR systems.
Alzahrani & Tang, (2024)	To optimize microservice-based SaaS deployment with improved energy efficiency.	Hybrid Particle Swarm Optimization (HPSO) with adaptive parameters + local optimizer.	Reduced energy consumption by 3.68% compared to GA.	Further improvements needed; may need testing on larger real-world systems.
Alzahrani & Tang, (2023)	To use a combinatorial optimization technique to lower energy usage in a microservice-based SaaS setup.	A genetic algorithm (GA) that has a mechanism for repair.	Reduced energy consumption by 37.55% versus traditional deployment.	GA may still face scaling issues; potential for hybrid or more adaptive methods.
Sharma & Bhardwaj (2022)	To propose efficient virtual machine allocation for cloud customers.	Modified Emperor Penguin Optimization algorithm; implemented in Java.	Enhanced VM allocation efficiency; demonstrates usefulness of bio-inspired optimization.	Implementation limited to Java/NetBeans; may need scalability testing.

3. Research Methodology

This research is aimed at developing a deep learning architecture that has the potential to predict the trends in Amazon AWS SaaS sales. To do this, the study begins by gathering a real-world dataset from Kaggle and preparing the data by filling in missing values, removing outliers, and standardizing the dataset. A deep neural network is developed and tested using different performance indicators. Under this approach, the study would provide useful insights and accurate forecasts that can be used to make superior decisions in SaaS arrangement. The methodology flowchart is shown in Fig 1.

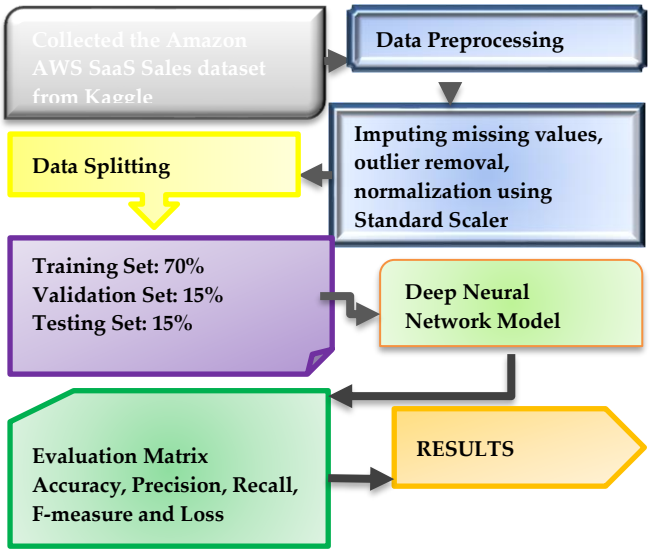


Fig. 1. Proposed Methodology Flowchart of Saas Instance Configurations for Cloud Resource Optimization

The following section highlights the steps that are illustrated in the SaaS Instance Configurations of Cloud Resource Optimization.

A. Data Collection

The research leveraged the Amazon AWS SaaS Sales dataset sourced from Kaggle. The dataset has 98 rows and 11 columns. It represents the history of a B2B SaaS company selling sales and marketing software on the AWS platform. It also suggests snapshots of 9,994 total transactions. The EDA is shown below:

