Enhancing Cloud Resource Allocation with a Deep Learning-Based Framework

Suhad Ibrahim Mohammad[1*], Ziyad Tariq Mustafa Al-Ta'i[2]
[1]*Department of Computer Science, College of Science ,University of Diyala, Iraq*
[2]*Department of Computer Science, College of Science ,University of Diyala, Iraq*
scicomphd222307@uodiyala.edu.iq

Abstract

Cloud computing has transformed the modern world of computing by enabling the provisioning of resources that are on demand and scalable. Nevertheless, the issue of efficient resource allocation persists due to unpredictable demand fluctuations and heavy duties. This paper investigates the allocation of cloud resources through the use of deep learning (DL) models, specifically Convolution Neural Networks (CNNs), Gated Recurrent Networks (GRNs), and Long Short Term Memory (LSTM) networks. CNNs acquire spatial patterns from cloud workload data, while GRNs acquire short-term resource usage dependency patterns. By acquiring long term patterns in work load variation, LSTMs further improve the accuracy of their predictions. Our proposed framework optimizes cloud resource allocation using these models, enhancing overall system performance, reducing energy consumption, and reducing latency. Experimental evidence indicates that our proposed deep learning framework is more precise and adaptable than conventional methodologies.

Keywords: Cloud computing, Deep learning, Resource allocation, Virtualization

تحسين تخصيص موارد السحابة باستخدام إطار عمل قائم على التعلم العميق

سهاد ابراهيم محمد[1]*, د. زياد طارق مصطفى[2]
[1]علوم حاسوب, كلية العلوم,جامعة ديالى, ديالى, العراق
[2]علوم حاسوب, كلية العلوم,جامعة ديالى, ديالى, العراق

الخلاصة

أحدثت الحوسبة السحابية تحولاً جذرياً في عالم الحوسبة الحديث، إذ أتاحت توفير موارد قابلة للتوسع عند الطلب. ومع ذلك، لا تزال مشكلة التخصيص الفعّال للموارد قائمةً بسبب تقلبات الطلب غير المتوقعة والمهام الشاقة. تبحث هذه الورقة البحثية في تخصيص موارد السحابة من خلال استخدام نماذج التعلم العميق (DL)، وتحديداً الشبكات العصبية التلافيفية (CNNs)، والشبكات المتكررة المبوّبة (GRNs)، وشبكات الذاكرة طويلة المدى قصيرة المدى (LSTM). تكتسب الشبكات العصبية التلافيفية أنماطاً مكانية من بيانات عبء العمل السحابي، بينما تكتسب شبكات GRNs أنماط اعتماد على استخدام الموارد قصيرة المدى. ومن خلال اكتساب أنماط طويلة المدى في تباين عبء العمل، تُحسّن شبكات الذاكرة طويلة المدى قصيرة المدى دقة تنبؤاتها. يُحسّن إطار عملنا المقترح تخصيص موارد السحابة باستخدام هذه النماذج، مما يُحسّن الأداء العام للنظام، ويُقلّل استهلاك الطاقة، ويُقلّل زمن الوصول. تُشير الأدلة التجريبية إلى أن إطار عمل التعلم العميق المُقترح لدينا أكثر دقةً وقابليةً للتكيف من المنهجيات التقليدية.

1. Introduction

Cloud computing has transformed the face of information technology (IT) by offering elastic, dynamic, and cost effective ways for individuals and businesses to access computing resources [1-5]. Because cloud computing can offer on demand resources like storage, computing power, and applications, consumers can carry out activities without the complexity of dealing with the hardware. However, as cloud based services continue to gain popularity; effectively managing resource allocation remains an ongoing concern.

The concern becomes more complicated considering the variable nature of workloads, varied application needs, and real time scale requirements [6]. Traditional cloud environments use predefined rules or heuristics for resource allocation, which fails with massive, complex, and dynamic data. Legacy approaches struggle to handle cloud applications' dynamics, resulting in resource underutilization, delay, and over provisioning. Higher expenses, lost energy, and ineffective system performance result from inefficiency. Intelligent, adaptive systems that scan, forecast, and dynamically provision resources according to changing needs in real time are being promoted to address these concerns [7], [8]. Cloud computing relies on virtualization to create virtualized representations of physical hardware for efficiency, scalability, and flexibility. It revolutionizes IT resource deployment and consumption by allowing numerous virtual machines on a single physical machine [9], [10], optimizing system performance, energy consumption, and data centre resources [11], [12]. Deep learning (DL) is better at comprehending and processing complex, high dimensional data. Deep learning has been applied to cloud computing resource management. CNNs, GRNs, and LSTM networks have extracted complex patterns and relationships from huge data. CNNs can identify geographical patterns and properties of data, making them ideal for observing and analyzing cloud resource utilization patterns. GRNs are suitable for temporal modeling of dependencies and correlation of short term resource use data. Finally, LSTMs learned long term dependencies, which is needed to accurately forecast resource demands from previous data [13], [14].

This article investigates the utilization of deep learning models in the Cloud Resource Allocation Strategy. We concentrate on the real time evaluation and management of resource requirements by employing CNNs, GRNs, and LSTMs. The proposed framework is designed to improve traditional methods by offering a system that is intelligent and adaptable, capable of dynamic allocation in response to changing conditions. In addition to increasing performance and reducing operating costs, this function also emphasizes the reduction in energy consumption, an important factor in the cloud calculation environment where energy efficiency is an important indicator of economic and environmental stability. We aim to guarantee that cloud services are energy-efficient and cost-effective and cannot only be scalable by optimizing resource allocation in response to immediate demand but also by guessing future resource requirements.

Many research investigations have found that DL approaches can provide intelligent and adaptive resources to address such difficulties [15], [16]. Future predictions and data driven decisions can be made using observed and unsecured learning, historic billing habits, and resource usage trends. Using the ML model, cloud providers may estimate resource demand, dynamically assign resources, and change.

