mathematics

MDPI

Article

Michaelis–Menten-Based Supervised Learning for Nonlinear Performance Modelling in Bulk-Service Queueing System

Pooja Gokul Joshi [1], Gopal Kumar Gupta [1,*], Maria Luminita Scutaru [2] and Sorin Vlase [2]

[1] Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune 440008, India; pooja.joshi.phd2024@sitpune.edu.in
[2] Department of Mechanical Engineering, Transilvania University of Brasov, 500036 Brasov, Romania; lscutaru@unitbv.ro (M.L.S.); svlase@unitbv.ro (S.V.)
* Correspondence: gopalgupta.iitbhu90@gmail.com or gopalkumar.gupta@sitnagpur.siu.edu.in

Abstract

Bulk service queueing systems such as $M/M^b/1/N$ and $M/M^b/c/N$, which are characterised by multiple servers c, fixed batch-size service b, and finite capacity N, indicate the performance measures that are numerically derivable but mathematically complex to evaluate across various parameter ranges. This study addresses that challenge by applying the Michaelis–Menten (M-M) regression model, originally formulated for enzyme kinetics, as an empirical substitute for these measures. The steady-state probabilities are obtained by forming the generator matrix Q, which is solved via LU decomposition process. This yields the exact expressions for average queue length L_q, average system length L, average waiting time W_q, effective arrival rate λ_{eff}, blocking probability p_{block}, and server utilisation. Identifying the inherently saturating, nonlinear relationship between these performance measures and the traffic intensity ρ, the M-M kinetics regression model, originally evolved for enzyme reaction rates, is utilised as an empirical substitute for capturing saturating behaviour. Levenberg–Marquardt, a non-linear least-squares regression algorithm, is used for estimating model parameters V_{max} and K_{max}. The fitted M-M model signifies excellent agreement with the exact solutions solved by using LU decomposition across a wide range of system configurations, with the coefficient of determination R^2 reaching as high as 99.5%. Beyond accuracy, the regression-based approach offers substantial computational efficiency over complex numerical evaluation, making it attractive for rapid system design and optimisation. Results establish Michaelis–Menten regression as a feasible non-linear regression approach for bulk queueing analysis, spanning classical queueing theory and modelling approaches. It offers practical applications for improving decision-making in the analysis of congestion, capacity optimisation, and dynamic resource allocation, which arise frequently in transportation networks, cloud computing job schedulers, and manufacturing assembly lines.

Keywords: bulk-service queueing models: $M/M^b/1/N$; $M/M^b/c/N$; non-linear regression; Michaelis–Menten model; cost function; simulation

MSC: 60K25; 68M20; 90B22; 60J27; 60K30

Check for updates

Academic Editor: Pouya Faroughi

Received: 16 September 2026
Revised: 15 October 2026
Accepted: 19 October 2026
Published: 22 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Whether it is a call centre assisting customers or a server executing computing tasks, most service systems have limited capacity and face sudden spikes in demand. Queueing theory offers a structured method to understand these systems by examining how jobs

wait, are served, and exit the system. Queueing theory serves as a foundational tool for understanding how finite resource systems manage random flows of service, customers, or tasks; it enables quantitative analysis of key performance metrics such as delay times, system utilization, and congestion. Among various queueing models, we will discuss the bulk-service queueing model, where the system services multiple tasks concurrently, unlike the single-task processing model. This is valuable where group processing is either more efficient or economically more beneficial. Real-world examples include toll plazas processing multiple vehicles in a lane, a data communication system that transmits information in packets, and computing platforms that execute tasks in batches to reduce latency and overhead. The key strength of this model lies in its potential to capitalise on economies of scale, costs, and enhance throughput, hence, making such models an ideal solution for high-volume, efficiency-driven service systems.

Recent developments in mathematical modelling include machine learning algorithms to access and simulate complex waiting conditions. Many investigations have shown that machine learning improves queueing theory, particularly for systems with limited buffers and changing behaviours [1]. These approaches allow data-driven predictions and operational measure learning, so they can be scaled up or down and utilised with varied queue setups. Machine learning techniques, including supervised and reinforcement learning, have been used to analyse performance and optimisation in $GI/G/K$ multi-server queueing systems where exact analytical solutions are computationally difficult to obtain [1]. In queueing scenarios, Michaelis–Menten kinetics give a straightforward and elegant mathematical explanation for how load level affects service rates [2]. Real-world applications have used machine learning and queueing theory. A supervised machine learning method utilises the M-M model, which provides an equation to describe the rate of enzymatic reactions historically used in biochemistry for estimating queue performance measures more accurately. The M-M model is particularly effective in reflecting how queue metrics saturate as systems reach their capacity limits [3]. We study the $M/M^b/c/N$ queueing model, a bulk-service, multi-server, finite-capacity system in which service occurs in batches of size b across c parallel servers, with system capacity bounded by N. The structural complexity of such models, arising from the batch-service mechanism and the interaction between multiple servers under finite capacity, results in generator matrices of higher dimension than their single-server counterparts. The study of bulk-service and multi-server queueing systems has a well-established analytical foundation. Early investigation into queues involving bulk service began with a study establishing foundational results on batch-processing behaviour [4]. This body of work was later brought together into a comprehensive treatment covering both bulk-arrival and bulk-service variants, along with their steady-state characterisation [5]. Multi-server extensions of these models, incorporating finite waiting capacity, have since been studied to reflect realistic constraints on system size, with performance measures derived through generator-matrix approaches [6,7].

Building on this foundation, recent research has begun to explore the integration of machine learning techniques specifically within bulk-service and multi-server queueing contexts. Approaches in this direction include regression-based modelling and curve-fitting of performance measures against system parameters. The present work focuses on a Michaelis–Menten regression framework for approximating performance measures of the $M/M^b/c/N$ queueing model, combining the analytical rigour of matrix-based steady-state solution with the computational efficiency of regression approximation. Its ability to represent performance patterns with clarity makes it a promising tool for real-time queue optimisation and performance monitoring.

Motivated by the call-centre and computing-server settings mentioned at the start of this section, where jobs are handled in groups under limited capacity, we develop a supervised

machine learning algorithm (Algorithm 1, Section 5) for the analysis and simulation of such waiting conditions. The algorithm learns the relationship between traffic intensity and congestion from data generated by the queueing model and then predicts the mean number of customers at unseen traffic intensities without solving the generator matrix again.

Algorithm 1 Supervised learning algorithm for the analysis of the $M/M^b/c/N$ queue

Require: b, c, N, μ; training grid $\rho_1, \ldots, \rho_{25} \in [0.1, 5.0]$
Ensure: Fitted parameters (V_{max}, K_{max}) and predictor $\hat{L}(\rho)$

Validation points: $\rho_i^* \leftarrow (\rho_i + \rho_{i+1})/2, \quad i = 1, \ldots, 24$

Data generation
for $i = 1$ **to** 25 **do**
    $\lambda_i \leftarrow \rho_i \, b \, c \, \mu$
    Build the generator matrix Q of order $N + 1$
    Solve $\boldsymbol{\pi}\tilde{Q} = \boldsymbol{e}_{N+1}$ by LU decomposition
    Compute L_i from $\boldsymbol{\pi}$
end for
Training set $\mathcal{D} \leftarrow \{(\rho_i, L_i)\}_{i=1}^{25}$ (The discrete-event simulation of Section 7 can supply the queue data.)

