[Gopinath, 5(12): December 2018]] ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

RESOURCE MANAGEMENT AND ALLOCATION IN IOT NETWORK USING DEEP LEARNING

Dr. Madhu Gopinath
Associate professor, Border Security Force institute of technology
madhugopinath.bsfit@gmail.com

ABSTRACT

Resource management and allocation in IoT networks play a crucial role in ensuring efficient data transmission, computational load balancing, and energy optimization. This study explores deep learning-based methodologies for enhancing resource allocation in dynamic IoT environments. A hybrid approach integrating Principal Component Analysis (PCA) for feature extraction and K-Means Clustering for resource categorization is proposed to optimize resource distribution. Additionally, Deep Reinforcement Learning (DRL) is employed to develop an adaptive model capable of dynamically allocating resources based on network demands. The Round Robin scheduling algorithm is incorporated to ensure fair resource allocation, reducing latency and enhancing system throughput. Performance evaluations demonstrate that the DRL-based model achieves 98.7% accuracy, with precision and recall exceeding 98%, significantly improving decision-making capabilities. Furthermore, K-Means Clustering effectively partitions IoT devices, enabling efficient resource distribution. Comparative analysis with traditional methods reveals a 29.2% reduction in response time, a 39% increase in throughput, and a 16.2% improvement in resource utilization. These findings underscore the effectiveness of deep learning-driven resource management strategies, making them suitable for real-time IoT applications requiring adaptive and intelligent allocation mechanisms. The proposed approach enhances IoT network efficiency by optimizing computational resources, reducing energy consumption, and improving overall performance, ensuring seamless operation in large-scale deployments.

Keywords: *Resource management and allocation, Internet of Things, Deep Learning, Principal Component Analysis (PCA), Deep Reinforcement Learning (DRL), K-Means Clustering.*

I. INTRODUCTION

The Internet of Things (IoT) possesses resulted in a novel era of intelligent applications, including smart cities, smart industries, and smart agriculture, by utilizing numerous IoT devices such as detectors, controls, and intermediaries to gather and analyze substantial volumes of data produced by IoT networks [1]. In recent times, the advancement of IoT software, the evaluation of efficient wireless services for IoT devices, and the challenge of "Resource Management (RM)" have become crucial. The comprehensive achievement of resource management entails the successful and flexible utilization of resources, including time, connectivity, and periodicity [2]. Consequently, enhanced throughput, elevated data rates, reduced interference, and improved coverage are crucial factors for RM in IoT-based wireless networks. Recent advancements in hardware and software have facilitated the proliferation of IoT networks comprising numerous linked devices, hence escalating the demands for handling, storing, and communication. In smart agricultural, numerous sensors are utilized to assess environmental conditions and welfare of animals, including temperature, humidity, light intensity, noise levels, and gas concentrations. The data generated by sensors is typically gathered by the access point and subsequently transmitted to the cloud server for additional analysis [4].

Resource allocation and management have been a major concern with the advent of Deep Learning (DL) modeling as the preferred tool that facilitates intelligent decision-making and adaptive optimization [5]. IoT environments are typically complex, scalable, and dynamic that conventional resource management practices find it difficult to handle. "Convolutional Neural Networks (CNNs)" and "Deep Reinforcement Learning (DRL)" are DL procedures that can process myriad real-time data, forecast network conditions, and diplomatically manage resource allocation to build the network as effective as possible [6]. The employment of DL will make IoT networks perform enhanced bandwidth allocation, energy efficiency, and load balancing, due to which there will be decreased latency, enhanced network performance, and increased system reliability [7].