2. Deep Learning Applications

Deep learning (DL) is a technique to realize artificial intelligence (AI) by realizing multilayered neural networks and it requires computational resources and massive training data. Some existing research points out that it is possible to find and deploy Deep Learning models to learn new updates when it is built on existing networks. However, each data holder only has some input data and a local model learned from it. It is the same concept as federated learning, and uses this paradigm to propose a novel decentralized Federated Transfer Learning framework using block chain that does not involve the data exchange itself. Learning tasks of the new example classes can be obtained in this framework, and learning the variety of each class is a correct example. The PLN and the SLN use the data holders to learn the class change of the pre trained large network through the proposed block chain-based protocol. Moreover, a class wise sampling procedure is proposed to prevent the one data holder from acquiring general knowledge about the new classes. Wide and diversified applications in terms of the number of different approaches with Deep Learning (DL) associated with block chain technology enhance the secure distributed computation's train ability or assist the context of DL, namely (1) real time arrhythmia classification and (2) miner node selection during block generation [17]. DL is as an evolving subset of Machine Learning (ML) plays a vital role due to a state-of-the-art performance in a variety of applications. It becomes increasingly complex to deliver high quality scaling models in place of very expensive human experts. Although it has achieved large success at different levels for several applications, the feature extraction issue is common. It is challenging when making it similar to the output task, and it requires many additional tasks or a lot of efforts with lots of examples until good results are achieved. When learning the representation, the model is applied following standard training methods and learning about the data. This should produce some semantically meaningful parameters used as a base for the task in order to extract the HA. Beyond extracting features, further supervised learning across steps is required. Distributed representation of words using less-dimensional vectors called word embeddings. These are learned based on the local context using shallow architecture. Alternative to model pre-training, the feature representations can be learned using self supervising learning [18].

Deep learning (DL) has revolutionized artificial intelligence (AI) by allowing machines to automatically learn complex patterns, interdependencies, and relationships from vast amounts of raw data without the need for human feature extraction [19]. DL is powering AI research and applications by automatically extracting complex patterns in data, revolutionizing healthcare, finance, autonomous systems, and robotics. Gated Recurrent Units (GRU), Convolution Neural Networks (CNN), and Long Short Term Memory (LSTM) networks are some of the most popular deep learning architectures that solve specific ML issues [20]. Convolution Neural Networks (CNNs) are a cornerstone of face identification, medical image analysis, autonomous vehicles, and surveillance tracking because of the fact that they are very good at picking up visual patterns and spatial hierarchies in data through convolution layers. Long Short Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) are specifically designed to handle sequential data and succeed at time related tasks like NLP, speech recognition, sentiment analysis, and time series forecasting. They use gating mechanisms to remember and forget selectively in order to keep long term data sequence dependence in the traditional recurrent neural networks (RNNs) with vanishing gradients. Deep learning algorithms process structured and unstructured data and enhance AI powered applications like virtual assistants, real time language translation, AI powered medical diagnosis, and financial risk analysis due to their longevity. DL has also progressed the creation of deep fakes, visual style transfer, and music generation. Greater processing power, massive data sets, and optimization methods power deep learning's unabated growth, transforming artificial intelligence. Thus, modern AI research relies on deep learning to advance reinforcement learning, explainable AI, and human computer interaction, impacting intelligent automation and decision making systems in industries (see Figure 1).

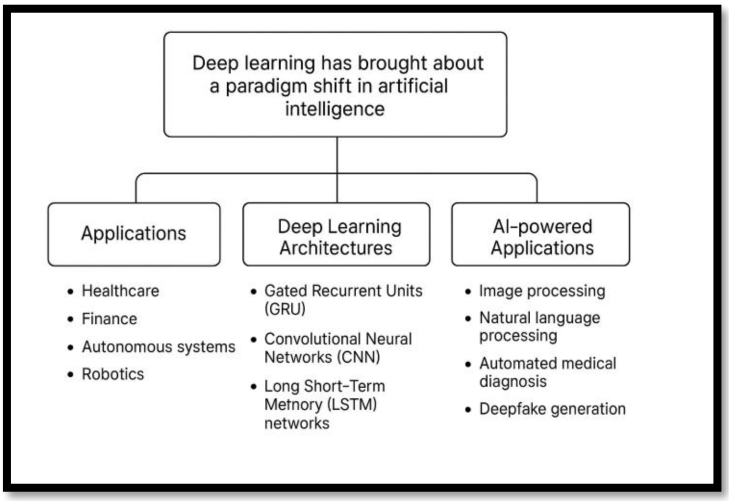

Deep learning has brought about a paradigm shift in artificial intelligence
Applications
Deep Learning Architectures
AI-powered Applications
• Healthcare
• Finance
• Autonomous systems
• Robotics
• Gated Recurrent Units (GRU)
• Convolutional Neural Networks (CNN)
• Long Short-Term Metnory (LSTM) networks
• Image processing
• Natural language processing
• Automated medical diagnosis
• Deepfake generation

Figure -1 Deep Learning architecture

3. Deep Learning (DL) Methods

The development of deep learning has revolutionized artificial intelligence by allowing robots to learn difficult patterns and associations from data. Gated Recurrent Units (GRU), Convolution Neural Networks (CNN), and Long Short Term Memory (LSTM) are the three types of deep learning architecture used most frequently. Computer vision, natural language processing (NLP), and time series forecasting are all areas that heavily rely on the application of these models because they can process all forms of data and tasks [21].

3.1 Convolution Neural Networks (CNNs)

CNNs are specially crafted networks to process spatial and image data. They possess convolution layers, which apply filters to select characteristic features such as shapes, textures, and edges [22]. CNNs compare well with ordinary fully connected networks in that they reduce parameters using shared weights, which is appropriate for large sized image recognition problems. CNNs find extensive applications in image classification, face detection, object detection, and medical imaging. They also enable applications like autonomous vehicles and video processing [23].

3.2 Long Short-Term Memory (LSTM) Networks

LSTM is a variation of Recurrent Neural Network (RNN) which has specifically been developed for application in sequence data such as text, speech, and time series data. RNNs suffer from the vanishing gradient problem which detains RNNs from memorizing long term relationships. LSTMs have universal applications in speech-to-text recognition, machine translation, chatbots, prediction of financial output, and sentiment analysis. As they have to maintain the relationships due to the long term memory, they are highly capable to handle complex sequence tasks [24].

3.3 Gated Recurrent Unit (GRU)

GRUs are analogous to LSTMs but with fewer computations and simpler to understand. GRUs can be applied in natural language processing (NLP), time series prediction, and speech recognition. GRUs gives the same kind of output as LSTMs but with fewer training periods and parameters. Thus, they are suitable for use in real-time when there is a problem of lightness of computation [25].

There are many applications of DL techniques. For image based solutions, CNNs are suitable, whereas LSTMs and GRUs are suitable for sequence data. Whereas LSTMs are better suited for longer sequences, GRUs need less work in terms of processing and are faster compared to LSTMs. Deep learning, being based largely on such architectures, allows for artificial intelligence application spanning a larger timeframe.

