Transient Numerical Analysis of Two Phase Repairable Queueing System with Vacation and Standby Server

K. BABY SAROJA, V. SUVITHA*, *Member, IAENG*

***Abstract*—We present an innovative approach to deep learning server infrastructure, emphasizing the importance of continuous service availability and efficiency. The model features a single server managing two distinct service phases: model training and model inference, each with separate queues. In the event of a server failure, a standby server seamlessly takes over, ensuring uninterrupted service. We also incorporate structured maintenance cycles which is mentioned as single vacation state, to optimize system reliability through server maintenance and enhancements. The research mainly focuses on transient numerical results, presenting a solution to the challenge of continuous service availability in the fast-evolving field of deep learning.**

***Index Terms*—Two Phase Queueing Model, Server Failure, Standby Server, Single Vacation, Deep Learning.**

I. Introduction

IN the era of deep learning, the demand for real-time model training and inference services has grown exponentially. The rapid advancement of deep learning has led to increased demands for robust server systems capable of efficiently managing the two primary phases of deep learning: model training and inference. Deep learning has revolutionized various domains, from natural language processing to computer vision. However, the dependence on deep learning models involves a server infrastructure that can effectively handle potential periods of unavailability caused by many sources.

In our proposed work, we have implemented an application utilizing a single server queueing model. For a real-life scenario, we have incorporated features such as server vacation and server breakdown, complemented by a standby server. Our model considers a single server that serves two phases of service with a single vacation. Once the system has become empty and has completed the ongoing service, the server commences a vacation. After the vacation ends, if the system is empty then the server remains idle and ready to serve new incoming customers. Upon returning from vacation, if the server finds that the system is not empty then the server swiftly returns to its regular service rate. This particular type of vacation is commonly referred to as a single vacation.

Manuscript received November 8, 2023; revised April 25, 2024.

K. BABY SAROJA is a Research Scholar in Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur - 603 203, Tamil Nadu, India (e-mail: bk8012@srmist.edu.in).

*V. SUVITHA is an Assistant Professor in Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur - 603 203, Tamil Nadu, India (Corresponding Author to provide e-mail: suvithav@srmist.edu.in).

Ye and Liu [27] studied a single server Markovian arrival process queueing system taking repairs and malfunctions into account. The threshold N-policy for a bulk queueing system with an unstable server and vacations was studied by Sharma [19]. Additionally, Singh et al. [20] used the generating function method to study a queueing model with vacation and to derive a variety of performance indicators. Wang's model was expanded to an N-policy for a two phase queueing system by Wang et al. [26], who concentrated on a single-arrival Erlangian service duration with an unstable server. Numerous authors have conducted in-depth research on queueing models that consider server vacations, as demonstrated by the publications of [6], [9], and [16]. Likewise there are several authors analysed the queueing model with server breakdown and its flexibility in [8], [13], [14] and [15].

Based on the field of queueing model, there is a wealth of literature, including both survey publications and books, that provide well-established practical examples for scenarios involving backup servers during server vacations and queueing models where servers are susceptible to failures and maintenance. Kolledath's survey paper [10] explored into various types of standby models, offering a comprehensive overview of the subject. An examination of a bulk service queueing model with two phases of heterogeneous service, a single primary server, a backup server, and early failures happening during several vacations was carried out by Ayyappan et al. [2]. The complicated dynamics of queueing systems with backup servers and several service phases are investigated in this study. The work of Padma et al. [17] focuses on a queueing model with C server and finite capacity which takes into consideration the server startup processes and considers two phases of operation without gating. The study explores the details of this particular queueing arrangement, offering perceptions on the behaviour and performance of the system.

Kumar and Soodan studied the queueing model with Transient numerical analysis using Runge-Kutta method and provided the numerical solutions [11]. Sudhesh and Vaithiyanathan [22] discussed a queue with single-server, where the arrival process is time-dependent, and they also considered server rates in the context of constant catastrophe rates. In their 1991 study, Zhang and Coyle [28] examined a queueing model without balking and catastrophes. They reported the computational solution of the Volterra integral equation via the Runge-Kutta algorithm and derived the boundary probability function in the form of a second-kind Volterra integral equation [4]. Singh and Gupta [5] obtained the time-dependent and steady-state solutions explicitly, in

their analysis. Jain and Singh [7] focused on the transient model with feedback queue, incorporating elements of disasters and discouragement within the system dynamics, while also considering time-independent parameters. Wagle [25] studied the time-depedent three server queueing model and provided transient solution for finite capacity. Recently, two phase queueing model has been analysed with vacation and server failure by [23] and [29] respectively.

A Q-learning technique was used in [21] to improve upon the maximum dropout probability computation method, leading to decreased sensitivity to parameters and increased overall network performance. Similarly, to enhance network performance, a unique active queuing management technique based on learning through reinforcement was presented in [24]. For active queuing management, a queueing network approach was investigated in [1]. Regarding machine learning, models are trained using a large dataset of system inputs and corresponding outputs, as discussed in [12] and [30]. In the context of queuing network models for software systems, which are often used to predict transaction response times under growing workload demands, these models assist in estimating the additional computing resources needed, as indicated in [3]. In [18], the application of deep learning in the context of two-phase service was discussed concerning private data.

The seamless integration of the training and inference stages in neural networks is essential for the development of artificial intelligence in the rapidly changing arena of deep learning. This complex infrastructure that effectively manages these two crucial phases as well as ensuring continuous service, efficient maintenance, and strong parameter sensitivity. Our work takes a fresh and inventive approach that combines theoretical clarity with practical relevance, as it sets out on a mathematical journey that serves as the foundation for a deep learning server infrastructure.

This paper begins with practical application overview in Section II. In section III, we elaborate on the mathematical model, and provide a comprehensive explanation of the transient equation along with its notations and dedicated to the transient numerical method in section IV. In section V, we examine various system performance measures. Finally the conclusion is presented in VI.

II. Practical Application

In deep learning, training and inference are two distinct phases of neural networks. Training encompasses the process of optimizing a neural network's parameters by exposing it to a dataset, facilitating the model to learn. On the other hand, inference pertains to the utilization of the trained model to generate predictions shown in Figure 1. Let's see how this example aligns with the provided scenario. Consider a practical application in image classification, particularly in determining whether an image contains a car or not. This example underscores the efficiency and real-world relevance of our infrastructure.