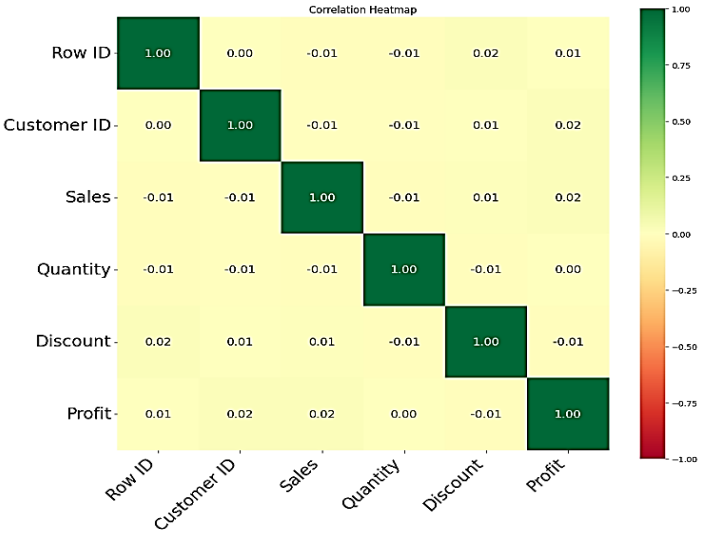


Fig. 2. Correlation Matrix

Fig 2 shows the correlation matrix, implying the pairwise correlation of features and attributes. As is clear, the features, such as Row ID, Customer ID, Quantity, Sales, Discount, and Profit, are correlated in the SaaS sample configurations.

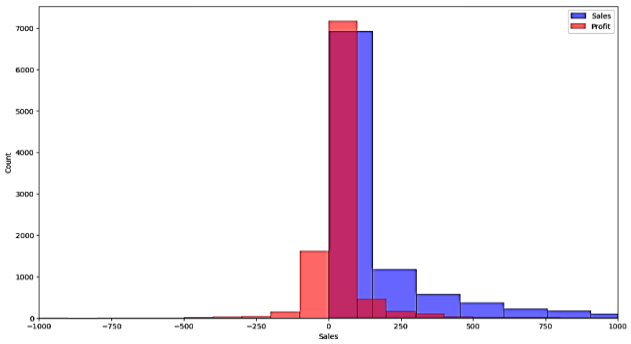


Fig. 3. Distribution of Sales and Profit

Fig 3 presents the histogram visualization of the distribution of sales and profit in the dataset. The sales data is widely spread, with values extending to the positive range, whereas the profit is clustered around 0 with some negative values.

B. Data Preparation

To ensure data consistency and quality for model testing and training, the raw dataset is rigorously preprocessed. The various preprocessing steps involved in this work include handling missing values.

- **Imputing missing values:** The process of imputing missing values entails substituting values that are deduced from the connections with other variables and the observations that include missing values.
- **Outlier removal:** Conditions were established to eliminate outliers based on real construction data using the missing value imputation procedure. For every column in the supplied dataset, outliers were eliminated separately. This procedure uses the missing value imputation approach to determine the number of outlier data points.
- **Normalization with Standard Scaler:** the numerical values were normalized to the scale of 0 and 1. Specifically, the standard method was applied to transform and normalize the features in a bounded range to ensure all the features contribute equally to the analysis. The data was scaled using the following Equation (1):

$$x_i' = \frac{x_i - \mu_i}{\sigma_i} \qquad (1)$$

$\boldsymbol{x_i'}$: normalized value for the i-th feature; $\boldsymbol{x_i}$: original value of the $\boldsymbol{i}$-th feature; $\boldsymbol{\mu_i}$: mean average of the i-th feature; $\boldsymbol{\sigma_i}$: standard deviation of the i-th feature.

C. Data Splitting

The deep learning algorithm has undergone extensive testing and training. The trials support dividing the dataset at a 70:15:15 ratio into training, validation, and testing sections.

D. Model Development

The deep neural network (DNN) model is applied in this part. The input layer of any Deep Neural Network design is crucial since its main job is to process the given data [24]. The DNN model is preferred over traditional ML techniques for a number of reasons. First, complicated natural data with strong nonlinearity cannot be handled by typical learning algorithms since they only employ one stacking layer. Second, conventional machine learning algorithms need human or technical expertise to choose the best data for accurate prediction. Technical skills include objects, whereas human skills involve people. In machine learning, a "learning algorithm" allows a software computer to simulate how a human learns to classify specific kinds of data. The hidden units in the DNN model are essential elements that actively contribute to learning. A model may become more effective during training if hidden layers are added; however, there is a cost. Processing time, model complexity, and forecast accuracy are the costs associated with this. The DNN model may be formalized as follows using Equation (2):

$$q_r = f(G_r + \sum_{w=1}^{a} L_w \psi_w^r) \qquad (2)$$

where q_r denotes the output at layer r, the bias value, and the layer r weight via neuron w.