Michealis–Menten Model selection
Choose

$$\text{first-order: } \hat{L}(\rho) = \frac{V_{max}\,\rho}{K_{max} + \rho} \quad \text{or} \quad \text{second-order: } \hat{L}(\rho) = \frac{V_{max}\,\rho^2}{K_{max} + \rho^2}$$

Training
$V_{max}^{(0)} \leftarrow 2 \max_i L_i, \quad K_{max}^{(0)} \leftarrow 1$
$(V_{max}, K_{max}) \leftarrow \arg\min SS_{res} = \sum_{i=1}^{25} \left(L_i - \hat{L}(\rho_i)\right)^2$ by non-linear least squares, subject to $V_{max}, K_{max} \geq 0$

Acceptance
if optimiser terminated successfully **and** V_{max}, K_{max} lie strictly inside the non-negativity bounds **then**
    accept the fit
else
    reject the fit
end if

Evaluation
Compute R^2, RMSE and MAE on the training grid $\{\rho_i\}_{i=1}^{25}$
Compute RMSE and MAE at $\{\rho_i^*\}_{i=1}^{24}$ **without** re-estimating (V_{max}, K_{max})

Prediction
function PREDICT(ρ)
    return $\hat{L}(\rho)$ ▷ no matrix is built, no linear system is solved
end function

Notations and Acronyms Used

Table 1 lists the acronyms, symbols and indices used throughout this paper.

Table 1. Acronyms, symbols and indices.

Symbols/Acronym	Definitions
M-M	Michaelis–Menten
ML	Machine learning
LU	Lower–upper triangular decomposition

Table 1. *Cont.*

Symbols/Acronym	Definitions
NLS	Non-linear least squares
CTMC	Continuous-time Markov chain
CI	Confidence interval (95%)
RMSE, MAE	Root mean square error, mean absolute error (in customers)
PSO, LSTM	Particle swarm optimisation; long short-term memory network
ID3	Iterative Dichotomiser 3, an inductive learning algorithm
ER	Emergency room
$M/M^b/1/N$	Single-server bulk-service queueing model with finite capacity N and fixed batch size b
$M/M^b/c/N$	Multi-server bulk-service queueing model with finite capacity N and fixed batch size b
$GI/M^b/c/N$	Renewal arrivals, exponential batch service, c servers, finite capacity
$GI/G/K$	General arrivals and service with K servers
E, S, P	Enzyme, substrate, product
k_1	Binding rate for $E + S \rightarrow ES$
k_{-1}	Dissociation rate for $ES \rightarrow E + S$
k_2	Product formation rate for $ES \rightarrow E + P$
v	Reaction or service rate
S	Substrate concentration or system load or traffic
V_{max}	Maximum possible rate or saturation point
K_{max}	Michaelis constant, i.e., value of [S] when $v = V_{max}/2$)
λ	Arrival rate
μ	Service rate
ρ	Traffic intensity
b	Fixed batch size
c	Multiple servers
N	Finite capacity
N_q	Maximum number of customers waiting in the queue
Q	Infinitesimal generator matrix, of order $N + 1$
$\tilde{\mathbf{Q}}$	$\mathbf{Q}$ with its last column replaced by $\mathbf{1}$
B_r, A_r, C_r	Blocks of $\mathbf{Q}$- transitions within level r, from level r to $r + 1$ and from level r to $r - 1$
I_b	Identity matrix of order b
λ_{eff}	Effective arrival rate
ρ'	Utilisation factor of the service station
n	total number of customers in the system, number of training points
$p_n(t)$	Probability of n customers in the system (steady state and at time t)
j	Number of customers waiting in the queue, $j = 0, \ldots, b - 1$ or $0, \ldots, N_q$
r	Number of busy servers, $r = 0, \ldots, c$
$X_q(t), X_s(t)$	Number waiting in the queue and number of busy servers at time t
$X(t)$	Total number in the system at time t, $X(t) = X_q(t) + bX_s(t)$
(j, r)	State of the CTMC
$p_{j,r}(t), p_{j,r}$	Probability of state (j, r) at time t and in steady state
i	Index of a traffic-intensity point, $i = 1, \ldots, m$
L	Mean number of customers in the system
L_q	Mean number of customers in the queue
L_s	Mean number of customers in the service

Table 1. *Cont.*

Symbols/Acronym	Definitions
p_{block}	Blocking probability, $p_{N_q,c}$
W	Mean waiting time in the system
W_q	Mean waiting time in the queue
p^{idle}	Probability that all servers are idle
p^{busy}	Probability that at least one server is busy
L^{busy}	Average number of customers in the system given r servers are busy
L_q^{busy}	Average number of customers in the queue given r servers are busy
L_{sr}^{busy}	Mean number in service given that at least one and at most r servers are busy $(r = 1, \ldots, c)$
$L_{\text{s1}}^{\text{busy}}$	Average number of customers in service given 1 server is busy
ρ_i^*, m_v	Validation points (midpoints of the fitting grid) and their number, $m_v = 24$
$\hat{L}(\rho)$	Predicted mean number in the system
SS_{res}, SS_{tot}	Residual and total sums of squares
$\bar{L}$	Mean of the exact values L_i
T	Simulation horizon after warm-up
W	Warm-up length (Section 7)
$\hat{L}_R$, $\hat{p}_R$	Time-averaged number in the system and blocking proportion in replication R for Simulation
R^2	Coefficient of determination
T(C)	Total cost rate
$C_{s0}c$	Initial cost per server per unit time, independent of server's batch capacity
$C_v bc$	Cost of provisioning c servers each of capacity b
$C_h L_q$	Holding cost for customers waiting in the queue
$C_{s1} L_s$	Service cost while customers are actually serving
$C_d \frac{\lambda_{eff}}{b}$	Fixed cost incurred every time a batch is dispatched
$C_l \lambda p_{block}$	Penalty incurred for each customer, using the blocking probability

2. Background and Literature Review

Unlike traditional queueing models that handle a single customer at a time, bulk queueing system involves the arrival or service of multiple customers simultaneously. These systems better reflect real-world scenarios in domains such as telecommunications, manufacturing, transportation, and parallel computing, where resources are often processed in groups or batches.

Over the decades, various researchers have proposed numerical and analytical techniques to better understand and solve complex queueing models. Foundational work in this area focused on discrete-time analysis of the single-server queue, establishing key metrics such as waiting time, queue length, and residual service time. Further developments in numerical techniques have also contributed significantly to queueing analysis [8,9]. Bulk-service queueing systems by Chaudhry and Templeton [5] systematically develop the analytical foundation for a broad class of bulk-arrival and bulk-service models with single- and multi-server variants. Alfa [10] provided a matrix-analytic treatment of discrete-time queues, offering algorithmic techniques for computing steady-state and transient behaviour in various queueing systems. The study of bulk-service queues surged through analytical and computational advancements that can be seen in [11–13]. Goswami et al. [14] examined a finite-buffer, multi-server bulk-service queue in which customers arrive according

to a general renewal process while service times follow an exponential distribution, corresponding to the $GI/M^b/c/N$ model. The growing complexity of system configurations was computationally addressed by Yu and Alfa [15], who developed an algorithm for computing queue length distributions in discrete-time bulk service systems with batch-size dependent service using matrix-analytical methods.