ISSN 2348 – 8034
Impact Factor- 5.070

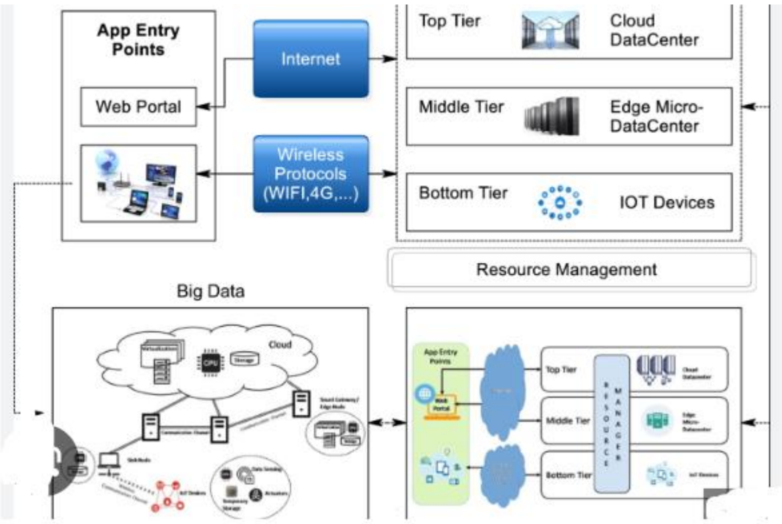


Figure 1: Resource management in IoT

This research is cantered on effective resource allocation and management in IoT networks because it seeks to enhance the highest functionality, minimize the lateness, and optimize the utilization of energy. The overall objective is to come up with a DRL-based framework that allocates resources constantly and adaptively while adapting models simultaneously. Apart from this, this paper develops an original DRL model that offers innovative resource decision-making abilities to the intelligent IoT resource management, yet they are neither utilized to ensure the optimal expansion of the system, nor do they consume more energy and also they are dynamic in the uncertain environment in the network.

II. REVIEW OF LITERATURE

The different types of resource management and allocation strategies in IoT networks that use deep learning and optimization techniques were studied by several researchers. The *Jayaprakash et al. (2021) [8]* developed an intelligent resource allocation scheme for the uplink non-orthogonal multiple access (NOMA)-IoT communications by integrating DRL and SARSA-learning. The authors show a reduced complexity and an improved throughput of the system. *Zhang (2021) [9]* introduced a novel architecture that allowed the deep neural network (DNN) models to be placed in the right positions to improve the allocation of resources, resulting in an increment of the inference accuracy by 31.4%. *Shuaib et al. (2023) [10]* proposed the Dynamic Energy-Efficient Load Balancing (DEELB) method, which led to the bandwidth consumption being reduced and therefore it was more efficient in IoT environments.

Based on previous studies, the deep learning-based techniques of dynamic resource optimization are shown to increase the levels of effectiveness, thereby offering a serious application potential in IoT resource management schemes. While Existing methods lack comprehensive integration of reinforcement learning with clustering algorithms for dynamic resource optimization. it is nonetheless that the issue of all real-time fault tolerance and scalability issues still has no solution in large-scale IoT deployments, i.e., deep learning approaches with these strategies are not yet fully practical for application.

Several studies have explored resource management and allocation strategies in IoT networks using deep learning and optimization techniques. Jayaprakash et al. (2021) [8] built an uplink NOMA-IoT connectivity resource allocation strategy that integrates DRL and SARSA-learning, demonstrating lower complexity and improved system throughput. Zhang (2021) [9] introduced an end-edge-cloud orchestration architecture that optimally places deep neural network (DNN) models to enhance resource allocation and improve inference accuracy by 31.4%. Shuaib et al. (2023) [10] proposed the “Dynamic Energy-Efficient Load Balancing (DEELB)” method, which significantly reduced bandwidth consumption, showcasing superior effectiveness in IoT environments.

Other studies include Ahmed et al. (2022) [11], who applied a heuristic-based multi-objective firefly algorithm for the maximization of data transmission of 96% throughput, 95% energy efficiency, and 85% spectrum efficiency. Nilima et al. (2024) [12] studied the requirements in terms of resources for the deployment of network management protocol in IoT applications with limited resources, and Costa et al. (2024) [13] proposed a hybrid topology following mesh and star mesh wireless sensor networks to increase efficiency in resource allocation. Periasamy et al. (2024) [14] proposed the ERAM-EE strategy with improved energy efficiency up to 18 Mbit/J for different IoT applications. Liu et al. (2025) [15] proposed an online delay-aware computation offloading strategy with up to 62.9% decrease in task execution delay when compared to conventional strategies.

While these studies present significant contributions to IoT resource management, research gaps remain in achieving real-time adaptive allocation with minimal computational overhead. Existing methods lack comprehensive integration of reinforcement learning with clustering techniques for dynamic resource optimization. Additionally, further exploration is required to enhance fault tolerance and scalability in large-scale IoT deployments, making deep learning-driven strategies more practical for real-world applications.