In phase 1, known as Model Training, customers submit datasets and training configurations. The server is entrusted with the task of training deep neural networks using the provided data and configurations. This phase is pivotal for model development and optimization. During the Model Training phase, our customers submit extensive datasets containing a diverse range of images, some featuring cars and others devoid of any automotive presence. The server takes on the monumental task of training deep neural networks to recognize and classify these images accurately. Through rigorous optimization and learning processes, the neural networks refine their parameters, ultimately becoming adept at discerning car-related features within images. This phase is critical for the development and optimization of the car detection model.

During phase 2, referred to as Model Inference, customers provide the data they wish to process using the pre-trained models. The server proficiently manages inference tasks, generating predictions or executing specific operations based on the models that were trained earlier. This phase holds particularly importance for real-time applications and decision-making. During the transition to the Model Inference phase, customers submit real-time image data, which may or may not contain cars, seeking prompt classification results. The server, having seamlessly shifted from the training phase, is now fully capable of processing these inference tasks. It leverages the knowledge gained during training to make accurate predictions, swiftly determining whether the submitted images indeed feature cars or not.

Furthermore, our framework includes a standby backup server ready to take over in the event of an unexpected problem with the main server such as hardware failure, software failure, loss of network connectivity, power outages etc. This redundancy assures continuous service availability, which improves the dependability of car image classification. Given the practical implications of detecting cars in photos quickly, the backup server is critical in this scenario. After completing the model training in phase one, the server effortlessly moves to phase two, where it handles inference tasks effectively. It uses trained models to process customer's data, ensuring a smooth transition between phases.

Essentially, during moments when there are no customer requests in either the training or inference queues, be it in phase 1 or phase 2, the primary server or, if applicable, the active backup server, undergoes a designated single vacation period. This interval presents an opportunity for various tasks such as system maintenance, model updates, and optimization of deep learning frameworks. After this period, the server returns to an busy state.

Upon re-entering the busy period, if customers have data to process or require model updates, the server promptly resumes service. Conversely, in the absence of active customer requests, it remains idle, awaiting new tasks to be submitted.

Our deep learning server infrastructure excels in this transition. With the short duration, it shifts from training to inference, showcasing its adaptability and responsiveness. The stability achieved in this transition ensures that car detection results are consistently reliable. This efficiency is a testament to the seamless integration of the two phases and the robustness of our infrastructure.

In this example, the efficiency and reliability of our deep learning server infrastructure shine through queueing

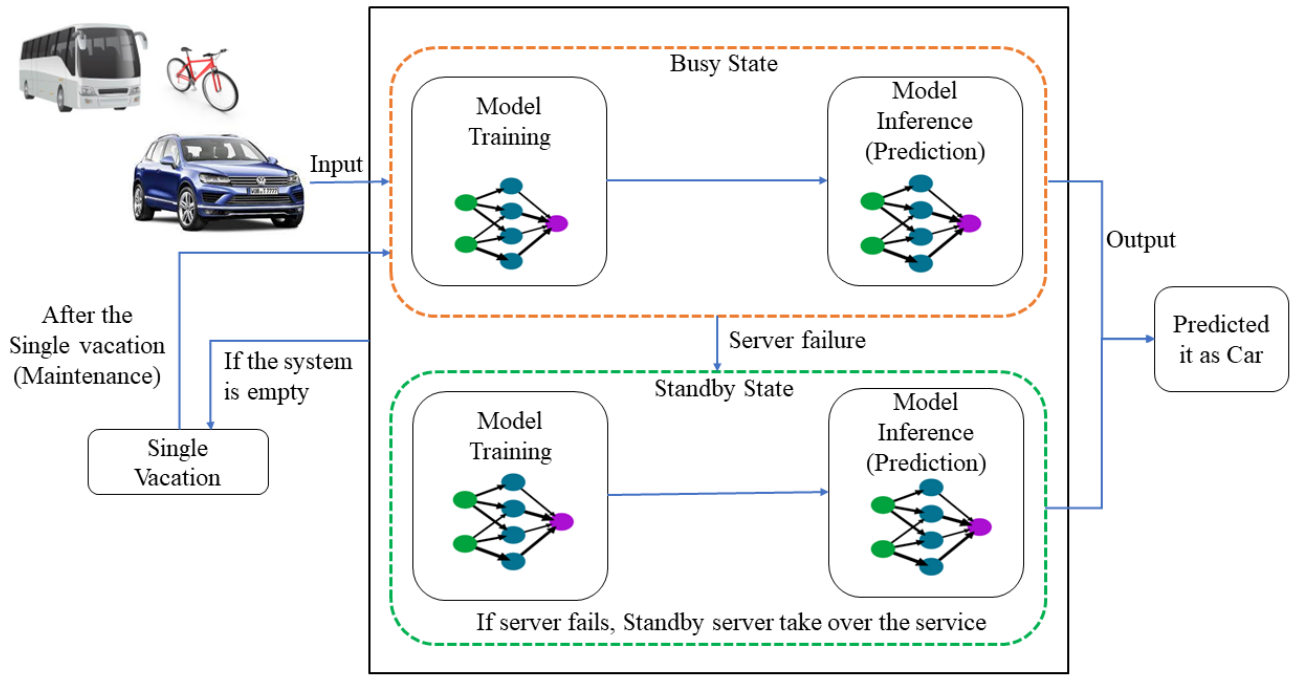

Fig. 1. Pictorial representation of Queueing-based Deep Learning Service Infrastructure

analysis in the below context, demonstrating its ability to seamlessly transition from training to inference and reliably classify images based on the presence of cars. The same principles apply to other deep learning applications, making our infrastructure a valuable asset in the realm of artificial intelligence.

III. Mathematical Model Description

We examine a queuing model for a single server that has two service phases in series with single vacation, standby server, server failure, and repair. The server provide service to each customer separately with limitless capacity in both phase 1 and phase 2.