E. Evaluation Matrix

The confusion matrix is a helpful tool that can be used to evaluate how well DL and ML models categorize data in comparison to the dataset's real labels. This matrix has four essential components:

The number of instances that were accurately classified as positive is known as True Positive (TP). False Positive (FP): The number of instances in which the model misclassifies cases that are actually negative as positive. The quantity of correctly categorized negatives is known as True Negative (TN). False Negative (FN): The number of truly positive cases that the model has classified as negative.

These factors form the foundation for evaluating the suggested model's categorization performance using the following metrics. The number of accurately identified events (both positive and negative) divided by the total is

the accuracy. Recall: It assesses the model's capacity to detect positive cases and is also known as the true positive rate or sensitivity. Precision: It assesses how well the model makes wise suggestions and lowers false positives. F-Measure: The F-Measure provides a balance between the two classification performance characteristics by combining recall and accuracy. Loss: Loss is used to quantify the model forecasts' actual values. As shown by Equations (3) to (6), the mathematical formulae used to calculate these parameters are often provided.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (5)$$

$$\text{F} - \text{measure} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}\right) \quad (6)$$

4. Experimental Results and Evaluation

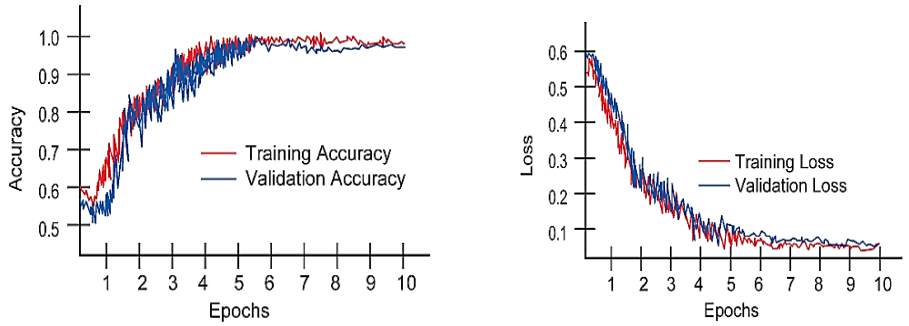
The experimental system consisted of a high-performance computing environment, which included five NVIDIA GeForce GTX 1080 GPUs; each of them had 8 GB of dedicated memory. The system was equipped with 8 GB of DDR4 RAM and powered by an Intel® Core™ i7-8700B processor with 12 MB cache and a maximum clock speed of 4.60 GHz.

A. Experimental Findings

Table II illustrates the performance of the DNN model applied to the optimization of SaaS instance configurations in a cloud environment. The performance results reflect an overall strong effectiveness with an accuracy of 96.33%, which means the predictions are very reliable. The model also preserves balanced classification performance by achieving 90.83% accuracy, 90.96% recall, and 90.86% F1-score, thus proving its ability to consistently identify the right resource configurations and at the same time keeping false positives and negatives very low.

TABLE II. Model Performance for Saas Instance Configurations for Cloud Resource Optimization

Metrics	Deep Neural Network
Accuracy	96.33
Precision	90.83
Recall	90.96
F1-Score	90.86



(a)Accuracy (b)Loss

Fig. 4. The Learning Progress in Terms of Training Accuracy, Validation Accuracy, Training Loss, and Validation Loss

Fig 4 shows model learning progress mapped out to an interval of 10 epochs. The training and validation accuracy curves in Fig 4(a) are both rising steadily, and then the plateau is probably the same for both, which points to the learning being effective and the model being able to generalize well. The loss of both training and validation is constantly going down and then is at low and stable levels throughout the period depicted in Fig 4(b), indicating smooth convergence and thus no major overfitting happening.