Parallel to the mathematical advancements of queueing theory, new possibilities have opened due to the emergence of machine learning for real-time adaptation and prediction of performance measures. One of the earliest instances of applying machine learning to stochastic modelling involved the use of the Iterative Dichotomiser 3 (ID3), an inductive learning algorithm within a queue simulation, demonstrating early synergy between machine learning and queueing systems [16]. A controller-based model leveraging machine learning for scheduling of a multi-queue in network communication system was proposed by Chen and Yen [17]. In transport and infrastructure, Wang et al. [18] proposed a model using a hybrid PSO-LSTM predictive framework integrating with queueing theory to optimise toll stations lane work scheduling. Tan and Teoh [19] applied LSTM neural networks to optimise sanitary fixture allocation in office buildings, achieving a 39.26% improvement in prediction accuracy over conventional $M/M/c$ queueing models. Moreover, Tan et al. [20] addressed the problem of cycle-based queue length estimation for enhancing traffic signal management.

In service overload predictions, the authors beat rule-based algorithms. Edge computing queue-aware resource ordering may reduce multi-server latency. This suggests that model training queueing awareness improves cloud platform performance. These improvements show that domain-specific knowledge and data-driven learning improve service systems. ER congestion has been examined using queueing models. Machine-learning-based queueing models may detect service difficulties and plan capacity [21]. Anussornnitisarn and Limlawan [22] designed an advanced queue-management system that used an artificial neural network to estimate customer waiting times, illustrating the broader applicability of neural-network-based prediction beyond any single service domain. Hijry and Olawoyin [23] employed deep learning algorithms to forecast patient waiting times within an emergency room queueing system, reporting improved prediction accuracy over traditional statistical approaches when handling the highly stochastic nature of emergency arrivals. Learning-enhanced delay forecasting outperforms classical techniques for multiclass services [24]. Li et al. [25] developed a machine learning framework to predict outpatient waiting times in a tertiary children's hospital, demonstrating that data-driven models can capture the nonlinear relationships between patient flow variables and waiting time more effectively than conventional estimation methods. Extending machine learning applications to more complex, multi-stage healthcare queues, Benevento, Aloini, and Squicciarini [26] undertook a comparative evaluation of several machine learning techniques for real-time waiting-time prediction in emergency departments to improve prediction accuracy. Complementing this emergency department focused work, Al-Mousa, Al-Zubaidi, and Al-Dweik [27] proposed a machine-learning-based framework for estimating patient wait times in healthcare facilities characterised by multi-stage queueing structures, where patients pass sequentially through several service stages rather than a single queue. Wu et al. [28] proposed a neural network training framework specifically tailored to the $GI/G/K$ queue, showing that neural architectures can be trained to approximate performance measures of general multi-server queueing systems where closed-form analytical solutions are difficult to obtain.

In comparison to the existing literature given in Table 2, the original contribution of this study offers key advantages in analytical rigour, system generalisability, and real-time adaptability. Whereas existing empirical machine learning approaches [18,19,22–27] fit

models with a large number of learned parameters, such as network weights, gate matrices, and tuning hyperparameters, none of which corresponds to a measurable property of the queueing system. Such models do not inherently respect structural constraints and their behaviour outside the training range is unconstrained. In contrast, the proposed Michaelis–Menten form uses only two or three parameters, each of which has a direct queueing interpretation as a saturation level or a half-saturation traffic intensity. Furthermore, while conventional analytical models utilised in [3,28] rely on restrictive exponential distributions or encounter heavy computational overhead, the proposed method seamlessly extends to generalised multi-server and multi-stage arrival and service distributions. Finally, unlike domain-specific architectures [20,21] that lack transferability beyond their targeted environments, this work provides a versatile structure capable of dynamically adapting to real-time workload shifts and variable buffer sizes with low operational latency.

Table 2. Comparative overview of related works.

Code	Applied Method/Model	Basic Goal of the Results
[1]	ML (Logistic Regression, Decision Tree, Random Forest, Neural Network) combined with analytical GI/G/K multi-server queueing models	Applications of ML algorithms in estimating performance metrics for the GI/G/K queue
[3]	Supervised learning with the Michaelis–Menten non-linear regression model	Approximating the mean number in system for finite-buffer queues
[18]	Queueing theory with PSO-LSTM	Optimisation of lane work-scheduling at toll stations to minimise congestion
[19]	LSTM neural network	Optimisation of sanitary facility allocation and usage in office buildings
[20]	Cycle-based estimation from sparse vehicle trajectory data	Queue length estimation at signalised intersections
[21]	Queueing optimisation and resource scheduling	Minimisation of latency and delay in multi-server cellular edge computing nodes
[22]	Artificial neural network	Waiting time prediction in advanced service queueing systems
[23]	Deep learning algorithms	Patient waiting time prediction within emergency room queue systems
[25]	Machine learning classifiers/regressors	Outpatient waiting time prediction in a tertiary children's hospital
[26]	Comparison of ML techniques	Real-time prediction and comparative forecasting of waiting times in emergency departments
[27]	ML-based estimation	Estimation of waiting time in multi-stage healthcare facilities with complex queue structures
[28]	Neural network and GI/G/K queue	Approximation of performance measures of a general multi-server queue

3. Michaelis–Menten(M-M) Model in Machine Learning

The M-M Model used to describe the kinetics of enzyme-catalysed reactions in biochemistry. It is used to understand how enzymes interact with a substrate to form a product. It is now employed in numerous domains, notably in performance models of systems that queue complex-behaving humans. In queueing theory, the M-M equation helps us comprehend and access saturation effects, which occur when service rates cease rising as queue lengths increase. The M-M equation is a non-linear model originally developed for

enzymatic reaction rates, but it is useful in modelling queueing systems where saturation effects occur. The M-M equation that generally relates the initial reaction velocity to the substrate concentration is given as [3]

$$v = \frac{V_{max} \times [S]}{K_{max} + [S]} \tag{1}$$

where v = Reaction or service rate, S = substrate concentration (analogous to system load or traffic), V_{max} = maximum possible rate (saturation point), K_{max} = Michaelis constant (value of [S] when $v = V_{max}/2$). This M-M equation tells us how the production rate (v) affects the change in the amount of raw material [S]. Graphically, the M-M equation can be represented as $\frac{dv}{d[S]} \geq 0$, which means the production rate [v] increases when we add more raw materials [S], and $\frac{d^2v}{d[S]^2} \leq 0$, where the rate of production slows down by adding more raw materials.

In this paper, M-M equations are used to understand how the mean number of customers in the system (L) changes at the change in traffic intensity (ρ) or arrival rate (λ). The M-M Model exhibits the saturation behaviour observed in real-world phenomena in many domains, including queueing systems. As the system load intensifies, the system performance, like average queue length, rises rapidly at first but gradually stabilises due to capacity constraints. This pattern is conceptually similar to enzyme saturation in biochemical systems.

In the field of machine learning, the M-M equation has been adopted as a regression model to capture relationships marked by threshold effects. In this research, we aim to address this by applying the Michaelis–Menten model as a non-linear regression approach to enhance performance in bulk-service queueing systems.