- Customers arrive at a rate λ according to a Poisson process. Each customer enters and goes through two phases. The customer waits for service for an arbitrary period of time in phase 1, which is exponentially distributed at rate μ_1.
- Upon completing phase 1, the customer proceeds to phase 2 and the service time is exponentially distributed with rate μ_2.
- After completing the service, if the system is empty, then the server goes on a single vacation. At that moment, if any customer arrives, they will queue up in phase 1. The duration of vacation is modelled with an exponential distribution with a rate of ϕ.
- While providing service, the server may experience unexpected failure with a rate of γ. In this case, the standby server instantly takes the control of the service, although at a lower rate of $\theta\mu_1$ for phase 1 and $\theta\mu_2$ for phase 2, where $0 < \theta < 1$. Note that there is no chance of the standby server fail.
- The standby server's service is promptly taken up by the main server after a repair with rate ω is finished. The server failure time, service time of the standby server, repair time are exponentially distributed.

Let $F_x(t)$ denote the number of customers in phase x at time t where $x = 1, 2$. Let $S(t)$ denotes the states of the server at time t. Let's assume the following system states:

$$S(t) = \begin{cases} 0, & \text{the main server is busy} \\ 1, & \text{the main server is under failure and} \\ & \text{the standby server is on progress} \\ 2, & \text{the main server is on vacation} \end{cases}$$

The system can be formulated as $\{F_1(t), F_2(t), S(t) : t \geq 0\}$ with a state space

$$\Omega = \{(i, j, s) : j \geq 0; s = 0, 1\} \cup \{(i, 2)\} \text{ for } i \geq 0$$

IV. Transient Equations and Analysis

We describe the probabilities $P_{i,j,s}(t)$ at time t using the following notations:

- $P_{0,0,0}(t)$ - Probability that the server is idle.
- $P_{i,j,0}(t)$ - Probability that there are i customers in phase 1, j customers in phase 2 and the server is busy.
- $P_{i,j,1}(t)$ - Probability that there are i customers in phase 1 and j customers in phase 2, the server is under failure and standby server takes over the service.
- $P_{i,2}(t)$ - Probability that there are i customers in phase 1, no customer in phase 2 and the server is under single vacation.

The transient equations of the model are

$$P'_{0,0,0}(t) = -(\gamma+\lambda)P_{0,0,0}(t) + \phi P_{0,2}(t) + \omega P_{0,0,1}(t), i,j = 0 \quad (1)$$

$$P'_{i,0,0}(t) = -(\lambda+\mu_1+\gamma)P_{i,0,0}(t) + \mu_2 P_{i,1,0}(t) + \omega P_{i,0,1}(t) + \lambda P_{i-1,0,0}(t) + \phi P_{i,2}(t), i \geq 1; j = 0 \quad (2)$$

$$P'_{0,j,0}(t) = -(\lambda+\mu_2+\gamma)P_{0,j,0}(t) + \mu_1 P_{1,j-1,0}(t) + \mu_2 P_{0,j+1,0}(t) + \omega P_{0,j,1}(t), i = 0; j \geq 1 \quad (3)$$

$$P'_{i,j,0}(t) = -(\lambda+\mu_1+\mu_2+\gamma)P_{i,j,0}(t) + \mu_1 P_{i+1,j-1,0}(t) + \mu_2 P_{i,j+1,0}(t) + \lambda P_{i-1,j,0}(t) + \omega P_{i,j,1}(t), i,j \geq 1 \quad (4)$$

$$P'_{0,0,1}(t) = -(\omega+\lambda)P_{0,0,1}(t) + \gamma P_{0,0,0}(t), i,j = 0 \quad (5)$$

$$P'_{i,0,1}(t) = -(\lambda+\theta\mu_1+\omega)P_{i,0,1}(t) + \theta\mu_2 P_{i,1,1}(t) + \gamma P_{i,0,0}(t) + \lambda P_{i-1,0,1}(t), i \geq 1; j = 0 \quad (6)$$

$$P'_{0,j,1}(t) = -(\lambda+\theta\mu_2+\omega)P_{0,j,1}(t) + \theta\mu_1 P_{1,j-1,1}(t) + \theta\mu_2 P_{0,j+1,1}(t) + \gamma P_{0,j,0}(t), i = 0; j \geq 1 \quad (7)$$

$$P'_{i,j,1}(t) = -(\lambda+\theta\mu_1+\theta\mu_2+\omega)P_{i,j,1}(t) + \theta\mu_1 P_{i+1,j-1,1}(t) + \theta\mu_2 P_{i,j+1,1}(t) + \lambda P_{i-1,j,1}(t) + \gamma P_{i,j,0}(t), \quad i,j \geq 1 \quad (8)$$

$$P'_{0,2}(t) = -(\lambda+\phi)P_{0,2}(t) + \mu_2 P_{0,1,0}(t) + \theta\mu_2 P_{0,1,1}(t), i = j = 0 \quad (9)$$

$$P'_{i,2}(t) = -(\lambda+\phi)P_{i,2}(t) + \lambda P_{i-1,2}(t), i \geq 1; j = 0 \quad (10)$$

with the initial state probabilities given by $P_{0,0,0}(0) = 1$, $P_{i,j,s}(0) = 0 \ \forall \ i = 0, j \geq 1, s = 0,1$, $P_{i,j,s}(0) = 0 \ \forall \ i \geq 1, j = 0, s = 0,1$, $P_{i,j,s}(0) = 0 \ \forall \ i \geq 1, j \geq 1, s = 0,1$ and $P_{i,2}(0) = 0 \ \forall \ i \geq 0$.

Considering the difficulty of explicitly getting an analytical solution for our proposed model, we use the fourth-order Runge-Kutta method to derive the transient solution. The numerical findings for the transient analysis are computed using the *ode45* function in the MATLAB software.

V. Performance Measures

We evaluate the system behaviour based on the following performance measure.