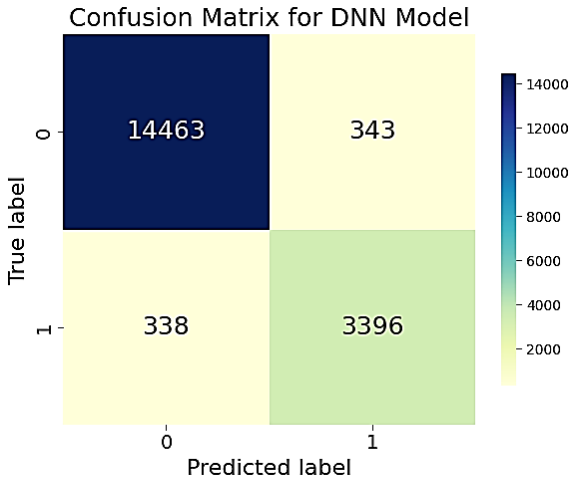


Fig. 5. Confusion Matrix of the Deep Neural Network

The Deep Neural Network's (DNN) confusion matrix is shown in Fig 5. The matrix clearly displays 14463 samples of class 0 and 3396 samples of class 1, showing the true negatives and true positives, respectively, whereas 343 samples of class 0 and 338 samples of class 1 represent the false positives and false negatives, respectively. Collectively, the proposed architecture demonstrates the superior performance.

B. Comparative Evaluation

Table III clearly demonstrates that the neural network performs efficiently in optimizing the cloud resources using SaaS configurations. The proposed architecture —a Deep Neural Network —performed well on the Amazon AWS SaaS sales database, achieving an accuracy of 96.33%. The other benchmarking models, likely KNN, LGBM, ARIMA, AdaPCA, and SV, show an accuracy of 79.72, 72.0, 89.3, 93.2, 89.7, respectively, across the multiple datasets, which is lower than the proposed model. Therefore, it is clear that the applied dataset and implemented model work efficiently.

TABLE III. Performance Comparison for Saas Instance Configurations for Cloud Resource Optimization

Ref.	Dataset	Model	Accuracy
[25]	WOA reduced, full variable & chi-squared derived datasets	KNN	79.72
[26]	KKBOX and Telco	LGBM	72.0
[27]	Publicly available Cloud data	ARIMA	89.3
[28]	Google Cluster Usage Traces Dataset	AdaPCA	93.2
[29]	Synthetic Big Data	SV	89.7
Prop.	Amazon AWS Saas Sales Dataset	DNN	96.33

The suggested Deep Neural Network approach distinctly beats the benchmark methods over various datasets, showcasing its ability to easily handle SaaS configuration data and make fewer wrong optimization choices. The model's remarkable generalization, consistent learning behavior, and even classification performance are indicators of its capability to cope with the difficulties of cloud resource optimization in terms of both traditional and modern baseline models combined.

5. Conclusion and Future Scope

Software as a service (SaaS) is a popular cloud-based software delivery strategy. SaaS systems provide their customers scalable, adaptable, and affordable services that can be accessed anywhere, at any time, and on demand. The findings of this research indicate that a Deep Neural Network is capable of analyzing SaaS sales data with high efficiency and precision in predicting the best configurations of cloud resources. The meticulous data preparation and thorough assessment resulted in exceptional model performance, which allowed for the attainment of not only high accuracy but also balanced classification metrics. The confusion matrix and learning curves provide more proof that the model was correctly trained and not overfitted. The DNN was contrasted with other conventional ML techniques, and it always came out on top, thus reinforcing its capability to detect intricate and nonlinear patterns in the dataset. In a nutshell, the results point out that DNN methodology is not only a good but also a reliable choice for enhancing cloud resource optimization in SaaS setups.

Future work may take advantage of better and more diverse SaaS datasets in order to develop the

generalization of the model. Moreover, for predictive accuracy, the use of advanced deep-learning architectures such as LSTMs or transformers can be a contributing factor. Apart from that, real-time deployment, automated hyperparameter tuning, and integrating cost-optimization metrics can do a lot to cloud resource optimization when it comes to practical environments.