Figure 1 shows a two stage procedure for training and testing deep learning models, including CNNs, LSTMs, GRUs, and ensembles. These models are used on a dataset combining Google Cloud Jobs (GoCJ) and Monte Carlo simulation data. The solution addresses a difficult resource allocation and workload forecasting challenge in virtualized cloud systems. The Google Cloud Jobs data set accurately depicts cloud infrastructure usage. Monte Carlo simulation is employed to model and simulate unpredictable scenarios or system reactions. Deep learning models identify patterns from this data during training to create long term predictions about system behavior and resource requirements. The evaluation method tests these models to see how well they estimate workload demands and distribute resources appropriately in a cloud system. This approach determines which resource forecasting models are most predictive, accurate, and operationally effective.

4. Methodology

The proposed framework employs deep learning techniques for cloud workload prediction and resource optimization. It follows a structured approach that includes data generation, preprocessing, model training, and performance evaluation. Our Proposed Framework is listed as below:

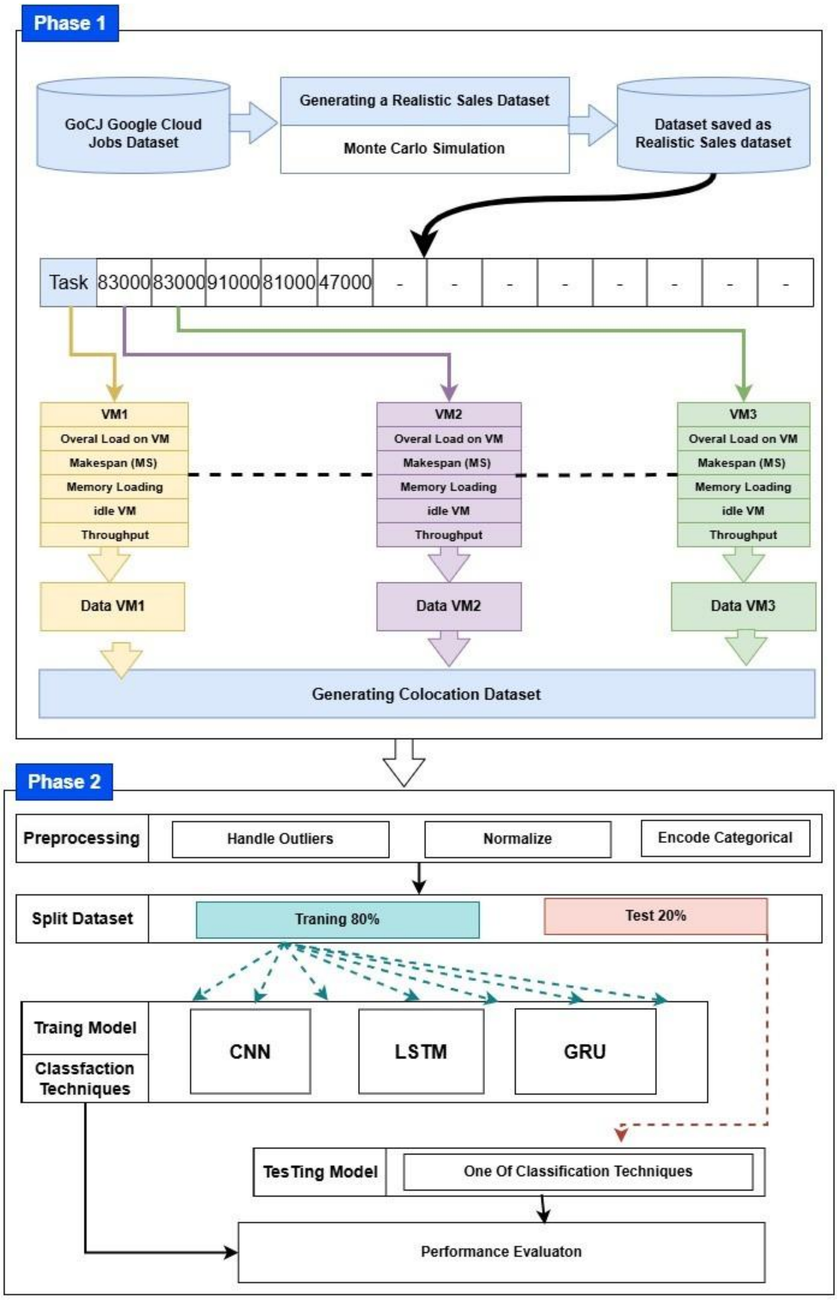


Figure – 2 Proposed Framework by applying deep learning with cloud

4.1. Phase 1: Data Generation

4.1.1. Google Cloud Jobs Dataset & Monte Carlo Simulation:

A Monte Carlo Simulation is incorporated into the dataset, which is obtained via Google Cloud Jobs (GoCJ). This simulation is used to build a realistic sales dataset, which is then saved for future study. This dataset is made up of a number of different jobs, each of which is uniquely recognised by an identifier (for example, 83000, 83009, etc.). The parameters are (arrival time, standard deviation and mean) this makes it possible to easily track and reference individual tasks within the dataset.

4.1.2. Virtual Machine (VM) Workload Data Collection:

The tasks are distributed across multiple Virtual Machines (VMs), specifically labeled as VM1, VM2, and VM3, each of which plays a critical role in executing and processing the assigned workload. To monitor and evaluate the performance of each VM, several key metrics are recorded throughout the execution of tasks. These metrics include the overall load on the VM, which reflects the total processing power being used at any given time; make span (MS), which is the total execution time required to complete a task or set of tasks; memory loading, indicating how much memory the VM utilizes during task execution; idle VM time, which measures the periods when the VM is not in use or underutilized; and throughput, which tracks the amount of work completed by the VM in a given time frame. By gathering these performance metrics, a detailed understanding of the resource demands and efficiencies of each VM is achieved. The data collected from each VM is stored separately as VM1 Data, VM2 Data, and VM3 Data, making it possible to analyze the performance of each machine individually. Once the data from the three VMs has been captured, it is then integrated and combined into a single comprehensive Collocation Dataset. This aggregated dataset allows for further processing, such as analyzing how tasks can be better distributed across VMs, identifying potential bottlenecks, optimizing resource allocation, and improving the overall efficiency of cloud infrastructure, particularly in a virtualized environment where resource management is crucial for maintaining performance and minimizing cost.