III. RESEARCH METHODOLOGY

Dataset Collection

For this study a "Resource Management for Cost Optimization in IoT" data set from Kaggle [16] to develop an efficient deep learning-based resource allocation system. The data contains complete information on the usage of IoT resources, in terms of cost optimization, energy efficiency, and network quality. Based on the given set of data, the deep learning models can learn to accelerate the decision-making process towards optimal bandwidth sharing, and minimum latency, and system scalability. The study is to give more emphasis to the primary goals of the IoT networks, which are the application of smart resource management methods over dynamic resource management that will significantly improve the overall system efficiency.

Data Preprocessing

Data Cleaning

The median and mean approximation method is employed to rectify data inadequacies by substituting data that is absent using the mean and median values, accordingly. The following provides full accounts of every strategy along with the corresponding equations:

- ***Mean imputation:*** The method of missing data imputation entails replacing absent data with the arithmetic means of additional numbers across a space or specified rows or situations. This approach is easy to understand starting with the assumption that missing quantities could show up at random.

$$Mean = \frac{\sum_{i=i}^{m} y_i}{m} \quad (1)$$

The total count of each non-absent integer in the row is represented by the parameter y_i, and the aggregate of these values is measured by m.

- ***Median Imputation:*** Statistical highlighting is an approach used to supplement data that is lacking by using the average of the remaining numbers in every row of an array of data or variables that are dependent. Contrasting with standard attribution, it is less susceptible to the influence of outliers and may be more suitable in some circumstances.

$$Median = Median\ (\{\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \dots \dots, \boldsymbol{y_n}\}) \quad (2)$$

The set $\{y_1, y_2, y_{3,}\ldots\ldots, y_n\}$ includes all characteristics for each row, excluding any omitted attributes.

Substitution of Missing Values

An object containing any number of resources deficiencies incapable of fully satisfying this approach input conditions. The missing values must be filled. This research substitutes the missing data with the mean of the relevant statistic. A method is shown whereby the mean of the pertinent measure is used to replace the value that is lacking. Assume a metric mv and its observations $\{mv_1, mv_2, mv_3, \ldots, mv_{100}\}$, with mv_{99} and mv_{100} being absent. The two missing data are supplemented employment:

$$mv_{99} = mv_{100} = \frac{1}{98}\sum_{i=1}^{98} mv_i \quad (3)$$

Normalization

This stage is crucial in data preprocessing that scales the extracted features in order to enhance the model training for SDP. This process also reduces such problems concerning different units and scales on features affecting the convergence of DL algorithms. Research standardizes data from experiments to facilitate testing and functioning, since most software measurements vary significantly in magnitude. The normalizing approach is used for optimal accuracy and rapid learning. This study uses the prevalent minimal-maximal normalization method to standardize the data. By using this stage, the suggested method may successfully discover patterns using the data. The lowest and maximum values of an indicator y are denoted by min(y) and max(y), accordingly.

$$Y' = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \tag{4}$$

Feature Selection

In this study, "**Principal Component Analysis (PCA)"** is employed for feature extraction to optimize resource allocation in IoT networks. PCA is utilized in this study for feature extraction to optimize the utilization of resources in IoT networks. PCA compresses big IoT data into lower dimensions without compromising the most significant features. PCA eliminates redundancy and computational complexity by transforming correlated resource parameters like network traffic patterns, device energy, and communication latency into uncorrelated principal components. PCA simplifies the complexity of the deep learning model by eliminating irrelevant or noisy features, thereby improving prediction accuracy and decision-making in IoT dynamic environments. The integration of principal component analysis with deep learning comprises three steps: standardization, covariance matrix computation, and eigenvalue decomposition.