References

1. S. Garg, "Predictive Analytics and Auto Remediation using Artificial Inteligence and Machine learning in Cloud Computing Operations," SSRN Electron. J., vol. 7, no. 2, pp. 1–5, 2025, doi: 10.2139/ssrn.5267117.
2. Sarraf and V. Pal, "Adaptive Deep Learning for Identification of Real-Time Anomaly in Zero-Trust Cloud Networks," ESP J. Eng. Technol. Adv., vol. 4, no. 3, pp. 209–218, September, 2024, doi: 10.56472/25832646/JETA-V4I3P122.
3. M. Konda, "Performance Evaluation of Machine Learning Inference Workloads in Containerized Cloud Computing Environments," Eastasouth J. Inf. Syst. Comput. Sci., vol. 3, no. 02, pp. 292–311, Dec. 2025, doi: 10.58812/esiscs.v3i02.949.
4. S. Kumar Chintagunta and S. Amrale, "Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation," Tech. Int. J. Eng. Res., vol. 9, no. 10, pp. 49–55, 2022, doi: 10.56975/tijer.v9i10.159996.
5. M. R. R. Deva and N. Jain, "Utilizing Azure Automated Machine Learning and XGBoost for Predicting Cloud Resource Utilization in Enterprise Environments," in 2025 International Conference on Networks and Cryptology (NETCRYPT), IEEE, May 2025, pp. 535–540. doi: 10.1109/NETCRYPT65877.2025.11102235.
6. Maddali, "An Efficient Bio-Inspired Optimization Framework for Scalable Task Scheduling in Cloud Computing Environments," Int. J. Curr. Eng. Technol., vol. 15, no. 03, pp. 229–238, May 2025, doi: 10.14741/ijcet/v.15.3.4.
7. V. Prajapati, "Role of Identity and Access Management in Zero Trust Architecture for Cloud Security: Challenges and Solutions," Int. J. Adv. Res. Sci. Commun. Technol., pp. 6–18, Mar. 2025, doi: 10.48175/IJARSCT-23902.
8. S. Garg, "Ai/Ml Driven Proactive Performance Monitoring, Resource Allocation And Effective Cost Management In Saas Operations," Int. J. Core Eng. Manag., vol. 6, no. 6, pp. 32–45, 2019, doi: 10.2139/ssrn.5267257.
9. K. K. Mohammed, "The Future is Cloud: Modernizing Big Data for the Cloud Era," Int. J. Sci. Res. Eng. Trends, vol. 11, no. 5, pp. 1–5, 2025.
10. R. Aleem, S. Malik, A. Kashif, and U. B. Tayab, "Cloud computing resources utilization forecasting using machine learning," Eng. Res. Express, vol. 8, no. 7, p. 075217, Apr. 2026, doi: 10.1088/2631-8695/ae556f.
11. Patel, "The Role of Amazon Web Services in Modern Cloud Architecture: Key Strategies for Scalable Deployment and Integration," Asian J. Comput. Sci. Eng., vol. 9, no. 4, pp. 1–9, 2024, [Online]. Available: www.ajcse.info
12. R. Patel, "Advancements in Renewable Energy Utilization for Sustainable Cloud Data Centers: A Survey of Emerging Approaches," Int. J. Curr. Eng. Technol., vol. 13, no. 05, pp. 447–454, Oct. 2023, doi: 10.14741/ijcet/v.13.5.7.
13. Y. Macha and S. K. Pulichikkunnu, "An Explainable AI System for Fraud Identification in Insurance Claims via Machine-Learning Methods," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 1391–1400, Jul. 2023, doi: 10.48175/IJARSCT-11978X.
14. S. Amrale, "A Novel Generative AI-Based Approach for Robust Anomaly Identification in High-Dimensional Dataset," Int. J. Adv. Res. Sci. Commun. Technol., vol. 4, no. 2, 2024, doi: 10.48175/IJARSCT-19900D.
15. N. Malali and S. R. Praveen Madugula, "Robustness and Adversarial Resilience of Actuarial AI/ML Models in the Face of Evolving Threats," Int. J. Innov. Sci. Res. Technol., vol. 10, no. 3, pp. 910–916, 2025, doi: 10.38124/ijisrt/25mar1287.
16. R. P. Mahajan and N. Jain, "Optimizing CT Image Quality through AI-based Reconstruction and Deep Learning Models for Enhanced Diagnostic Accuracy," in 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), IEEE, Apr. 2025, pp. 1–7. doi: 10.1109/ICDCECE65353.2025.11035138.
17. L. Zhang and J. Bai, "Optimal Collaborative Configuration Strategy of IaaS Resources Under Multiple Pricing Models for Maximizing SaaS Providers' Expected Revenue," Electronics, vol. 15, no. 3, pp. 1–29, Jan. 2026, doi: 10.3390/electronics15030568.
18. S. S. Mohapatra, R. R. Kumar, D. Muduli, and S. K. Sharma, "Deep Learning Driven Bug Detection and Severity Classification in Multi-Tenant Cloud Systems," in 2025 International Conference on Innovations in Intelligent Systems: Advancements