- Mean System Size

$$L_s(t) = \sum_{i=0}^{N}\sum_{j=0}^{N} i(P_{i,j,0}(t) + P_{i,j,1}(t)) + \sum_{i=0}^{N} iP_{i,2}(t)$$

- Probability that the server is busy

$$P_b(t) = \sum_{i=0}^{N}\sum_{j=0}^{N} P_{i,j,0}(t)$$

- Probability that the server is under failure and standby is on progress

$$P_s(t) = \sum_{i=0}^{N}\sum_{j=0}^{N} P_{i,j,1}(t)$$

- Probability that the server is on vacation

$$P_v(t) = \sum_{i=0}^{N} P_{i,2}(t)$$

VI. Cost Analysis

We demonstrate the cost per unit time spent on various operations as follows in order to evaluate the total cost, which includes numerous cost elements per unit time:

- C_h: The holding cost for each customer in the system.
- C_b: The system's operating expenses while it is in busy.
- C_s: The system's operating expenses while it is in failure state and provides service through standby server.
- C_v: The system's operating expenses while it is on vacation.
- C_r: Cost associated with the repairing of the main server.
- C_x: Main server expenses on phase x service for a customer where $x = 1, 2$.
- C_y: Standby server expenses on phase $y-2$ service for a customer where $y = 3, 4$.

So, the Total Cost (TC) at time t is obtained as

$$TC(t) = L_s(t)C_h + P_b(t)C_b + P_s(t)C_s + P_v(t)C_v\omega C_r + \mu_1 C_1 + \mu_2 C_2 + \theta\mu_1 C_3 + \theta\mu_2 C_4$$

VII. Numerical Analysis

In this work, we present a comprehensive analysis of our deep learning server infrastructure through graphical representations. We illustrate the transient behavior of the model using a numerical example with the following parameter values: $\lambda = 1.5$, $\mu_1 = 2$, $\mu_2 = 2.5$, $\theta = 0.4$, $\gamma = 1.5$, $\omega = 2$, $\phi = 3$.

The transient probabilities estimated using the Runge-Kutta method for a specific time $t = 1$ and capacity $N = 6$ for the numerical purpose are shown in Table I. These computed probabilities of the system being in different states at the specified time. By summing the transient probabilities for all states, we find that the total probability is precisely one, thereby confirming the precision and effectiveness of the proposed model. This finding supports the reliability of the Runge-Kutta method and the model's ability to accurately capture the system's time-dependent behavior.

Further numerical work, we have truncate the system with finite capacity for $N = 4$ and for $N = 6$. We have compared the results of mean system size for those capacity levels.

In Figures 2, 3, 4, and 5, we provide plots that depict the probabilities over time. These figures reveal a common trend where the probabilities initially rise before reaching a stable state. Particularly noteworthy is the probability $P_{i,j,s}(t)$, which exhibits a distinct pattern of having its highest value at the outset and subsequently undergoing a gradual decline before stabilizing. This behavior is attributed to the system's initial condition, specifically $P_{0,0,0}(0) = 1$.

Furthermore, in Figures 6, 7, 8, 9, 10, and 11, we illustrate the evolution of mean system sizes over time across a spectrum of parameter values, including λ, μ_1, μ_2, γ, ω, and

TABLE I
PROBABILITES OF $P_b(1)$, $P_s(1)$ AND $P_v(1)$.

j \ i	0	1	2	3	4	5	6	Total
$P_b(1)$								
0	0.1042000000	0.1215340884	0.0918100341	0.0516503017	0.0230746362	0.0085534696	0.0035992692	0.4044217992
1	0.0479985384	0.0400028026	0.0238079295	0.0110066020	0.0041639476	0.0014484350	0.0003506319	0.1287788870
2	0.0112386401	0.0078273882	0.0039116130	0.0015482146	0.0005236764	0.0001433123	0.0000291033	0.0252219479
3	0.0016765464	0.0010076333	0.0004352794	0.0001522711	0.0000440004	0.0000103347	0.0000017714	0.0033278367
4	0.0001727676	0.0000911938	0.0000347859	0.0000106491	0.0000026893	0.0000005483	0.0000000804	0.0003127144
5	0.0000129673	0.0000060907	0.0000020515	0.0000005542	0.0000001231	0.0000000220	0.0000000028	0.0000218116
6	0.0000008537	0.0000003209	0.0000000931	0.0000000221	0.0000000043	0.0000000007	0.0000000001	0.0000012949
$P_s(1)$								
0	0.0630000000	0.0848632671	0.0682819696	0.0396824737	0.0180694032	0.0067715706	0.0028959513	0.2835646355
1	0.0298124691	0.0275996736	0.0171610792	0.0081291266	0.0031216132	0.0010793981	0.0002910259	0.0871943857
2	0.0064223670	0.0049353444	0.0025775316	0.0010463735	0.0003568307	0.0001012142	0.0000227310	0.0154623924
3	0.0008578633	0.0005676529	0.0002567180	0.0000919545	0.0000272358	0.0000066883	0.0000012694	0.0018093822
4	0.0000784549	0.0000455358	0.0000181822	0.0000057408	0.0000014972	0.0000003214	0.0000000523	0.0001497846
5	0.0000052106	0.0000026867	0.0000009507	0.0000002662	0.0000000615	0.0000000116	0.0000000016	0.0000091889
6	0.0000003145	0.0000001273	0.0000000387	0.0000000095	0.0000000019	0.0000000003	0.0000000000	0.0000004922
$P_v(1)$								
0	0.0327335590	0.0117672209	0.0037798555	0.0010829112	0.0002776840	0.0000640926	0.0000133978	0.0497187210
Total								1.0000000000

ϕ. These figures depict how the mean system size evolves from an initial state and gradually converges to a stable value over time.

In Figure 6, we fix the parameters as $\mu_1 = 2$, $\mu_2 = 2.5$, $\theta = 0.4$, $\gamma = 1.5$, $\omega = 2$, $\phi = 3$ and it indicates that as the parameter λ increases, there is a corresponding rise in the mean system size. This happens because when more customers enter the system, the overall system size increases.

For both Figures 7 and 8, we fixed the parameters as $\lambda = 1.5$, $\theta = 0.4$, $\gamma = 1.5$, $\omega = 2$ and $\phi = 3$. From Figure 7, it becomes evident that the mean system size exhibits an upward trend with a decrease in the parameter μ_1 for the fixed value of $\mu_2 = 2.5$. Similarly, in Figure 8, a decrease in the parameter μ_2 is associated with an increase in the mean system size for fixed value of $\mu_1 = 2$.