4.2. Phase 2: Data Processing & Model Training

4.2.1. Data processing Training:

Pre-processing of data begins with outliers, where the abnormalities in the data are identified and removed to ensure that the model is trained using clean and reliable data. Subsequently, data is normalized, i.e., scaled to bring all values to a consistent range, so features with larger numerical ranges do not dominate the model's learning process. Categorical data is then converted into numeric form so that non numeric attributes become model training worthy. The preprocessed dataset is then split into three datasets: training set (80%), which will be used to train the model; validation set (20%), for model tuning and hyper parameter optimization, with the preprocessed data, various deep-learning models are trained. They are CNN (Convolution Neural Network), which is largely used for extracting spatial features; LSTM (Long Short-Term Memory), a recurrent neural network (RNN) model that is used for sequence-based prediction handling; and GRU (Gated Recurrent Unit), an optimized version of LSTM, which is utilized for sequential data.

4.2.2. Testing & Performance Evaluation

Once the models are trained, they undergo a rigorous performance testing phase to verify whether they can make accurate predictions. During this phase, classification techniques are employed to quantify the accuracy of the model, helping determine how accurately each model can classify data and predict results based on the given input features. Various performance measures such as precision, recall, and F1-score can be measured to provide a comprehensive study of the performance of each model. On the basis of the experiments, the optimal model that performs with the best accuracy and reliability is chosen, which also possesses the greatest potential to generalize new, unseen data and perform best in real world applications. This model is therefore selected to deploy.

5. Experimental and Results

5.1 Datasets

The GoCJ dataset is a valuable resource for cloud work- load prediction and resource allocation. It offers differing job sizes that can be generated through formulas in an Excel document as shown a table 1. Monte Carlo simulation introduces randomness to the dataset to offer more representative cloud workload environments. The data set is crucial for task distribution between virtual machines (VMs) in a manner that maximizes the allocation of resources through key performance indicators such as virtual machine selection, load distribution, make span, throughput, wait time, and system load. It maximizes task scheduling, prevents VM overloading, and optimizes cloud system performance.

5.2 Convolution neural network (CNN)

CNNs are best at local spatial feature extraction a dropout layer of 0.3 is employed to reduce over fitting by randomly dropping 30% of the neurons while training. Next, the model applies a 1D convolution layer with kernel size 3x64x64 and utilizes 64 filters. Here, ReLU activation is employed as well to enable the model to learn sequence pattern features. The following MaxPooling1D layer helps down sample by a factor of 2, lowering dimensions without losing the most important features. After the convolution process, two fully connected (dense) layers are employed in the model. The first dense layer has 128 units using the ReLU activation,

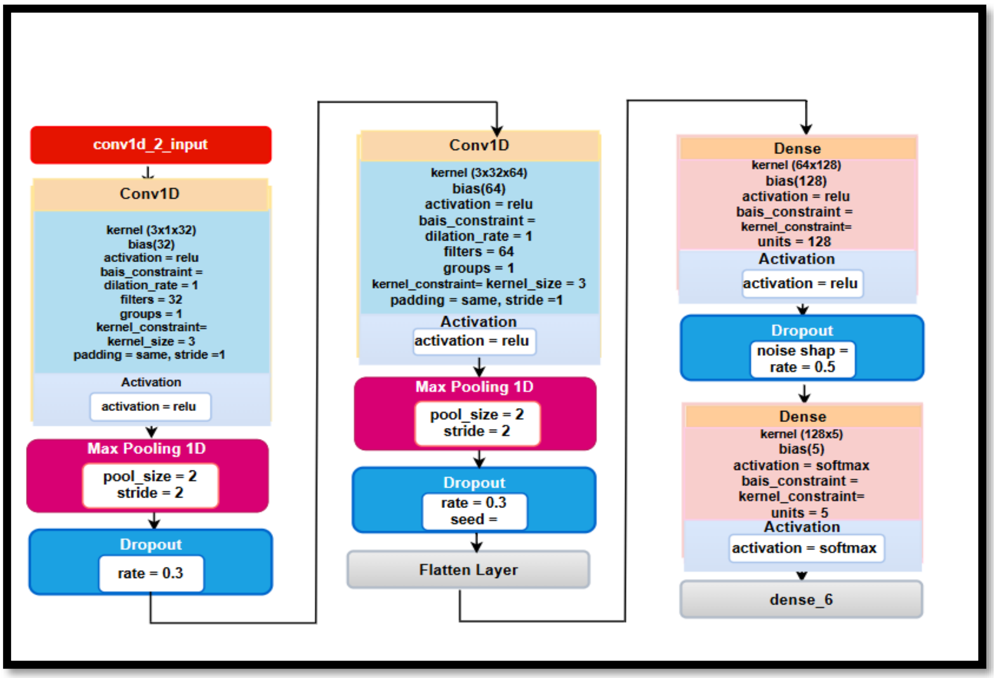


Figure -3 Convolution neural network (CNN)

5.3 Long Short Term Memory (LSTM)

LSTM layers to take advantage of the power of both models in processing sequential data whereas LSTMs can model long-term temporal dependencies and thus the mix here is very much suitable for applications involving both spatial and sequential data, such as time series prediction and sequence classification. The sequential data is input to the input layer, which is then input to the LSTM layer. The LSTM layer is configured with a kernel size of 1x256 and recurrent kernel size of 64x256 to learn temporal dependencies in the data. It has the ReLU activation function for the kernel and the sigmoid activation function for the recurrent kernel with return sequences equal true so that the LSTM outputs a sequence of values to be processed further. Following the LSTM, a dropout layer of 0.3 is employed to reduce over fitting by randomly dropping 30% of the neurons while training.