in Computing, Communication, and Cybersecurity (ISAC3), IEEE, Jul. 2025, pp. 1–6. doi: 10.1109/ISAC364032.2025.11156655.

19. R. Burle, S. Gaurkhede, P. Gourshettiwar, S. Izankar, S. Gundewar, and U. Pacharaney, "Hybrid Machine Learning Based Predictive Model To Improve Cloud Application Performance In Cloud Saas," in 2024 2nd International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS), IEEE, Nov. 2024, pp. 167–172. doi: 10.1109/ICETEMS64039.2024.10965099.

20. R. Zhao, Z. Lei, Z. Zhao, and D. Zhang, "Research on Cloud Computing Architecture and Optimization of Human Resource Management System," in 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India: IEEE, Feb. 2024, pp. 1–5. doi: 10.1109/ICICACS60521.2024.10498361.

21. Alzahrani and M. Tang, "Energy-Aware Microservice-Based SaaS Deployment in a Cloud Data Center Using Hybrid Particle Swarm Optimization," in IEEE Access, IEEE, 2024, pp. 140884–140899. doi: 10.1109/ACCESS.2024.3462894.

22. Alzahrani and M. Tang, "A Microservice-based SaaS Deployment in a Data Center Considering Computational Server and Network Energy Consumption," in 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA: IEEE, Jul. 2023, pp. 505–515. doi: 10.1109/CLOUD60044.2023.00067.

23. V. Sharma and S. Bhardwaj, "Enhanced Resource Control in Cloud Environment using Optimized VM Allocation in Data Centers," in 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), IEEE, Dec. 2022, pp. 1–5. doi: 10.1109/ICATIECE56365.2022.10046918.

24. V. Pal and S. K. Chintagunta, "Transformer-Based Graph Neural Networks for RealTime Fraud Detection in Blockchain Networks," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 1401–1411, Jul. 2023, doi: 10.48175/IJARSCT-11978Y.

25. M. Kotan, Ö. Faruk Seymen, L. Çallı, S. Kasım, B. Çarklı Yavuz, and T. Över Özçelik, "A novel methodological approach to SaaS churn prediction using whale optimization algorithm," PLoS One, vol. 20, no. 5, pp. 1–19, May 2025, doi: 10.1371/journal.pone.0319998.

26. Mavridou, E. Vrochidou, M. Selvesakis, and G. A. Papakostas, "Beyond Accuracy: Benchmarking Machine Learning Models for Efficient and Sustainable SaaS Decision Support," Futur. Internet, vol. 17, no. 10, p. 467, Oct. 2025, doi: 10.3390/fi17100467.

27. Gandhi, A. Jain, and H. Ave, "Cloud Cost Optimization Strategies Using Machine Learning Algorithms," Int. J. Innov. Sci. Res. Technol., vol. 9, no. 11, pp. 3573–3593, November, 2025, doi: 10.5281/zenodo.14831440.

28. S. C, B. ST, and S. S, "Achieving cloud resource optimization with trust-based access control: A novel ML strategy for enhanced performance," MethodsX, vol. 15, no. 103461, pp. 1–18, June, Dec. 2025, doi: 10.1016/j.mex.2025.103461.

29. N. K and G. A. Kumar, "Recent Improvements in Cloud Resource Optimization with Dynamic Workloads using Machine Learning," 2024, SSRN. doi: 10.2139/ssrn.4803863.