Furthermore, for Figures 9 and 10, the fixed parameters values are $\lambda = 1.5$, $\mu_1 = 2$, $\mu_2 = 2.5$, $\theta = 0.4$ and $\phi = 3$. In Figure 9, an observed escalation in the mean system size aligns with an increase in the server failure rate γ with $\omega = 2$. This relationship is explained by the system experiencing server failure more frequently, causing delays in processing input data and potentially coinciding with new arrivals, thereby contributing to the overall increase in the mean system size.

In Figure 10, an increase in the repair rate ω is associated with a decrease in the mean system size with $\gamma = 1.5$. This implies that as the repair rate rises, the main server undergoes faster repairs and returns to the busy period more swiftly. Consequently, the server transitions back to regular service rates, leading to a reduction in the mean system size with higher repair rates.

In Figure 11, we fix the parameter values as $\lambda = 1.5$, $\mu_1 = 2$, $\mu_2 = 2.5$, $\theta = 0.4$, $\gamma = 1.5$, $\omega = 2$ and a notable observation is that the mean system size is higher for a vacation rate of $\phi = 1.5$ compared to $\phi = 3.5$. This trend is logical considering that an increasing vacation rate implies shorter duration of server inactivity. Consequently, the main server returns to the busy period more rapidly as the vacation duration decreases, corresponding to an increase in the rate. This, in turn, results in a higher mean system size for a smaller vacation rate.

Figure 12 shows how the service rates in phase 1 and phase 2 affect the mean system size. To assess this, we keep other factors constant at $N = 4$, $\lambda = 1.5$, $\theta = 0.4$, $\gamma = 1.5$, $\omega = 2$, and $\phi = 3$. When both phases have equal high service rates, the system size decreases. Likewise both phases have equal and lower service rates makes the mean system size higher. If the service rate μ_1 in phase 1 is higher than the service rate μ_2 in phase 2 (i.e., $\mu_1 > \mu_2$) , it leads to a larger system size, meaning customers in phase 1 receive service quickly, but overall service is slowed by phase 2.

Conversely, when $\mu_2 > \mu_1$, the system size is smaller, indicating that faster service in phase 2 reduces the overall system size. The system then switches back to phase 1 promptly, serving waiting customers there. This aligns the decrease in mean system size.