Solving the reaction mechanism [29]:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P \tag{2}$$

Equation (1) is obtained, mapped as

$$L(\rho) = \frac{L_{max} \times \rho}{K_{max} + \rho} \tag{3}$$

A second-order variant, which exhibits a sigmoidal shape, is defined as

$$v = \frac{V_{max} \times [S]^2}{K_{max} + [S]^2} \tag{4}$$

This form introduces an initial convex phase followed by a concave saturation, better capturing systems where L grows slowly at very low ρ before accelerating. Both forms are fitted to data via non-linear least squares (NLS) by minimising

$$\sum [L_i - v(\rho_i; V_{max}, K_{max})]^2 \tag{5}$$

4. Bulk-Service Queueing System

This section provides a detailed analysis of the mathematical framework underlying the study and is divided into five subsections. The first subsection analyses the $M/M^b/1/N$ queueing model. The second develops the derivation and analysis of the $M/M^b/c/N$ model, while the third presents its associated performance measures. The fourth subsection reports the numerical results obtained through LU decomposition, which further illustrates and carries out sensitivity analyses for the two analytical queueing models. The final subsection examines the extent to which the behaviour of these models resembles the pattern captured by the M-M form.

4.1. Analysis of $M/M^b/1/N$ Model

Consider a queueing model in which arrival follows a Poisson process with rate λ. Service is provided in fixed-size batches of size b. Service time distribution follows exponential distribution with rate μ. If $n < b$, then no service occurs because the batch requirement is not met. If $n \geq b$, then a batch of size b is serviced. Maximum number of waiting customers is N_q, and at most, b customers can be served at a time. Hence, the total number of customers (waiting and in service) that the system can hold is $N = N_q + b$. Let $p_n(t)$ be the steady-state probability that there are n customers in the entire system (waiting and in service), $n = 0, 1, ..., N$. The state variable n always denotes the total number of customers in the system.

The differential-difference equation satisfied by $p_n(t)$ is obtained using the following Chapmann–Kolmogorov dynamics.

Balance equations are given by:

$$\begin{aligned} -\lambda p_0 + \mu p_b &= 0 \\ -\lambda p_j + \lambda p_{j-1} + \mu p_{j+b} &= 0, \quad 1 \leq j \leq b-1 \\ -(\lambda+\mu) p_j + \lambda p_{j-1} + \mu p_{j+b} &= 0, \quad b \leq j \leq N_q \\ -(\lambda+\mu) p_j + \lambda p_{j-1} &= 0, \quad N_q + 1 \leq j \leq N-1 \\ -\mu p_j + \lambda p_{j-1} &= 0, \quad j = N \end{aligned} \tag{6}$$

For arbitrary batch size b and capacity N, the generator matrix Q is an $(N+1) \times (N+1)$ matrix, which can be seen in [5].

4.2. Analysis of $M/M^b/c/N$ Queueing Model

Consider a queueing system in which customers arrival follows a Poisson process with rate $\lambda > 0$. The system is served by c identical, parallel servers, each of which processes customers in fixed-size batches of b. A server becomes active only when at least b customers are available for it to draw upon, and once activated, it serves the entire batch as a single unit. The batch service time follows an exponential distribution with rate μ per server. The system has finite capacity N (waiting + service). This model takes the form of the standard $M/M/c/K$ queue by replacing batch service of fixed size b = 1.

Let $X_q(t)$ be the number of customers waiting in the queue at time t, i.e.,

$$X_q(t) = \begin{cases} 0 \quad to \quad (b-1), & \text{up to (c} - 1) \text{ server} \\ 0 \quad to \quad (N_q), & \text{for } c^{th} \text{ server} \end{cases}$$

where $N_q = N - (b * c)$.

$X_s(t)$ denotes the number of servers busy at time t, i.e.,

$$X_s(t) = \begin{cases} 0, & \text{if server is idle} \\ r, & \text{if } r \text{ servers are busy} \end{cases}$$

Clearly, $X_q(t), X_s(t)$ constitute a two-dimensional continuous-time Markov chain with state space

$\{(j,r) : 0 \leq j \leq (b-1), 0 \leq r \leq (c-1)\} \cup \{(j,r) : 0 \leq j \leq (N_q), r = c\}$.

The joint state probabilities at time t are defined as:

- Let $p_{0,r}(t)$ be the probability that no customers are waiting in the queue and r servers are busy and
- $p_{j,r}(t)$ be the probability that j customers are waiting in the queue and r servers are busy for $1 \leq j \leq b-1, 0 \leq r \leq c-1$, and for $1 \leq j \leq N_q, r = c$.

Now, each balance equations can be written as:

$$\begin{aligned}
-\lambda p_{0,0} + \mu p_{0,1} &= 0 \\
-\lambda p_{j,0} + \lambda p_{j-1,0} + \mu p_{j,1} &= 0, \quad 1 \le j \le b-1 \\
-(\lambda + r\mu) p_{0,r} + \lambda p_{b-1,r-1} + (r+1)\mu p_{0,r+1} &= 0, \quad 1 \le r \le c-1 \\
-(\lambda + c\mu) p_{0,c} + \lambda p_{b-1,c-1} + c\mu p_{b,c} &= 0 \\
-(\lambda + r\mu) p_{j,r} + \lambda p_{j-1,r} + (r+1)\mu p_{j,r+1} &= 0, \quad 1 \le r \le c-1, \quad 1 \le j \le b-1 \\
-(\lambda + c\mu) p_{j,c} + \lambda p_{j-1,c} + c\mu p_{j+b,c} &= 0, \quad 1 \le j \le N_q - b \\
-(\lambda + c\mu) p_{j,c} + \lambda p_{j-1,c} &= 0, \quad N_q - b + 1 \le j \le N_q - 1 \\
-c\mu p_{N_q,c} + \lambda p_{N_q-1,c} &= 0, \quad j = N_q
\end{aligned} \tag{7}$$

To make the construction of Q transparent, the states are grouped into levels according to the number of busy servers $X_s = r$. Levels $0, 1, \ldots, c-1$ each contain b states (j, r), $j = 0, \ldots, b-1$, while level c contains $N_q + 1$ states (j, c), $j = 0, \ldots, N_q$. Arranging the states level by level, following Figure 1, yields a block-tridiagonal Q composed of three types of blocks

- B_r collects the transitions within level r, namely, arrivals that add a customer to the queue, together with the diagonal exit rates;
- A_r collects transitions from level r to level $r+1$, which occur when an arrival completes a batch of size b and a free server starts service;
- C_r collects transitions from level r to level $r-1$, which occur when one of the r busy servers completes service at rate $r\mu$.

The level $r = c$ has a different dimension because the waiting room is only used when all servers are busy. Hence, A_{c-1}, C_c, and B_c are rectangular or larger boundary blocks with dimensions specified in Figure 2. Figure 2 shows how each arrow class in Figure 1 maps to a block of Q.

Generator matrix Q The infinitesimal generator matrix Q of the continuous-time Markov chain $X_q(t), X_s(t); t \ge 0$ is obtained directly from the balance equations governing the transitions of the system and is given in Figure 2.