➢ ***Standardization***

Standardization makes the mean of each feature equal to 0 and the standard deviation equal to 1, thus meaning that all of the features make equal contribution to the analysis regardless of the range that they came from. This step is important because, as we have already mentioned, PCA is sensitive to the scale of the features; the features that are measured on a large scale could dominate the results. The formula for standardization is:

$$U_{standardized} = \frac{U - \sigma}{\mu} \tag{5}$$

➢ ***Covariance matrix (CM)***

CM of the features is calculated to analyze the correlations of the features as well as the variance for each one. The covariance matrix gives the extent of association between the two variables, which is very essential for PCA. The formula for the covariance between two features U_i and Uj is:

$$Cov(U_i, U_j) = \frac{1}{n-1} \sum_{k=1}^{n} (U_{ik} - \bar{U}_i)(U_{jk} - \bar{U}_j) \tag{6}$$

Here n be a set of findings, and $\bar{U}_i$ and $\bar{U}_j$ denote the means of U_i and U_j correspondingly.

➢ ***Eigenvalue Decomposition***

It is performed on CM to extract the principle component. This entails identification of the eigenvalues as well as the respective eigenvectors. Eigenvalues are measures of the degree of spread or variance in each principal component while eigenvectors are a description of the orientation of each of these components [17]. The decomposition is given by:

$$Cov(U) = V \Lambda V^T \tag{7}$$

Denote V as the matrix representing the eigenvector, and Λ is the diagonal matrix.

Allocating resources using K-Means Clustering

In the proposed research, the user's preference for offloading computations for IoT user u_i is defined as x_i, where $x_i = \{d_i, Pr_i\}$ is the feature vector that includes the user's distance from the gateway (di) and the probability of offloading calculations (Pr_i). In order to group Internet of Things (IoT) users into H clusters according to their priorities, we propose a priority-driven user clustering method that is based on the classic K-means algorithm and minimizes the "sum of squared errors (SSE)".

$$\min_{C_h, c_h} \sum_{h=1}^{H} \sum_{x_i \in C_h} ||x_i - c_h||_2^2 \tag{8}$$

Let H represent the initial quantity of clusters, C_h denote cluster h, and x_i represent the user interface's preference feature. We find the cluster's centroid, c_h, via as:

$$c_h = \frac{1}{|C_h|} \sum_{x_i \in C_h} x_i \tag{9}$$

where |·| is the cardinality.

If the chosen cluster number H' is less than the actual cluster frequency H, the SSEs significantly decrease when the cluster number is increased by one. Conversely, the SSE exhibits no significant alterations. The optimal number of clusters is identified at the inflection point, referred to as the elbow. The efficiency index of the elbow technique, namely the "Sum of Squared Errors (SSE)", is defined as:

$$SSE = \sum_{h=1}^{H'} \sum (x_i, c_h)^2. \tag{10}$$

Deep learning model selection

The gateway employs centralized user clustering to categorize IoT users into various groups according to their unique traits and priorities. It is assumed that all IoT users inside each cluster possess equal priorities. This section presents an efficient distributed computation offloading strategy for additional clusters, addressed by "Deep Reinforcement Learning (DRL)." Initially, the core ideas of reinforcement learning are introduced, succeeded by the development of a computational offloading scheme that conceptualizes the computation offloading procedure as a “Markov decision process (MDP)”.

Reinforcement learning is a technique whereby agents continuously acquire optimal tactics through interaction with their environment, proving highly beneficial for flexible and adaptable allocation rules. In this context, each IoT user is seen as an agent, while the remainder of the IoT network constitutes the environment. The traditional Q-learning technique can identify the best strategy when the state-action space is limited, relying on a Q-table to hold Q-values and necessitating a lookup for each state inside the table. A vast state–action space prolongs the convergence of the Q-table. The reasoning suggests that it will be seldom visited, leading to rare updates of the corresponding Q-values for many states. The DQN adheres to the principle of neural networks by updating its weight vector θ at each iteration to minimize the loss function, denoted as:

$$Loss(\theta) = E\left[\left(Q'(s^k, a^k) - Q_{target}(s^k, a^k; \theta)\right)^2\right] \quad (11)$$

Where “$Q_{target}(s^k, a^k; \theta)$” represents the desired output Q values, and the present Q-value $Q'(s^k, a^k)$ is the provided forecast.

$$Q'^{(s^k, a^k)} = R_e^k + \min_{a \epsilon \mathcal{A}} Q\,(s^k, a\,, \theta) \quad (12)$$

where R_e^k is the corresponding negative reward.