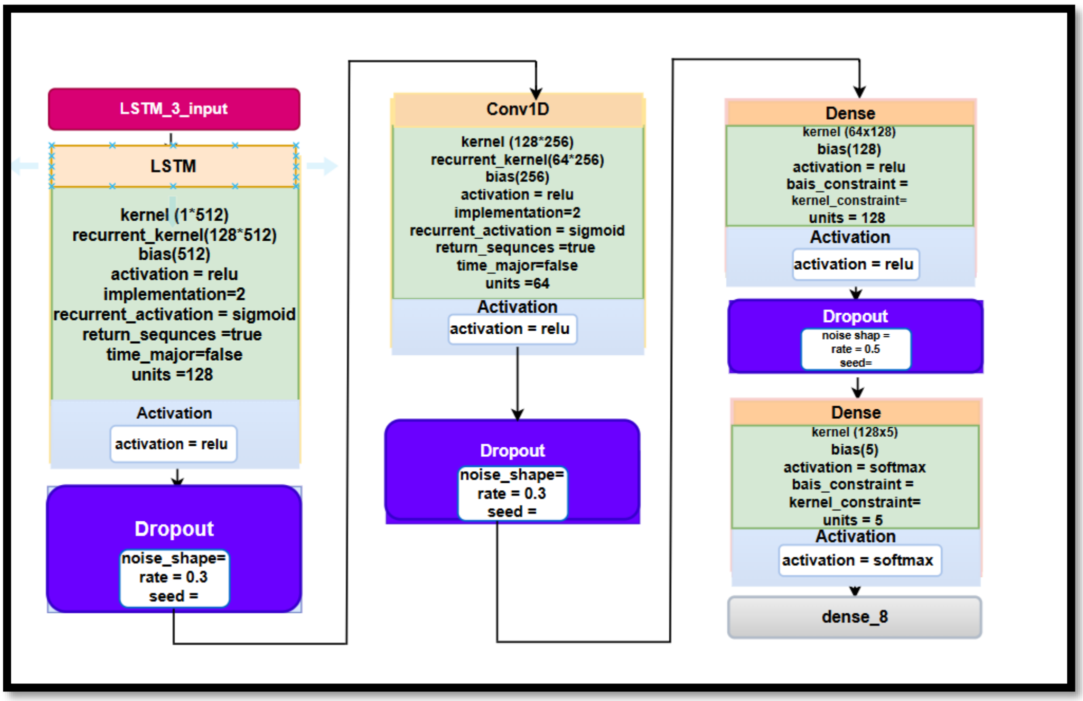


Figure -4 Long Short Term Memory (LSTM)

5.4 Hybrid GRU + CNN Model for Sequential Data

The model presented combines GRU and CNN to process sequential data efficiently. GRU is a variant of LSTM and is highly efficient in processing temporal dependencies in sequential data, while CNNs are optimized to extract local features. Combining both components, the hybrid model becomes highly effective in operations such as time-series prediction, speech-to-text, and sequence classification. The architecture begins with an input layer accepting sequential data. Then comes the GRU layer where the kernel size is 1x192 and recurrent kernel size is 64x192, so that the GRU can capture temporal relationships within the sequence. The GRU uses ReLU as the activation function on the kernel and recurrent activation as sigmoid. The model is initialized with return_sequences=True so the GRU will return the entire sequence of data rather than the final state, that way all the information from each time step will pass through to the subsequent layers. A Dropout layer with dropout=0.3 is then employed to combat over fitting by randomly disabling 30% of the neurons when training. Next, a 1D Convolution layer is introduced with kernel size 3x64x64 and 64 filters. In this case, the ReLU activation is employed, and the convolution layer is followed by MaxPooling1D. The max-pooling operation reduces the sequence dimensionality by half; with pool size 2and strides 2, so that the most important features are retained. The second Dropout layer at a rate of 0.3 is employed to again prevent over fitting. The Flatten layer is then applied to flatten the data into a 1D vector. The model then flows through two dense layers. The first dense layer contains 128 units with the activation function being ReLU. The second dense layer has 5 units with softmax activation, giving a probability distribution over five target classes. Figure 4 present combining the GRU's temporal dependency learning ability with the CNN's local feature learning capability, this hybrid model is able to effectively learn both the sequence level and local feature patterns of the data and is thus suited to a large variety of sequence-based prediction problems.

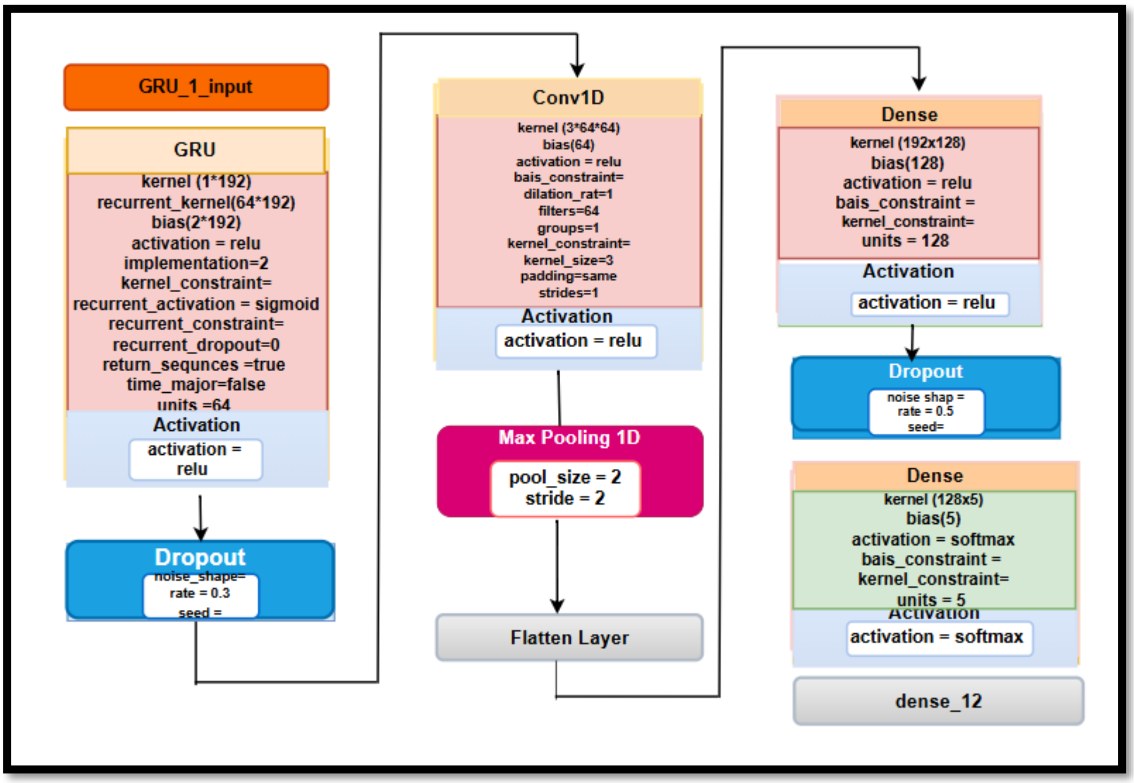


Figure -5 Represents combining a GRU with CNN

Figure 6 presents a 3D scatter plot illustrating the relation- ship between Job Size, Arrival Time, and Service Time for 10,000 tasks. The X-axis represents the job size in megabytes, the Y-axis denotes the arrival time in seconds, and the Z- axis indicates the execution time required for each task. The distribution of data points visually captures the correlation between these parameters, making the plot useful for analysis, interpretation, and statistical evaluation of task execution patterns.