Block B_0 $(b \times b)$:

$$B_0 = \begin{pmatrix} -\lambda & \lambda & 0 & \cdots & 0 \\ 0 & -\lambda & \lambda & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\lambda & \lambda \\ 0 & 0 & \cdots & 0 & -\lambda \end{pmatrix}_{b \times b} \tag{8}$$

Block A_0 $(b \times b)$:

$$A_0 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \lambda & 0 & \cdots & 0 \end{pmatrix}_{b \times b} \tag{9}$$

Block C_1 $(b \times b)$:

$$C_1 = \mu I_b = \begin{pmatrix} \mu & 0 & \cdots & 0 \\ 0 & \mu & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mu \end{pmatrix}_{b \times b} \tag{10}$$

Block B_1 $(b \times b)$:

$$B_1 = \begin{pmatrix} -(\lambda+\mu) & \lambda & 0 & \cdots & 0 \\ 0 & -(\lambda+\mu) & \lambda & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -(\lambda+\mu) & \lambda \\ 0 & 0 & \cdots & 0 & -(\lambda+\mu) \end{pmatrix}_{b\times b} \tag{11}$$

General block A_r $0 \le r < c-1$ $(b \times (b))$:

$$[A_r]_{i,i'} = \begin{cases} \lambda, & i = b-1,\ i' = 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

General block C_r, $1 \le r \le c-1$ $(b \times b)$:

$$C_r = r\mu\, I_b \tag{13}$$

General block B_r, $1 \le r \le c-1$ $(b \times b)$:

$$[B_r]_{i,i} = -(\lambda + r\mu), \qquad [B_r]_{i,i+1} = \lambda, \qquad i = 0, 1, \ldots, b-2 \tag{14}$$

Boundary block A_{c-1} $(b \times (N_q+1))$:

$$[A_{c-1}]_{i,j'} = \begin{cases} \lambda, & i = b-1,\ j' = 0 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Block C_c $((N_q+1) \times b)$:

$$[C_c]_{j,i'} = \begin{cases} c\mu, & j = i', \quad 0 \le j \le b-1 \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Block B_c $((N_q+1) \times (N_q+1))$:

$$[B_c]_{j,j+1} = \lambda, \quad j = 0, \ldots, N_q - 1; \qquad [B_c]_{j,j-b} = c\mu, \quad j = b, \ldots, N_q \tag{17}$$

$$[B_c]_{j,j} = \begin{cases} -(\lambda + c\mu), & j = 0, \ldots, N_q - 1 \\ -c\mu, & j = N_q \end{cases} \tag{18}$$

$$B_c = \begin{pmatrix} -(\lambda+c\mu) & \lambda & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & -(\lambda+c\mu) & \lambda & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & & & & \vdots \\ 0 & \cdots & 0 & -(\lambda+c\mu) & \lambda & 0 & \cdots & 0 & 0 \\ c\mu & 0 & \cdots & 0 & -(\lambda+c\mu) & \lambda & \cdots & 0 & 0 \\ 0 & c\mu & 0 & \cdots & 0 & -(\lambda+c\mu) & \ddots & & \vdots \\ \vdots & & \ddots & & & & \ddots & \ddots & 0 \\ 0 & \cdots & & c\mu & 0 & \cdots & 0 & -(\lambda+c\mu) & \lambda \\ 0 & \cdots & & 0 & c\mu & 0 & \cdots & 0 & -c\mu \end{pmatrix} \tag{19}$$

If this Markov chain is positive recurrent, then there exists a stationary vector $\boldsymbol{\pi} = [\boldsymbol{\pi_0}, \boldsymbol{\pi_1}, \boldsymbol{\pi_2}, \ldots, \boldsymbol{\pi_c}]$ that denotes row vector for stationary probabilities where $\boldsymbol{\pi_r} = [\boldsymbol{p_{0,r}}, \boldsymbol{p_{1,r}}, \ldots, \boldsymbol{p_{(b-1),r}}]$ for $0 \le r \le (c-1)$ and $\boldsymbol{\pi_c} = [\boldsymbol{p_{0,c}}, \boldsymbol{p_{1,c}}, \ldots, \boldsymbol{p_{N_q,c}}]$ such that

$$0 = \boldsymbol{\pi} Q, \boldsymbol{\pi}\mathbf{1} = \mathbf{1}, \tag{20}$$

where **1** is a column vector of ones. The state space has finite capacity N; arrivals are blocked when $X(t) = N$. The chain is therefore irreducible and positive recurrent, and the stationary probability vector is uniquely determined. The generator matrix Q has order $N = 1$, where $N + 1 = bc + N_q + 1$.

$$\boldsymbol{\pi}\,\tilde{Q} = \boldsymbol{e}_{N+1}, \tag{21}$$

where $\boldsymbol{e}_{N+1} = [0, 0, 0, ..., 1]$. Here, $\tilde{Q}$ is obtained from the generator matrix Q by replacing its last column with the column vector **1**. For an irreducible finite chain, $\tilde{Q}$ is nonsingular, so Equation (21) has a unique solution. This solution is the joint stationary distribution of all states (j, r).

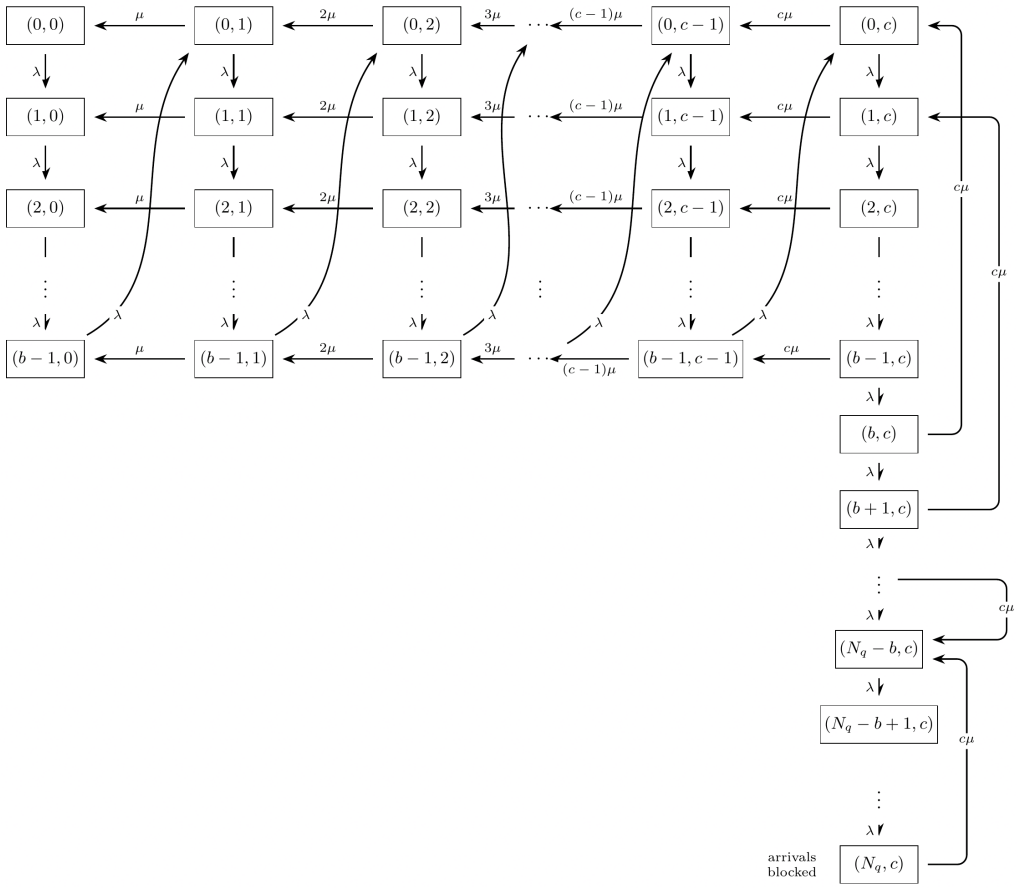


Figure 1. State-transition diagram illustrating the block structure of the infinitesimal generator matrix Q for the $M/M^b/c/N$ queueing model.