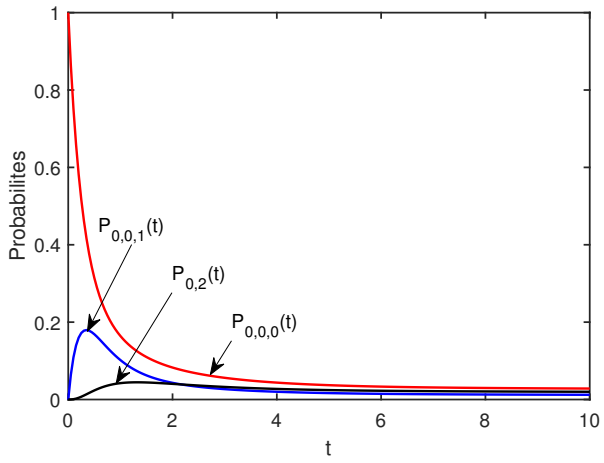

Fig. 2. Time Vs Probabilities $P_{0,0,s}$ and $P_{0,2}$

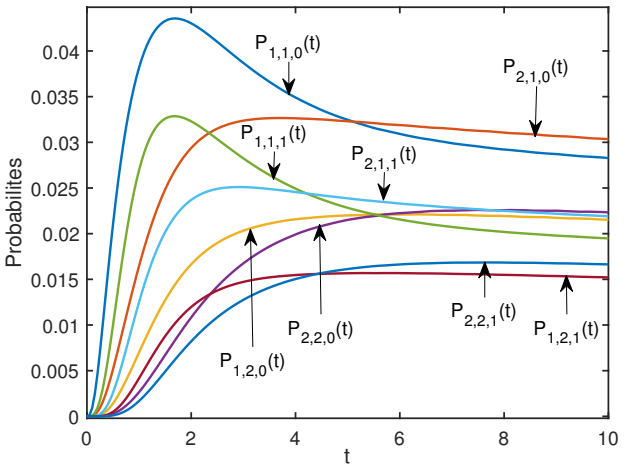

Fig. 5. Time Vs Probabilities $P_{i,j,s}$

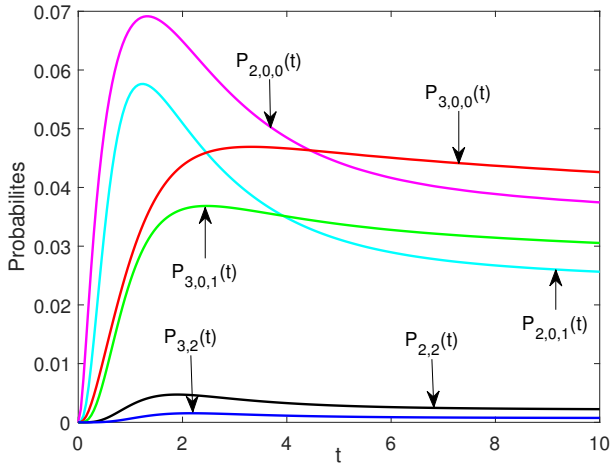

Fig. 3. Time Vs Probabilities $P_{i,0,s}$ and $P_{i,2}$

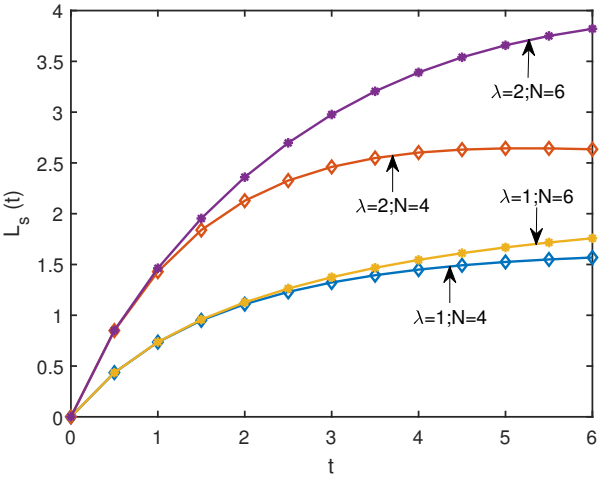

Fig. 6. Time Vs Mean system size for various λ

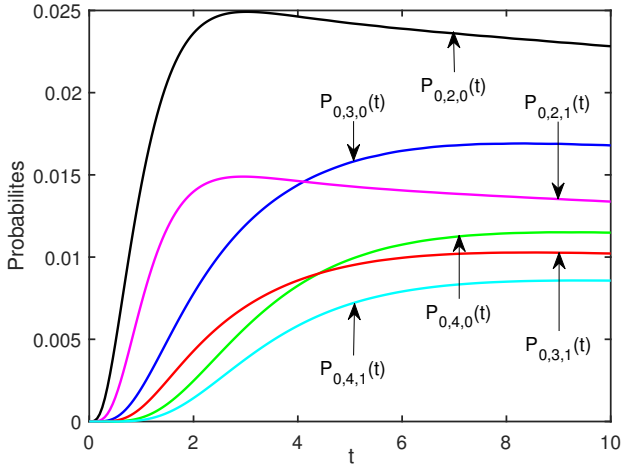

Fig. 4. Time Vs Probabilities $P_{0,j,s}$

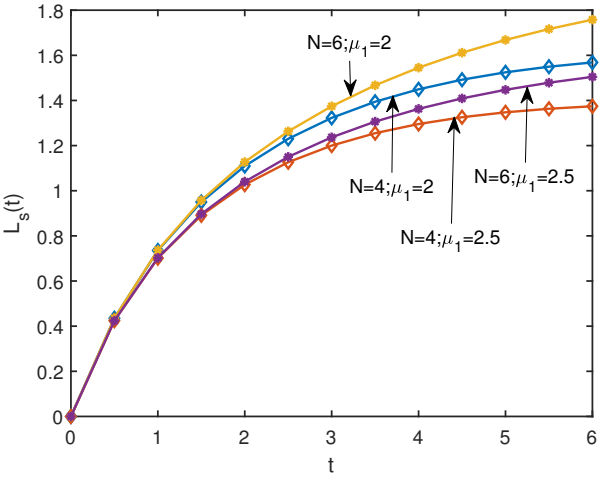

Fig. 7. Time Vs Mean system size for various μ_1

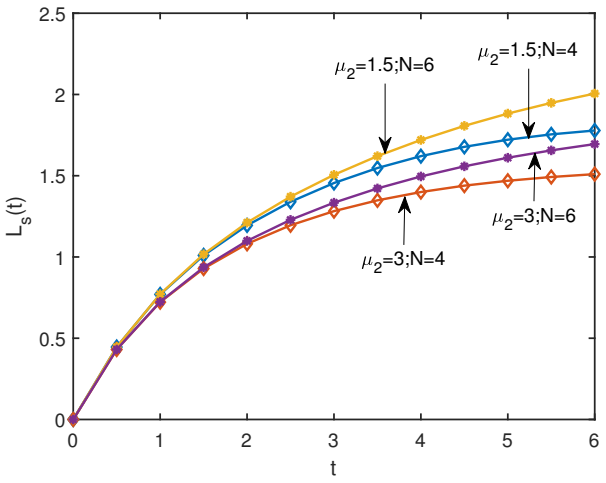

Fig. 8. Time Vs Mean system size for various μ_2

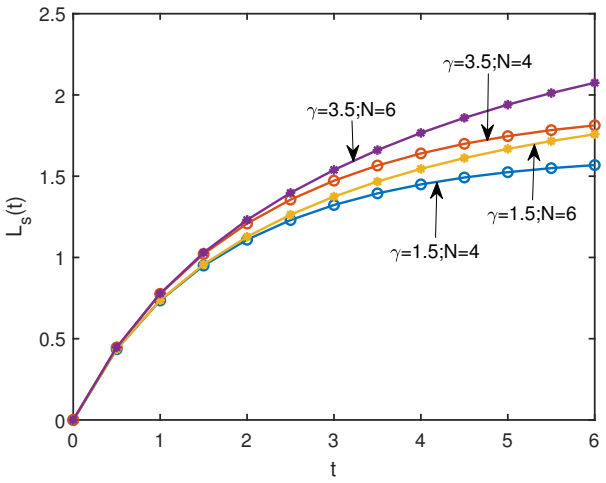

Fig. 9. Time Vs Mean system size for various γ

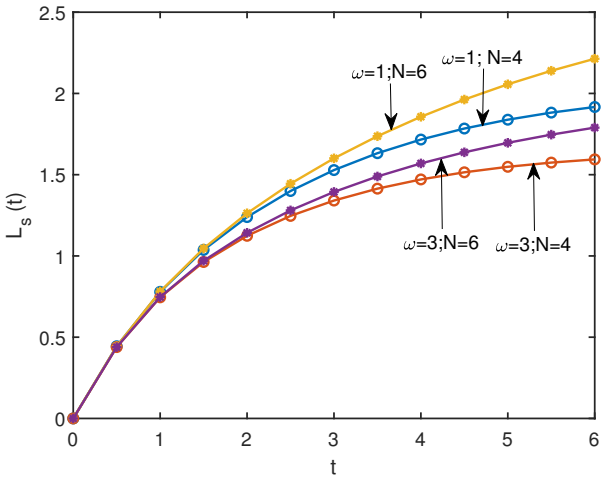

Fig. 10. Time Vs Mean system size for various ω

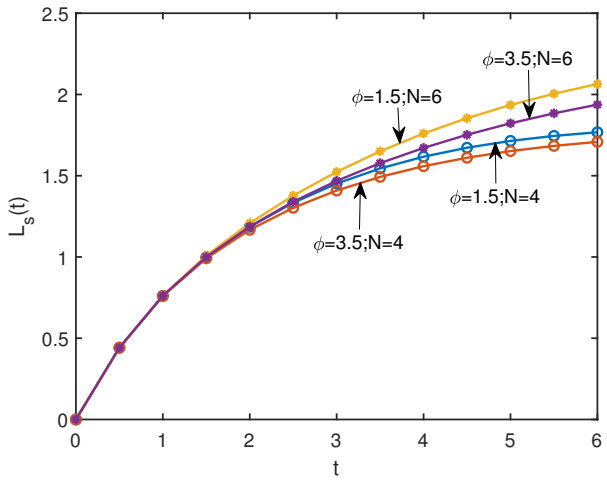

Fig. 11. Time Vs Mean system size for various ϕ

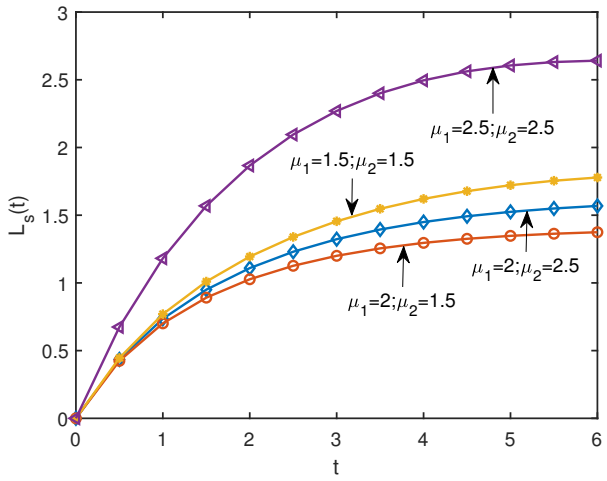

Fig. 12. Time Vs Mean system size for various μ_1 and μ_2

TABLE II
TC FOR VARIOUS VALUES OF λ WITH RESPECT TO TIME t

λ \ t	2	4	6	8	10
1.0	537.7336	554.3275	565.0020	571.0240	574.3321
1.5	558.5157	588.1881	603.8361	611.3473	614.6467
2.0	580.0650	617.2947	632.0076	636.8175	637.4237
2.5	600.6411	638.6240	648.6727	649.6222	651.6636
3.0	618.9546	652.5943	657.4704	658.6470	659.2899

TABLE III
TC FOR VARIOUS VALUES OF γ WITH RESPECT TO TIME t

γ \ t	2	4	6	8	10
1.0	576.8162	612.8578	627.0095	631.6394	632.2256
1.5	580.0650	617.2947	632.0076	636.8175	637.4237
2.0	582.5703	620.6912	635.8002	640.7214	641.3263
2.5	584.5588	623.3720	638.7727	643.7655	644.3595
3.0	586.1743	625.5401	641.1630	646.2030	646.7822

TABLE IV
TC FOR VARIOUS VALUES OF ϕ WITH RESPECT TO TIME t

ϕ \ t	2	4	6	8	10
1.0	580.0650	617.2947	632.0076	636.8175	637.4237
1.5	580.8892	618.3604	634.0126	639.5075	640.5381
2.0	581.3971	618.6559	634.5561	640.3039	641.5106
2.5	581.7278	618.7235	634.6937	640.5547	641.8471
3.0	581.9550	618.7248	634.7127	640.6319	641.9715

In the graphical representations depicted in Figures 6, 7, 8, 9, 10, and 11, a consistent trend emerges: the mean system size exhibits an increase as the system capacity N rises. This pattern aligns with expectations, indicating that as the system's capacity grows, the mean system size also increases across various parameter interpretations. This transient approach, combined with graphical representation, provides a holistic view of deep learning server infrastructure, offering insights into its transient behavior and the influence of parameter variations on the mean system size.

For the cost analysis, we fix the parameters as $\lambda = 2, \mu_1 = 1.5, \mu_2 = 2.5, \theta = 0.6, \gamma = 1.5, \omega = 2, \phi = 3, C_h = 30, C_b = 60, C_s = 80, C_r = 70, C_1 = 40, C_2 = 45, C_3 = 50, C_4 = 55, C_v = 25$. Tables II, III, IV and V display the variation in Total Cost (TC) of the system by varying the λ, γ, ω and ϕ respectively.