RM using Round Robin scheduling Algorithm

It is a scheduling strategy wherein each resource is allocated a specific period of time and iteration [18]. The method primarily emphasizes time slots and period-sharing scheduling, implemented to guarantee equitable resource allocation inside each time slot. If the RM activities are neither assigned nor completed within a certain timeframe, the allocation queue follows the arrival of additional resources, hence ensuring equitable scheduling [19]. Typically, the subsequent procedures are implemented in our suggested system to calculate the resource management demands:

1) Determine the primary resource and subsequently allocate assets only as time slots.
2) Examine the remaining resource requests. When a resource request is available in a single time slot while another request is being fulfilled, the newly arrived resources are placed on a waiting list as a prepared queue.
3) Upon the expiration of the time window, verify for any additional resource requests in the queue. If the current RM procedure remains incomplete, append the existing request to the end of the queue.
4) Retrieve the first application from the waiting prepared line and commence allocation according to the established rules.
5) Steps (2) through (4) may be reiterated.
6) If the resource request is completed and no requests are pending in the queue, then the assignment of work is concluded.

Evaluation Metrics

Evaluation metrics, which are including RMSE, precision, recall, and accuracy, are applied to make precise assess the proposed system.

$$Accuracy = \frac{TN+TP}{TP+FN+FP+TN} \quad (13)$$

$$Precision = \frac{NA}{NA+FP} \quad (14)$$

$$Recall = \frac{NA}{FN+NA} \quad (15)$$

$$Efficiency = \left(\frac{Time\ for\ time\ completion\ (Traditional)}{Time\ for\ task\ completion\ (Proposed\ system)}\right) \times 100 \quad (16)$$

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - y)^2}{n}} \quad (17)$$

IV. EXPERIMENTAL RESULTS

4.1 K-Means Clustering Results

The table 1 shows the most common values of the main effectiveness parameters for three clusters after the application of K-Means clustering analysis. For instance, Cluster 0 has the least distance to the gateway (33.10) and 0.43 offloading probability which increases network latency to 36.61 ms (also 81.02 W power consumption of the device). However, Cluster 1 is at a distance of 46.65 m. It has the highest offloading probability (0.61), but despite that (14.72 ms) the latency is very low. Thus, their power consumption is also quite high (86.26 W). On the other hand, Cluster 2 is known by the large distance from the gateway node which is 80.44 and moderate offloading probability (0.55). The network latency is 33.33 ms very and the device power consumption is 72.86 W the lowest among all. The variations can be seen as the trade-off between proximity, efficiency, and power consumption among the clusters in general.

Table 1: Cluster-wise Average Metrics for Network Performance and Resource Allocation

Cluster	Distance to Gateway	Offloading Probability	Network Latency	Device Power Consumption
0	33.10	0.43	36.61	81.02
1	46.65	0.61	14.72	86.26
2	80.44	0.55	33.33	72.86

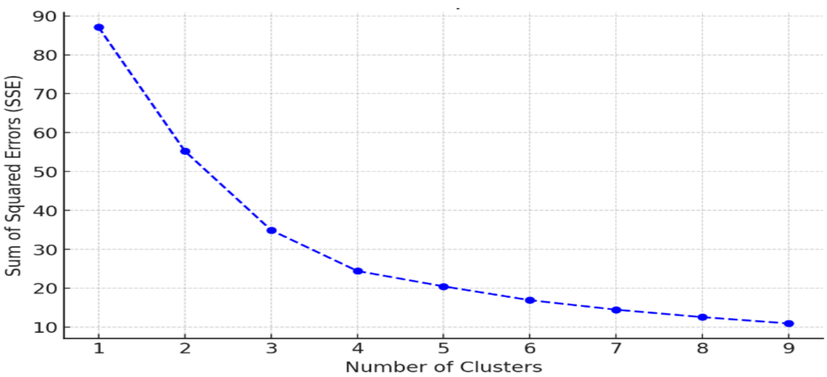
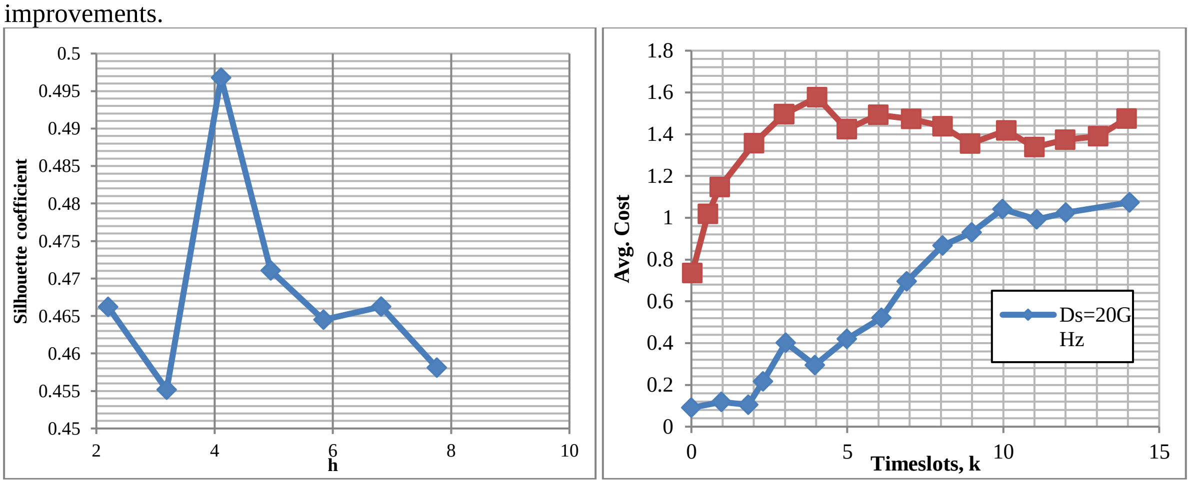