Equation (21) is solved by LU decomposition (Gaussian elimination) by obtaining the transpose of Equation (21), from which we obtain $\tilde{Q}^T\boldsymbol{\pi}^{\boldsymbol{T}} = e^T_{N+1}$. After that, $\tilde{Q}^T$ is factorised as $\tilde{Q}^T = LU$, where L is a unit lower-triangular matrix, and U is an upper-triangular matrix. The solution is obtained in two successive steps involving triangular systems. Forward substitution yields y from $Ly = e^T_{N+1}$, and backward substitution then gives $\boldsymbol{\pi}^{\boldsymbol{T}}$ from $U\boldsymbol{\pi}^{\boldsymbol{T}} = y$. This yields every component $p_{j,r}$ of $\boldsymbol{\pi}$ in a single pass, and no iteration is needed. The computed vector was checked by verifying that all $p_{j,r} \geq 0$ and $\sum p_{j,r} = 1$.

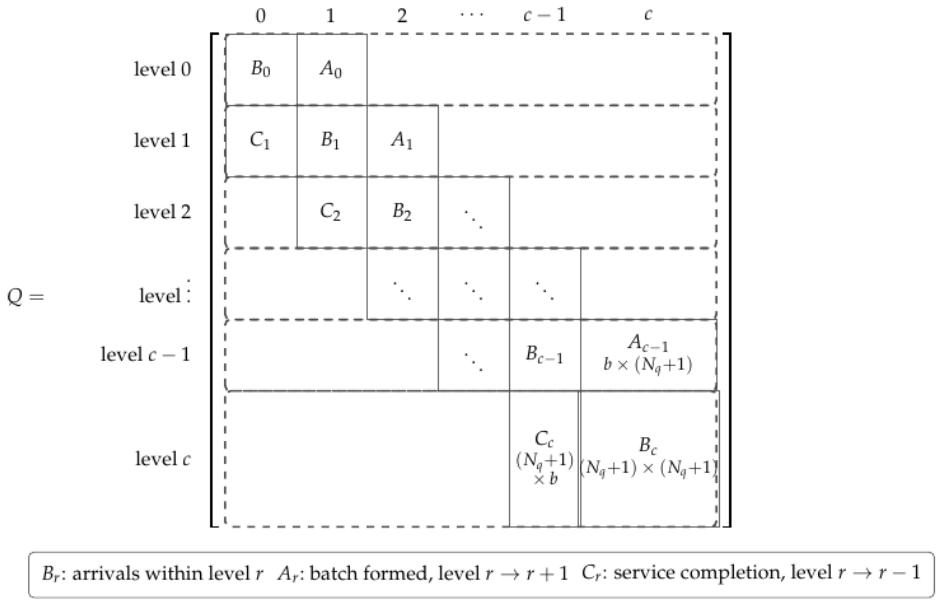


Figure 2. Block structure of the generator matrix Q. Each dashed box represents a level $r = X_s$ of the state transition diagram. All blocks are $b \times b$ except those labelled with their dimensions.

4.3. Performance Measures

Performance measures for $M/M^b/c/N$ are given below:

(i) Mean number of customers in the queue:

$$L_q = \sum_{j=0}^{b-1} \sum_{r=0}^{c-1} j\, p_{j,r} + \sum_{j=0}^{N_q} j\, p_{j,c} \tag{22}$$

(ii) Mean number of customers in the system:

$$L = \sum_{j=0}^{b-1} \sum_{r=0}^{c-1} (j + rb)\, p_{j,r} + \sum_{j=0}^{N_q} (j + cb)\, p_{j,c} \tag{23}$$

(iii) Mean number of customers in service:

$$L_s = \sum_{j=0}^{b-1} \sum_{r=0}^{c-1} (rb)\, p_{j,r} + \sum_{j=0}^{N_q} (cb)\, p_{j,c} \tag{24}$$

(iv) Blocking Probability:

$$p_{block} = p_{N_q,c} \tag{25}$$

(v) Effective arrival rate:

$$\lambda_{eff} = \lambda(1 - p_{block}) \tag{26}$$

(vi) Utilisation factor of the system:

$$\rho = \frac{\lambda}{bc\mu} \tag{27}$$

(vii) Utilisation factor of the service station:

$$\rho' = \frac{\lambda_{eff}}{bc\mu} \tag{28}$$

(viii) Mean waiting time in the system (using Little's law):

$$W = \frac{L}{\lambda_{eff}} \tag{29}$$

(ix) Mean waiting time in the queue (using Little's law):

$$W_q = \frac{L_q}{\lambda_{eff}} \tag{30}$$

(x) Average number of customers in the system when the server is idle:

$$L^{Idle} = \frac{\sum_{j=0}^{b-1} j * p_{j,0}}{p^{Idle}}, where, \tag{31}$$

$$p^{Idle} = \sum_{j=0}^{b-1} p_{j,0} \tag{32}$$

(xi) Average number of customers in the system given r servers are busy:

$$L^{busy} = \frac{\sum_{j=0}^{b-1} \sum_{r=1}^{c-1} (j + rb)\, p_{j,r} + \sum_{j=0}^{N_q} (j + cb)\, p_{j,c}}{p^{busy}}, where \tag{33}$$

$$p^{busy} = \sum_{j=0}^{b-1} \sum_{r=1}^{c-1} p_{j,r} + \sum_{j=0}^{N_q} p_{j,c} \tag{34}$$

(xii) Average number of customers in the queue given r servers are busy:

$$L_q^{busy} = \frac{\sum_{j=0}^{b-1} \sum_{r=1}^{c-1} j\, p_{j,r} + \sum_{j=0}^{N_q} j\, p_{j,c}}{p^{busy}} \tag{35}$$

(xiii) Average number of customers in service given r servers are busy:

$$L_{sr}^{busy} = \frac{\sum_{j=0}^{b-1} \sum_{r=1}^{c-1} (rb)\, p_{j,r} + \sum_{j=0}^{N_q} (bc)\, p_{j,c}}{p^{busy}} \tag{36}$$

4.4. Numerical Results

This section presents numerical results in the form of Table 3, which gives the steady-state joint probability distribution for $M/M^b/c/N$ for b = 10, c = 7, $\lambda = 50$, and N = 90 with $\mu = 1$. The joint probabilities in Table 3 are obtained by solving Equation (21) with the LU decomposition procedure described in Section 4.2, and the column totals sum to 1, as required by the normalisation condition. All the performance measures given above are evaluated for the given parameters and are represented at the bottom of Table 3. For these parameters, the offered load is $\rho = \lambda/(bc\mu) = 0.714$. The column totals give the distribution of the number of busy servers. For $r = 5$, total is 0.2347, and $r = 4, 5, 6$ together carry 63.1% of the probability, so the system usually works with four to six of the seven servers busy. All seven servers are busy with probability 0.1828, while all are idle with probability only $p^{idle} = 0.000778$. The mean number in service, $L_s = 49.005$, equals λ_{eff}/μ. Within a column, the probabilities increase with j for $r = 0, \ldots, 4$, for example, from 0.000284 to 0.001920 for $r = 1$, because customers accumulate while a batch is being formed. They are nearly flat for $r = 5$ and decrease with j for $r = 6$ and $r = 7$, apart from a small step at $j = b$ in the last column. In the last column, the probability falls to 0.019893 at $j = 20$. In that state, the system is full, arrivals are blocked, and only a service completion can leave it, so $p_{block} = p_{N_q,c} = 0.01989$, and about 2% of arriving customers are lost.