In all the Tables II, III, IV and V, we observe that the TC increases as time t grows. In Tables II, III and IV, for increasing values of λ, γ and ϕ, TC is also increases, respectively. However, Table V presents an intriguing observation. Here, we find that as the repair rate ω increases, the TC decreases. This phenomenon is likely due to the fact that a higher repair rate ensures quicker system availability, enabling customers to access service from the main server sooner after repairs. Thus, the TC increases over time but decreases while the repair rate ω increases.

VIII. CONCLUSION

We constructed a queueing model in this research that effectively handles deep learning infrastructure, as illustrated by the practical case of picture classification (i.e., determining if an image has an automobile or not). Through the implementation of a two-phase service model that includes single vacation, server failure and standby, we ensure that customers may still receive deep learning services even in the event of unexpected disruptions. The transient numerical solutions

TABLE V
TC FOR VARIOUS VALUES OF ω WITH RESPECT TO TIME t

ω \ t	2	4	6	8	10
1.0	585.2638	624.9033	640.6849	645.8138	646.4433
2.0	580.0650	617.2947	632.0076	636.8175	637.4237
3.0	577.0524	613.0328	627.1529	631.7678	632.3461
4.0	575.1067	610.3160	624.0555	628.5376	629.0910
5.0	573.7515	608.4357	621.9097	626.2951	626.8271

highlight the effectiveness of the system's resource allocation, allowing for precise and reliable image classification. Our fresh approach combines mathematical precision with real-world applications, ensuring consistent performance and reliable decision-making, ultimately advancing the field of artificial intelligence. In future, we can extend the model with Markovian Arrival Process with phase type service using Matrix Geometric Method.

REFERENCES

[1] D. A. AlWahab, G. Gombos and S. Laki, "On a Deep Q-Network-based approach for active queue management", *Joint European Conference on Networks and Communications and 6G Summit, IEEE*, pp. 371-376, 2021.

[2] G. Ayyappan, M. Nirmala and S. Karpagam, "Analysis of repairable single server bulk queue with standby server, two phase heterogeneous service, starting failure and multiple vacation ", *International Journal of Applied and Computational Mathematics*, vol. 6, no. 52, pp. 1-22, 2020.

[3] R. O. Baldwin, N. J. Davis Iv, S. F. Midkiff and J. E. Kobza, "Queueing network analysis: concepts, terminology, and methods", *Journal of systems and software*, vol. 66 , no. 2, pp. 99-117, 2003.

[4] H. Brunner, "On the history of numerical methods for Volterra integral equations", *CWI Newslett*, vol. 11, no. 3, pp. 1-18, 1986.