Figure 3 presents two plots depicting the job size distribution and task arrival times. The left plot highlights job sizes, which are primarily concentrated around lower values, while the right plot visualizes task arrival times along with their unique IDs. These plots offer valuable insights into task frequency, size distribution, and scheduling patterns within the dataset.

5.5 Evaluation Metrics

In many fields where precise predictions are crucial, performance measurements are needed to evaluate categorization models. Precision is the computation of, from all the projected positive examples, precisely identified positive ones. When the cost of false alarms is high, precision a gauge of the model’s capacity to prevent false positives is more important. Sensitivity or true positive rate, sometimes known as recall, gauges the model’s identification of the true positive case count. It shows how well the model detects positive examples and is therefore important when, e.g., disease diagnosis misses a positive case has major repercussions. The harmonic mean of memory and accuracy, the F1-score offers a fair evaluation considering false positives and false negatives. It comes especially helpful in cases of unequal class distribution. Real negative rate, there are feature (over all load, load memory, make span , throughput and wait) that describe sample how distributed the tasks in virtual machine that describe by class.

$$Peccision\ = \frac{TP}{TP+FP}\text{--------------------(1)}$$

$$Recall\ = \frac{TP}{TP+FN}\text{--------------------(2)}$$

$$F1 - Score\ = 2 * \frac{precsion*Recall}{Precsion*Recal}\text{--------------------(3)}$$

$$S\,pecificity = \frac{TN}{TN+FP}\text{--------------------(4)}$$

Table1- GoCJ Excel worksheet generator

df_OL	Lm_p	makespan	Throughput	waite	df_task_save	Class
132.1681	87.4339	6.971751	198.779	0	40000	V_Memory0
132.1681	87.4339	6.971751	198.779	0	40000	V_Memory0
157.4679	87.4339	6.971751	198.779	0	40000	V_Memory0
105.9448	98.3628	6.966102	21.606	0	45000	V_Memory0
51.63073	60.1115	6.985876	1261.816	0	27500	V_Memory0
26.10203	60.1115	6.985876	1261.816	0	27500	V_Memory0
26.10203	60.1115	7	0	0	27500	V_Memory0
110.943	98.3628	6.966102	21.606	0	45000	V_Memory0
108.9243	60.1115	6.985876	1261.816	0	27500	V_Memory0
47.0887	32.7891	7	0	0	15000	V_Memory0
49.27735	32.7891	7	0	0	15000	V_Memory0
35.56492	60.1115	6.985876	1261.816	0	27500	V_Memory0
35.56492	60.1115	7	0	0	27500	V_Memory0
35.56492	60.1115	7	0	0	27500	V_Memory0
144.4884	98.3628	6.966102	21.606	0	45000	V_Memory0
103.3578	98.3628	6.966102	21.606	0	45000	V_Memory0
116.4144	98.3628	6.966102	21.606	0	45000	V_Memory0
55.54988	60.1115	6.985876	1261.816	0	27500	V_Memory0
137.2394	98.3628	6.966102	21.606	0	45000	V_Memory0
39.64868	32.7891	7	0	0	15000	V_Memory0
21.28185	32.7891	7	0	0	15000	V_Memory0

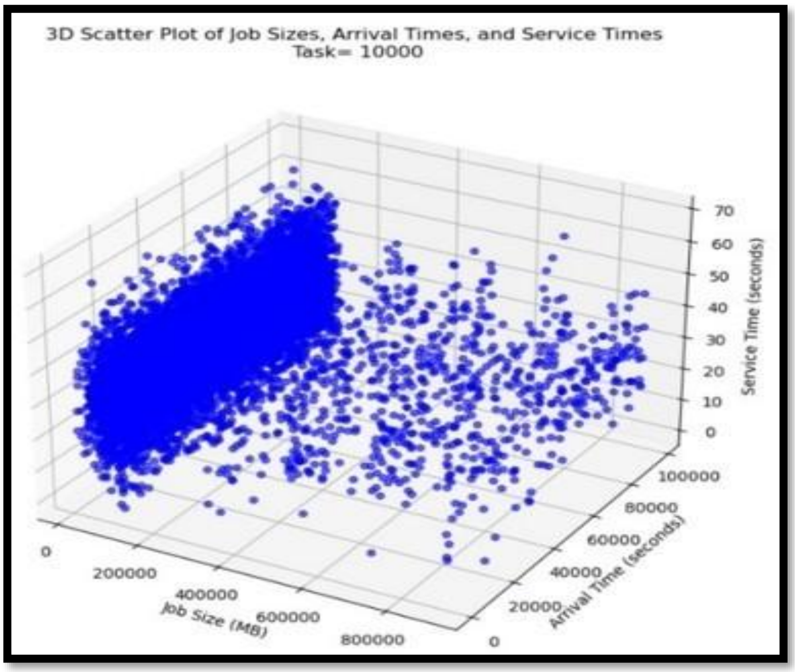


Figure -6 3D scatter plot size, arrival times, and service times

6. Results and Discussion

The deep learning based cloud resource provisioning performance was compared with a robust dataset developed from Google Cloud Jobs (GoCJ) and fine-tuned using Monte Carlo simulations. The dataset was preprocessed via outlier treatment, data normalization, and categorical feature encoding to prepare it properly for training the model. In addition, the Monte Carlo simulation enhanced the dataset's resilience to allow the models to accept various probabilistic scenarios and better replicate real cloud resource requirements. The models performed outstandingly in workload management, with the models performing outstandingly in resource prediction and optimization. The data was then split into train, validation, and test sets in the ratios of 80The models were evaluated on various classification metrics, including precision, recall

Table 2- F1-value performance of three algorithms

Datasets	CNN	GRU	LSTM
$V_{memory}0$	0.99	0.98	1
$V_{memory}1$	0.99	0.97	1
$V_{memory}2$	0.99	0.97	0.99
$V_{memory}3$	0.96	0.98	0.99

F1-score, and specificity, to assess their accuracy in predicting cloud workload fluctuations and resource allocation optimization. Precision is the number of correctly predicted positive instances out of all the predicted positives. At the same time, recall (sensitivity) measures the model's performance in identifying actual positive instances. The F1-score, as a harmonic means of recall and precision, ensures that the assessment remains balanced, particularly in a class imbalance scenario. Furthermore, specificity evaluates the model's ability to label the negative instances accurately and provides additional insight into its performance in general.