Table 3. Joint Distribution Table for $M/M^b/c/N$ with $b = 10$, $c = 7$, $N = 90$, $\lambda = 50$, and $\mu = 1$.

j/r	$r = 0$	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$
$j = 0$	0.000006	0.000284	0.002339	0.008542	0.017765	0.023713	0.021798	0.014509
$j = 1$	0.000013	0.000388	0.002795	0.009466	0.018659	0.023867	0.021175	0.013703
$j = 2$	0.000024	0.000509	0.003287	0.010402	0.019493	0.023935	0.020516	0.012876
$j = 3$	0.000037	0.000649	0.003815	0.011342	0.020264	0.023922	0.019823	0.012045
$j = 4$	0.000053	0.000808	0.004377	0.012283	0.020970	0.023831	0.019102	0.011222
$j = 5$	0.000072	0.000987	0.004971	0.013218	0.021608	0.023667	0.018359	0.010424
$j = 6$	0.000096	0.001187	0.005596	0.014144	0.022177	0.023436	0.017598	0.009650
$j = 7$	0.000124	0.001409	0.006250	0.015055	0.022677	0.023141	0.016826	0.008910
$j = 8$	0.000157	0.001653	0.006929	0.015947	0.023107	0.022788	0.016049	0.008205
$j = 9$	0.000196	0.001920	0.007633	0.016815	0.023468	0.022382	0.015272	0.007540
$j = 10$	–	–	–	–	–	–	–	0.009057
$j = 11$	–	–	–	–	–	–	–	0.007945
$j = 12$	–	–	–	–	–	–	–	0.006969
$j = 13$	–	–	–	–	–	–	–	0.006113
$j = 14$	–	–	–	–	–	–	–	0.005362
$j = 15$	–	–	–	–	–	–	–	0.004704
$j = 16$	–	–	–	–	–	–	–	0.004126
$j = 17$	–	–	–	–	–	–	–	0.003619
$j = 18$	–	–	–	–	–	–	–	0.003175
$j = 19$	–	–	–	–	–	–	–	0.002785
$j = 20$	–	–	–	–	–	–	–	0.001989
Total	0.000778	0.009792	0.047993	0.127214	0.210189	0.234681	0.186518	0.182835
$L = 54.352237$ $L_q = 5.346902$			$L_s = 49.005335$ $p_{\text{block}} = 0.019893$			$p^{\text{idle}} = 0.000778$ $p^{\text{busy}} = 0.999222$		
$\lambda_{\text{eff}} = 49.005335$ $W_q = 0.109109$			$L^{\text{idle}} = 6.698166$ $L^{\text{busy}} = 54.389358$			$L_q^{\text{busy}} = 5.345850$ $L_s^{\text{busy}} = 49.043509$		
$L_{s1}^{\text{busy}} = 10.000000$ $L_{s2}^{\text{busy}} = 18.305484$			$L_{s3}^{\text{busy}} = 26.347187$ $L_{s4}^{\text{busy}} = 33.608743$			$L_{s5}^{\text{busy}} = 39.715919$ $L_{s6}^{\text{busy}} = 44.350179$		

After numerically obtaining the joint distribution table, a sensitivity analysis is performed for $M/M^b/1/N$, illustrating the graphs of L versus ρ for different values of buffer capacity N and server capacity c. These plots offer insight into how each parameter individually influences the congestion behaviour of the system, thereby enabling a comprehensive understanding of the model's performance characteristics under diverse operating conditions.

The graphs in Figure 3 show the results of the mean number of customers for varying traffic intensity $1 \leq \rho \leq 24$ and $\mu = 8$ with system capacity of $N = \{2, 4, 6, 8\}$ for $M/M^2/1/N$, indicating an increase in L as N increases. Similarly, the graphs in Figure 4 provide the results for $M/M^2/c/N$ for the mean number in the system L and varying number of servers $c = \{1, 2, 3, 4\}$ for varying ρ, with $\mu = 8$ and $N = 8$. This figure demonstrates that all curves rise sharply and then converge near $L \approx 7.5$ for $\rho = 3$.

A similar analysis is carried out for the $M/M^b/c/N$ model, wherein three sets of plots are generated to depict the variation in the mean system size L with respect to the traffic intensity ρ, examined separately for varying buffer capacity N, batch size b, and number of servers c.

Figure 5 shows mean number L versus ρ for b = 3, c = 2, and $N \epsilon \{6, 8, 10, 12\}$ with $\mu = 8$. The curves exhibit the saturating concave pattern characteristic of bulk-service systems. As compared to the standard $M/M/1/N$ systems, which correspond to the b = 1 case validated in Section 5.3, two effects are visible. Firstly, the curves are shifted rightward because the server remains idle until a batch is formed. Secondly, the initial slope near $\rho = 0$ is higher because service cannot begin immediately upon the first arrival.

Figure 6 shows that all curves rise steeply and converge tightly near $L \approx 29.5$ by $\rho = 3$. Again, this figure shows that the curves rise more gradually and flatten at $\rho = 3$, because b = 3 creates greater queueing/blocking sensitivity to the number of servers.

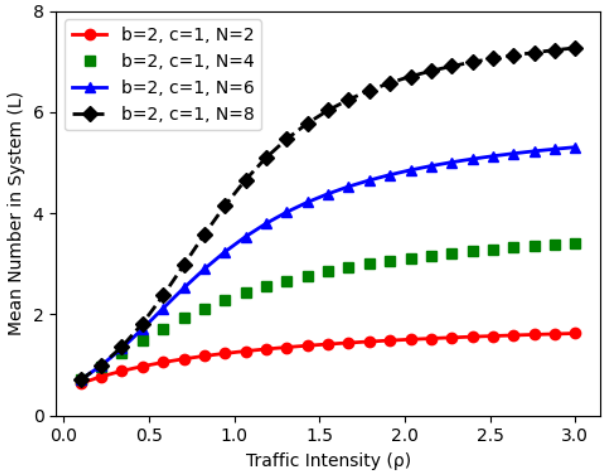


Figure 3. Effects of ρ on L with varying N for fixed b = 2 and c = 1.

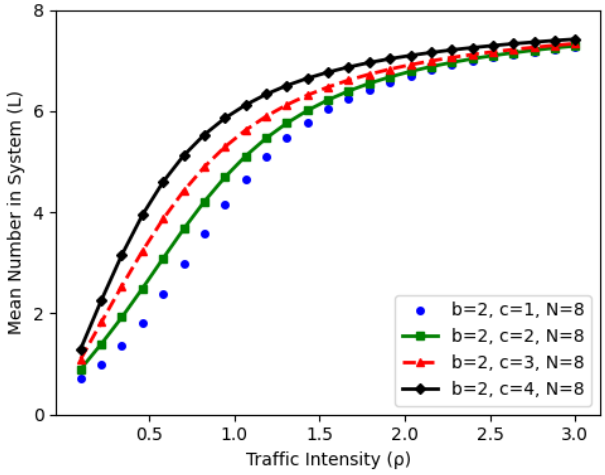


Figure 4. Effects of ρ on L with varying c for fixed b = 2 and N = 8.