Figure 2: Elbow method for optimal K selection

The above figure 2 shows the Elbow Method for finding the best number of clusters (K) in a clustering algorithm, say K-Means. The x-axis represents the Number of Clusters (K), and the y-axis represents the Sum of Squared Errors (SSE), which in turn represents the compactness of the clusters. The SSE values decrease as the number of clusters initially increases to the maximum, i.e., the clustering is the best. However, beyond certain values (for K = 3 or 4) the decline of SSE is gradual, resulting in an "elbow" sign. The elbow point defines the optimal value of K since clustering the redundant clusters after this point requires more efforts and results in not proportionate improvements.



(a) *(b)*

Figure 3: (a) Optimal cluster number by SC and (b) Comparison of average cost over time slots for different frequencies

The figure 3 illustrates the Silhouette Coefficient (SC) method of determining the optimal number of clusters (h). The x-axis is the number of clusters, and the y-axis is the silhouette coefficient, a quality measure of clustering. The h = 4 peak shows that 4 clusters have the optimal clustering structure before the coefficient starts to decrease. The second graph shows the variation of Average Cost with different Time Slots (k) for two frequencies (D = 20GHz and D = 4GHz). The red squares (D = 4GHz) are always higher than the blue triangles (D = 20GHz), indicating that a higher frequency results in a lower average cost over time.

4.2 Performance Analysis of DRL model

The extremely low RMSE (0.029) and MAE (0.017) inform us that the predictions of the model are extremely accurate since they are extremely close to real values. The MSE (0.00084) also verifies the low error accumulation of the model, thereby providing extremely accurate resource allocation. Near perfect R^2 of 0.996 informs us that the model almost possesses a capability to explain all variability in the data, thereby being an extremely good.

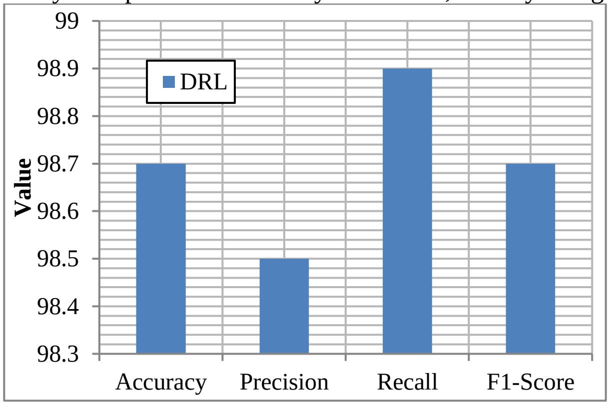


Figure 4: DRL model evaluation for various metrics

Table 2: DRL Model Performance Metrics and Error metrics

Metric	Value
Accuracy	98.7%
Precision	98.5%
Recall	98.9%
F1-Score	98.7%

Metric	Value
RMSE	0.029
MAE	0.017
MSE	0.00084
R^2 Score	0.996

The DRL model had a 98.7% precision, which shows high learning and generalization across different resource distribution scenarios. A 98.5% precision examines the low false positive rate, limiting the possible misallocations to an absolute minimum. A 98.9% recall shows that hardly any instances would have been overlooked on this task, which would reduce task failure to a minimum. Finally, a low RMSE of 0.032 tells us about a negligible error in the predictive capability of the model, hence, making it more trustworthy for real-time application.