[5] G. S. Bura and S. Gupta, "Time Dependent Analysis of an $M/M/2/N$ Queue With Catastrophes", *Reliability: Theory and Application*s, vol. 14, no. 1, 79-86, 2019.

[6] B. T. Doshi, "Queueing systems with vacations - a survey", *Queueing systems*, vol. 1, pp. 29-66, 1986.

[7] M. Jain and M. Singh, "Transient analysis of a Markov queueing model with feedback, discouragement and disaster", *International Journal of Applied and Computational Mathematics*, vol. 6, no. 31, pp. 1-14, 2020.

[8] K. Kalidass, J. Gnanaraj, S. Gopinath and K. Ramanath, "Transient analysis of an $M/M/1$ queue with a repairable server and multiple vacation", *International Journal of Mathematics in Operational Research*, vol. 4, no. 2, pp. 193-216, 2014.

[9] W. M. Kempa, "Transient workload distribution in the $M/G/1$ finite-buffer queue with single and multiple vacations", *Annals of Operations Research*, vol. 239, no.2, pp. 381-400, 2016.

[10] S. Kolledath, "Survey on queueing models with standbys support", *Yugoslav Journal of Operations Research*, vol. 28, no.1, pp. 3-20, 2017.

[11] R. Kumar and B. S. Soodan, "Transient numerical analysis of a queueing model with correlated reneging, balking and feedback", *Reliability: Theory & Applications*, vol. 14, no.4, pp. 46-54, 2019.

[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F. E. Alsaadi, "A survey of deep neural network architectures and their applications", *Neurocomputing*, vol. 234, pp. 11-26, 2017.

[13] L.V. Shengli, W. U. Yueying, L. I. Jingbo and R. Wang, "The $M/M/1$ Repairable Queueing System with Variable Input Rates and Failure Rates ", *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 784-790, 2022.

[14] L.V. Shengli, "Two repairmen machine repairable system with flexible repair policy", *IAENG International Journal of Applied Mathematics*, vol. 50, no. 2, pp. 336-341, 2020.

[15] L. I. Jing and L. I. Tao, "An $M/G/1$ $G-$queue with Server Breakdown, Working Vacations and Bernoulli Vacation Interruption", *IAENG International Journal of Applied Mathematics*, vol. 50, no. 2, pp. 421-428, 2020.

[16] T. Naishuo and G. Z. Zhe, "Vacation Queueing Models Theory and Applications", *International Series in Operations Research and Management Science*, vol. 93, 2006.
[17] Y. Padma, A. R. Reddy and L. V. Rao, "Matrix geometric approach for $M/M/C/N$ queue with two phase service", *International Journal of Engineering Science and Advance Technology*, vol. 2, no.2, 166-175, 2012.
[18] M. S. Riazi, B. D. Rouani and F. Koushanfar, "Deep learning on private data", *IEEE Security and Privacy*, vol. 17, no.6, pp. 54-63, 2019.
[19] R. Sharma, "Threshold N-policy for $M^X/H_2/1$ queueing system with un-reliable server and vacations", *Journal of International Academy of Physical Sciences*, vol. 14, no. 1, pp. 41-51, 2010.
[20] C. J. Singh, M. Jain and B. Kumar, "Analysis of $M/G/1$ queueing model with state dependent arrival and vacation", *Journal of Industrial Engineering International*, vol. 8, pp. 1-8, 2012.
[21] Y. Su, L. Huang and C. Feng, "QRED: A Q-learning-based active queue management scheme", *Journal of Internet Technology*, vol. 19, no. 4, pp. 1169-1178, 2018.
[22] R. Sudhesh and A. Vaithiyanathan, "Time-dependent single server Markovian queue with catastrophe", *Appl. Math. Sci*, vol. 9, no. 66, pp. 3275-3283, 2015.
[23] R. Tian, Y. Song, Y. Xue and Y. Hao. "Equilibrium Analysis of the $M/M/1$ Unreliable Queue with Two-phase Vacations and Vacation Interruption", *IAENG International Journal of Applied Mathematics*, vol.54, no. 1, pp. 10-19, 2024.
[24] N. Vucevic, J. Pérez-Romero, O. Sallent and R. Agustí," Reinforcement learning for active queue management in mobile all-ip networks", *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, pp. 1-5, 2007.
[25] B. R. Wagle and R. P. Ghimire, "Performance analysis of load based $M/M/3$ transient queueing system with finite capacity", *TWMS Journal Of Applied And Engineering Mathematics*, vol.14, no.1, pp. 382-394, 2024
[26] K. H. Wang, K. W. Chang and B. D. Sivazlian, " Optimal control of a removable and non-reliable server in an infinite and a finite $M/H_2/1$ queueing system", *Applied Mathematical Modelling*, vol. 23, no.8, pp. 651-666, 1999.
[27] Q. Ye and L. Liu, "Analysis of $MAP/M/1$ queue with working breakdowns", *Communications in Statistics-Theory and Methods*, vol. 47, no.13, pp. 3073-3084, 2018.
[28] J. I. Zhang and E. J. Coyle, "The transient solution of time-dependent $M/M/1$ queues", *IEEE Transactions on Information Theory*, vol. 37, no. 6, pp. 1690-1696, 1991.
[29] Y. Zhang, Z. Ma, Y.Liu and S. Wang. "Analysis of P2P network system based on two-stage service and failure repairability", *Concurrency and Computation: Practice and Experience*, vol.35, no.23, pp. e7765, 2023.
[30] Z. Zhang, "Improved adam optimizer for deep neural networks", *IEEE/ACM 26th International Symposium on Quality of Service*, pp. 1-2, 2018.

K. Baby Saroja is pursuing her Ph.D. in Mathematics as a full-time research scholar at SRM Institute of Science and Technology, Kattankulathur, Chennai. She received her Bachelor's degree in Mathematics from Bharathidasan University in 2019 and her Master's degree in Mathematics from the University of Madras in 2021.

V. Suvitha is working as an Assistant Professor in the Department of Mathematics at SRM Institute of Science and Technology, Kattankulathur, Chennai. Her areas of interest are Stochastic Processes and Queueing Theory. She obtained her PhD degree in 2016. She has 7+ years of teaching experience. She has published more than 10 research papers in reputed national and international Journals.