Among the models experimented with, all deep learning techniques were performed (see Figure 7). The Figure effectively resolved sequence learning and feature extraction, benefiting from the strength of CNN, LSTM, and GRU models. The techniques improved the model's ability to capture short-term fluctuations and long-term trends of cloud workloads, resulting in more precise resource allocation decisions (see tables I, II, III and IV). Lastly, the results emphasize the need to incorporate diverse learning strategies to enhance predictive performance and optimize cloud computing performance.

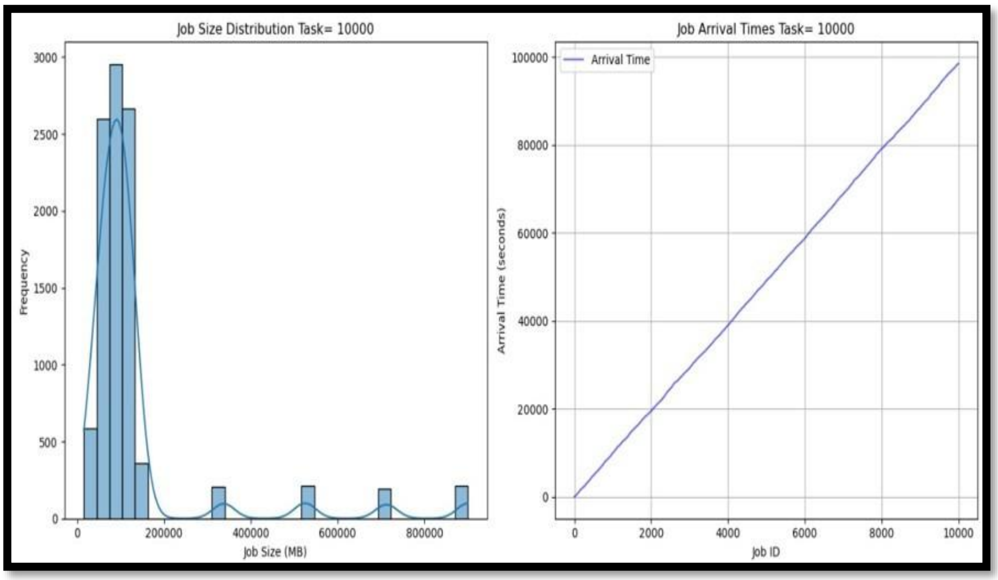
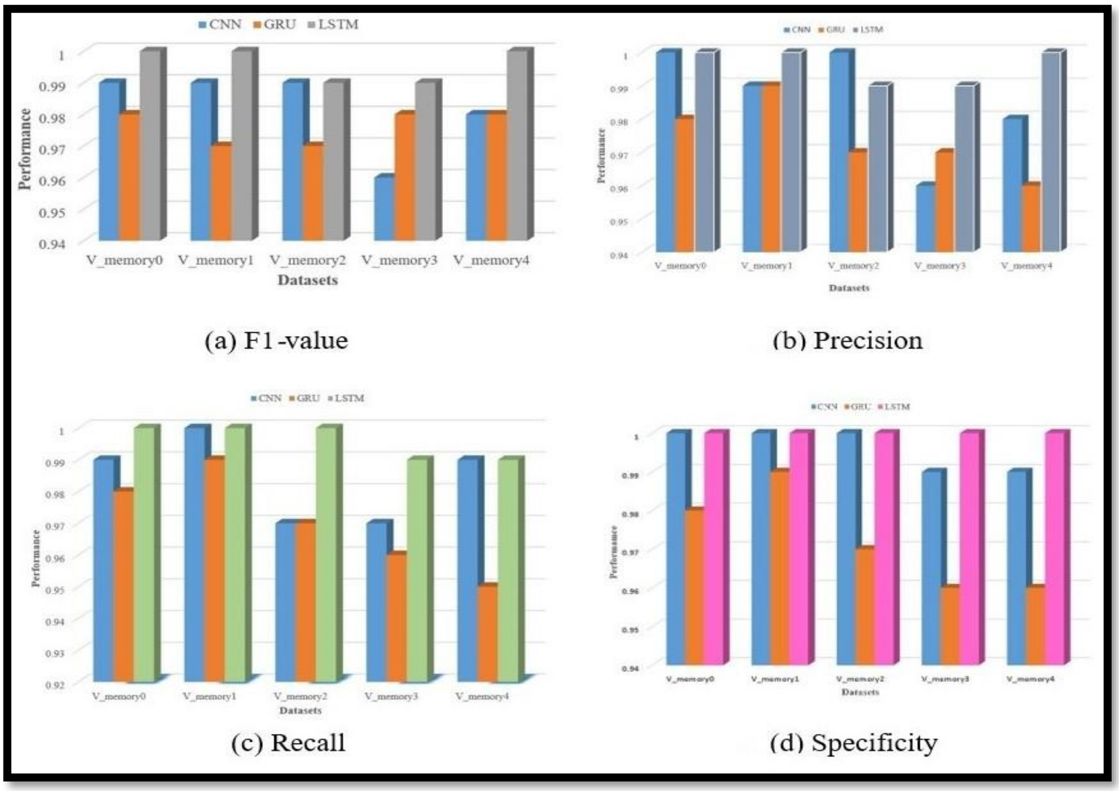


Figure -7 Left Plot: Job Size Distribution (Task = 10000) Right Plot: Job Arrival Times (Task = 10000)



(a) F1-value

(b) Precision

(c) Recall

(d) Specificity

Figure -8 The performance of three algorithms

Table 3- Precision performance of three algorithms

Datasets	CNN	GRU	LSTM
$V_{memory0}$	0.99	0.98	1
$V_{memory1}$	0.99	0.99	1
$V_{memory2}$	1	0.97	0.99
$V_{memory3}$	0.96	0.97	0.99
$V_{memory4}$	0.98	0.96	1

Table 4- Recall performance of three algorithms

Datasets	CNN	GRU	LSTM
V	0.99	0.98	1
$V_{memory1}$	1	0.99	1
$V_{memory2}$	0.97	0.97	1
$V_{memory3}$	0.97	0.96	0.99
$V_{memory4}$	0.99	0.95	0.99

Table 5- Specifity performance of three algorithms

Datasets	CNN	GRU	LSTM
$V_{memory0}$	1	0.98	1
$V_{memory1}$	1	0.99	1
$V_{memory2}$	1	0.97	1
$V_{memory3}$	0.99	0.96	1
$V_{memory4}$	0.99	0.96	1