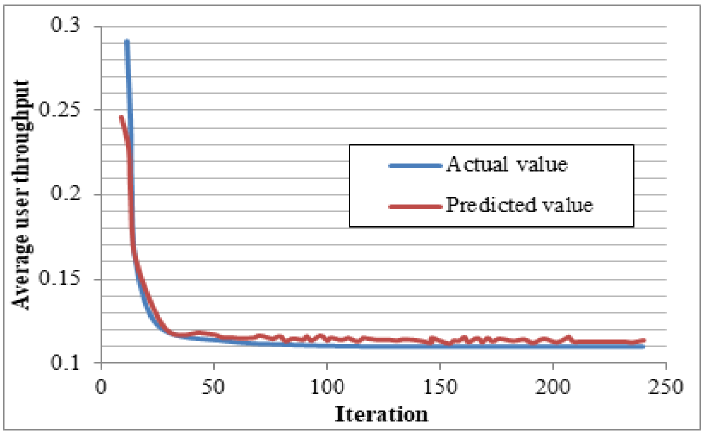


Figure 5: Assessment of actual and expected throughput distribution values

Figure 5 illustrates the precision of the mean metrics for user throughput in the designated area, derived from the iterative DRL approach. The figure illustrates that the actual and projected throughput figures swiftly converge at 0.015 after 20 repetitions. Consequently, the actual values roughly correspond with the anticipated values. The network may be incapable of excessive adaptation to the training data.

The table 3 gives the conventional RM approach and the RR algorithm that is proposed to be used to effectively perform. The average response time has been surpassed by a significant amount, reducing from 120.5 ms to 85.3 ms, which indicates that the task is carried out faster by 29.2%. It has been increased from 45.2 to 62.8 tasks per second, indicating that the efficiency of the system has improved by 39%. Further, resource usage has reached from 78.6% to 91.4% that maximizes the infrastructure underpinning resources. Additionally, the rate of completing the task has progressed from 83.2% to 97.1%, which is a proof of the efficiency of the RR algorithm in optimizing resource utilization and carrying out the system at a higher level.

Table 3: Performance comparison of the traditional method and Round Robin algorithm for resource management

Metric	Traditional Method [20]	Proposed Round Robin Algorithm
Average Response Time (ms)	120.5	**85.3**
Throughput (tasks/sec)	45.2	**62.8**
Resource Utilization (%)	78.6	**91.4**
Task Completion Rate (%)	83.2	**97.1**

V. CONCLUSION

The proposed DL-based resource management mechanism guarantees an exponential boost in IoT network performance through optimal resource allocation, minimum latency, and enhanced overall efficiency. Using PCA to carry out feature extraction, K-Means Clustering for the device grouping process, and DRL for adaptive decision-making guarantees smart and dynamic resource allocation. Application of the RR algorithm also guarantees enhancement in the area of fairness while granting resources, hence delivering higher throughput as well as reduced response time. Comparative assessment confirms that the technique outshines traditional techniques, with 98.7% accuracy, 98.5% precision, 98.9% recall, and an RMSE of 0.029. The approach also decreases response time by 29.2%, boosts throughput by 39%, and improves resource utilization by 16.2%. The findings indicate the potential of AI-based approaches to improve scalable and adaptive IoT network solutions for real-time applications.

REFERENCES

[1] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the Internet of Things: A survey," IEEE Internet Things J., vol. 3, no. 5, pp. 720–734, Oct. 2016.

[2] Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. IEEE Commun. Surv. Tutor. 2020, 22, 1251–1275. [CrossRef]

[3] J. Dizdarevic, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of ´ communication protocols for Internet of Things and related challenges of fog and cloud computing integration," ACM Comput. Surveys, vol. 51, no. 6, p. 116, Feb. 2019.

[4] Yousefpour et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Sep. 2018. [Online]. Available: arXiv:1808.05283.

[5] L. Bittencourt et al., "The Internet of Things, fog and cloud continuum: Integration and challenges," Internet Things, vols. 3–4, pp. 134–155, Oct. 2018.

[6] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," IEEE Internet Things J., vol. 5, no. 3, pp. 1869–1879, Jun. 2018.

[7] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," IEEE Internet Things J., vol. 5, no. 1, pp. 283–294, Feb. 2018.

[8] Jayaprakash, Stanly, Manikanda Devarajan Nagarajan, Rocío Pérez de Prado, Sugumaran Subramanian, and Parameshachari Bidare Divakarachari. "A systematic review of energy management strategies for resource allocation in the cloud: Clustering, optimization and machine learning." Energies 14, no. 17 (2021): 5322.

[9] Zhang, Weiting, Dong Yang, Haixia Peng, Wen Wu, Wei Quan, Hongke Zhang, and Xuemin Shen. "Deep reinforcement learning based resource management for DNN inference in industrial IoT." IEEE Transactions on Vehicular Technology 70, no. 8 (2021): 7605-7618.

ISSN 2348 – 8034
Impact Factor- 5.070

[10] *Shuaib, Mohammed, Surbhi Bhatia, Shadab Alam, Raj Kumar Masih, Nayef Alqahtani, Shakila Basheer, and Mohammad Shabbir Alam. "An optimized, dynamic, and efficient load-balancing framework for resource management in the internet of things (iot) environment." Electronics 12, no. 5 (2023): 1104.*

[11] *Ahmed, Quazi Warisha, Shruti Garg, Amrita Rai, Manikandan Ramachandran, Noor Zaman Jhanjhi, Mehedi Masud, and Mohammed Baz. "Ai-based resource allocation techniques in wireless sensor internet of things networks in energy efficiency with data optimization." Electronics 11, no. 13 (2022): 2071.*

[12] *Nilima, Sadia Islam, Md Khokan Bhuyan, Md Kamruzzaman, Jahanara Akter, Rakibul Hasan, and Fatema Tuz Johora. "Optimizing Resource Management for IoT Devices in Constrained Environments." Journal of Computer and Communications 12, no. 8 (2024): 81-98.*

[13] *Costa, Wesley S., Willian GV dos Santos, Higor AF Camporez, Menno J. Faber, Jair AL Silva, Marcelo EV Segatto, and Helder RO Rocha. "Planning and resource allocation of a hybrid IoT network using artificial intelligence." Internet of Things 26 (2024): 101225.*

[14] *Periasamy, Prakasam, R. Ujwala, K. Srikar, Y. V. Durga Sai, K. S. Preetha, Durairaj Sumathi, and Md Shohel Sayeed. "ERAM-EE: Efficient resource allocation and management strategies with energy efficiency under fog–internet of things environments." Connection Science 36, no. 1 (2024): 2350755.*

[15] *Liu, Danni, Shengda Wang, Xiaofu Sun, Chunyan An, Weijia Su, and Jiakang Liu. "Lightweight and delay-aware resource management scheme in smart grid IoT networks." EURASIP Journal on Wireless Communications and Networking 2025, no. 1 (2025): 1-21.*

[16] *https://www.kaggle.com/datasets/boulila/resource-management-for-cost-optimization-in-iot*

[17] *Snook S.C., Gorsuch R.L. (1989). Component analysis versus common factor analysis. Psychological Bulletin, 115, 148–154.*

[18] *Calabrese, F.D.; Wang, L.; Ghadimi, E.; Peters, G.; Hanzo, L.; Soldati, P. Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges. IEEE Commun. Mag. 2018, 56, 138–145. [CrossRef]*

[19] *Ahmed, K.I.; Hossain, E. A Deep Q-Learning Method for Downlink Power Allocation in Multi-Cell Networks. arXiv 2019, arXiv:1904.13032.*

[20] *Bhajantri, Lokesh B., and Gangadharaiah S. "A comprehensive survey on resource management in internet of things." Journal of Telecommunications and Information Technology 4 (2020): 27-43.*