7. Conclusion and Future works

This study evaluated cloud resource allocation performance using deep learning techniques, specifically in models like CNN, LSTM, and GRU. The results indicated that the models particularly outperformed regarding accuracy, prediction reliability, and resource usage. Using deep learning, the proposed models could thoroughly analyse workload variations, optimise resource allocation, and reduce inefficiencies in virtual cloud infrastructures. The findings confirm the need to integrate various learning architectures towards predictive capability, load balancing augmentation, and delay execution. In addition, the research showed that the performance measures for classification, such as precision, recall, F1-score, and specificity, offered significant information concerning model performance. In general, this research showcases the potential of deep learning in enhancing cloud resource management, leading to more effective scheduling, reduced energy consumption, and enhanced system reliability. With these work directions, researchers can further enhance deep learning- based cloud resource allocation mechanisms to enhance cloud computing in terms of efficiency, adaptability, and affordability.

8. References

[1] R. Bhatia, A. Sharma, and D. Sharma, “Embracing the synergy of cloud computing and business intelligence,” in *Smart Systems: Engineering and Managing Information for Future Success: Navigating the Land- scape of Intelligent Technologies*. Springer, 2025, pp. 191–214.

[2] P. K. Bal, S. K. Mohapatra, T. K. Das, K. Srinivasan, and Y.-C. Hu, “A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques,” *Sensors*, vol. 22, no. 3, p. 1242, 2022.

[3] V. Saravanan, P. Sreelatha, N. R. Atyam, M. Madiajagan, D. Saravanan, H. P. Sultana *et al.*, “Design of deep learning model for radio resource allocation in 5g for massive device,” *Sustainable Energy Technologies and Assessments*, vol. 56, p. 103054, 2023.

[4] X. Wu, L. You, R. Wu, Q. Zhang, and K. Liang, “Management and control of load clusters for ancillary services using internet of electric loads based on cloud–edge–end distributed computing,” *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 267–18 279, 2022.

[5] A. S. Hussein, “An efficient high secure combined framework for cloud computing: An empirical study,” *IAR Journal of Engineering and Technology*, vol. 4, pp. 1–6, 2021.

[6] J.-B. Wang, J. Wang, Y. Wu, J.-Y. Wang, H. Zhu, M. Lin, and J. Wang, “A machine learning framework for resource allocation assisted by cloud computing,” *IEEE Network*, vol. 32, no. 2, pp. 144–151, 2018.

[7] K. Patil and B. Desai, “Intelligent network optimization in cloud environments with generative ai and llms,” 2024.

[8] M. S. Al-Asaly, M. A. Bencherif, A. Alsanad, and M. M. Hassan, “A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment,” *Neural Computing and Applications*, vol. 34, no. 13, pp. 10 211–10 228, 2022.

[9] B. Liu, L. Yu, C. Che, Q. Lin, H. Hu, and X. Zhao, “Integration and performance analysis of artificial intelligence and computer vision based on deep learning algorithms,” *arXiv preprint arXiv:2312.12872*, 2023.

[10] Q. Zhou, L. Wang, and S. Wu, “Resource management optimisation for federated learning-enabled multi-access edge computing in internet of vehicles,” *International Journal of Sensor Networks*, vol. 42, no. 1, pp. 15–28, 2023.

[11] D. Saxena, A. K. Singh, and R. Buyya, “Op-mlb: an online vm prediction-based multi-objective load balancing framework for resource management at cloud data center,” *IEEE Transactions on Cloud Com- puting*, vol. 10, no. 4, pp. 2804–2816, 2021.

[12] S. Badri, D. M. Alghazzawi, S. H. Hasan, F. Alfayez, S. H. Hasan, M. Rahman, and S. Bhatia, “An efficient and secure model using adaptive optimal deep learning for task scheduling in cloud computing,” *Electronics*, vol. 12, no. 6, p. 1441, 2023.

[13] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, “Machine learn- ing for resource management in cellular and iot networks: Potentials, current solutions, and open challenges,” *IEEE communications surveys & tutorials*, vol. 22, no. 2, pp. 1251–1275, 2020.

[14] J. Kim, J. Park, J. Noh, and S. Cho, “Completely distributed power allocation using deep neural network for device to device communication underlaying lte,” *arXiv preprint arXiv:1802.02736*, vol. 2320, 2018.

[15] N. S. Jaber, A. S. Hussein, and H. S. Almalikee, “A new approach to predict the location of petroleum reservoirs using ffnn,” in *2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, 2019, pp. 168–175.

[16] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, “A novel hybrid approach based on rough set for classification: An empirical comparative study.” *Journal of Multiple-Valued Logic & Soft*

Computing, vol. 33, 2019.

[17]X. Song, R. Chai, and Q. Chen, "Joint task offloading, cnn layer scheduling and resource allocation in cooperative computing system," in *Communications and Networking: 14th EAI International Conference, ChinaCom 2019, Shanghai, China, November 29–December 1, 2019,Proceedings, Part I 14*. Springer, 2020, pp. 129–142.

[18]A. Iqbal, M.-L. Tham, and Y. C. Chang, "Convolutional neural network- based deep q-network (cnn-dqn) resource management in cloud radio access network," *China Communications*, vol. 19, no. 10, pp. 129–142, 2022.

[19]S. Swarup, E. M. Shakshuki, and A. Yasar, "Task scheduling in cloud using deep reinforcement learning," Procedia Computer Science, vol. 184, pp. 42–51, 2021.

[20]M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," Cognitive Robotics, vol. 3, pp. 54–70, 2023.

[21]S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, "A review of recurrent neural network architecture for sequence learning: Comparison between lstm and gru," 2021.

[22]F. M. Salem and F. M. Salem, "Gated rnn: the gated recurrent unit (gru) rnn," Recurrent neural networks: from simple to gated architectures, pp. 85–100, 2022.

[23]K. Ong, S.-C. Haw, and K.-W. Ng, "Deep learning based- recommendation system: an overview on models, datasets, evaluation metrics, and future trends," in Proceedings of the 2019 2nd international conference on computational intelligence and intelligent systems, 2019,pp. 6–11.

[24]G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in Computer science on-line conference. Springer, 2023, pp. 15–25.

[25]A. Korotcov, V. Tkachenko, D. P. Russo, and S. Ekins, "Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets," Molecular pharmaceutics, vol. 14, no. 12, pp. 4462–4475, 2017.