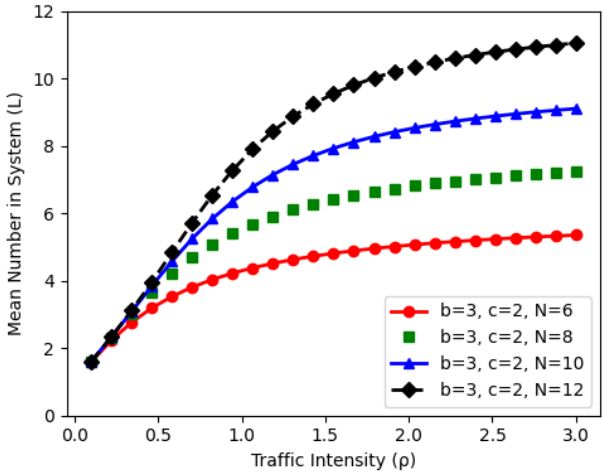


Figure 5. $M/M^3/2/N$ system: Effects of ρ on L with varying N for fixed b = 3 and c = 2.

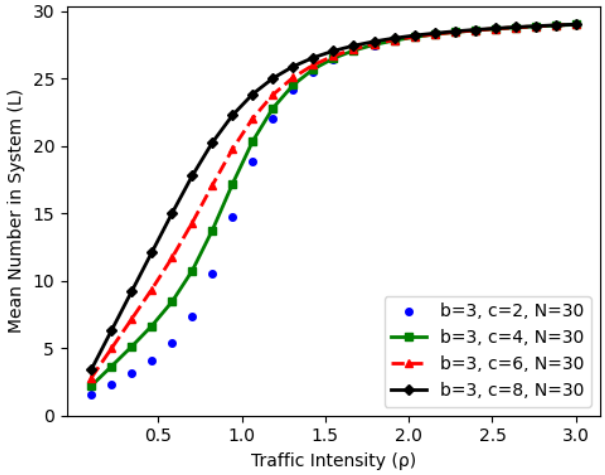


Figure 6. $M/M^3/c/N$ system: Effects of ρ on L with varying c for fixed b = 3 and N = 30.

Figure 7 shows the plot of the mean number in system L with traffic intensity ρ for different values of b. At low traffic intensity, L increases with batch sizes, and as ρ increases, the curves converge and eventually overlap near saturation.

Here, $\lambda = \rho bc\mu$ with $\mu = 8$ is common for Figures 3–7. (i) At light load ($\rho = 0.1$), L is determined by batch formation rather than by capacity. For $b = 2$, $c = 1$, it is 0.70, 0.71, 0.71 for $N = 4, 6, 8$; for $b = 3$, $c = 2$, it is about 1.6 for every N, and it grows with b (4.00, 4.90, 5.80, and 6.70 for $b = 5, \ldots, 8$ with $c = 4$, $N = 40$). (ii) At heavy load ($\rho = 3$), L approaches but stays below N. For $N \geq 6$, it reaches 88–97% of N, i.e., $L = 7.27$ for $N = 8$ with $b = 2$, $c = 1$; $L = 11.04$ for $N = 12$ with $b = 3$, $c = 2$; and about 29.0 for $N = 30$ with $b = 3$). (iii) The effect of N and c is largest at moderate load. For $b = 3$, $c = 2$, and $\rho = 1$, $L = 4.30, 5.53, 6.55, 7.57$ for $N = 6, 8, 10, 12$, whereas all $N \geq 8$ give the same $L = 1.60$ at $\rho = 0.1$. For $b = 2$, $N = 8$, and $\rho = 1$, $L = 4.39, 4.89, 5.45, 5.98$ for $c = 1, \ldots, 4$, but the four curves differ by only 0.15 at $\rho = 3$ (7.27,7.28,7.33,7.42). The effect of the batch size reverses with load, at $\rho = 0.1$, larger b gives larger L, whereas at $\rho = 3$, it gives slightly smaller L (38.50, 38.27, 38.07, and 37.92 for $b = 5, \ldots, 8$), because each completion removes a larger batch from a full system.

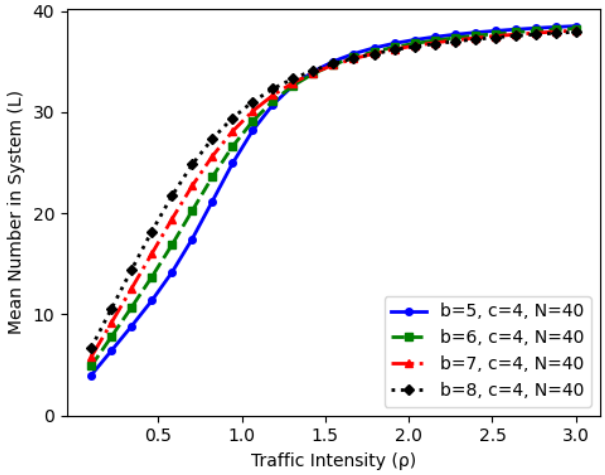


Figure 7. $M/M^b/4/N$ system: Effects of ρ on L with varying b for fixed c = 4 and N = 40.

4.5. Nature/Behaviour Reflecting in the Queueing Models

In this section, the results obtained from the graphical representations are subsequently analysed to characterise the underlying queueing system behaviour, reflecting the pattern captured by the M-M regression model, described in Section 3.

- Monotonicity: For fixed b, c, N, as ρ increases from 0 to ∞, more customers arrive per unit time. By a coupling argument on the underlying Markov chains at two traffic intensities $\rho_1 < \rho_2$, the stationary distribution at ρ_2 stochastically dominates that at ρ_1, ensuring $L(\rho_2) \geq L(\rho_1)$.
- Boundary conditions and Saturation: Graphical outputs are examined in detail to draw inferences regarding the behaviour of the queueing model. When $\rho = 0$, no arrivals occur and the system is empty, so $\mathrm{L}(0) = 0$. As $\rho \to \infty$, the arrival rate overwhelms all service capacity and the system fills to capacity:

$$\lim_{\rho \to \infty} L \leq N.$$

  These two boundary conditions, $\mathrm{L}(0) = 0$ and $\lim_{\rho \to \infty} L \leq N$, exactly match those of the M-M function v when $V_{max} \approx N$.
- Concavity for Large Traffic: Once $\rho > 1$, the queue reaches towards its buffer ceiling. In this regime, the marginal increase in L per unit increase in ρ diminishes because additional arriving customers find the buffer increasingly full and are lost (blocked). The concavity for $\rho > 1$ combines with the initial convexity near $\rho = 0$, which can be observed from the graphs obtained for numerical results.

5. Machine Learning Framework

The framework below is a supervised machine learning algorithm. The inputs are traffic intensities, the targets are queue performance measures generated by the queueing model, and the learned function predicts these measures for new loads. Algorithm 1 gives its steps. In this section, we outline the overall framework adopted in this paper. First, the mean queue length values are obtained from the analytical solutions of the $M/M^b/1/N$ and $M/M^b/c/N$ systems across a range of traffic intensities. For each configuration (b, c, N), the training data consist of pairs $\{(\rho_i, L_i)\}$ for $\{i = 1, \ldots, n\}$. Values ρ_i are sampled on the equally spaced grid from $(0.1, \ldots, 5.0)$, i.e., for n = 25 points. L_i is computed by exact numerical solution of the global balance equations, assembled as a finite linear system, and solved by the LU decomposition method. After the data generation process, these values are then used to estimate the parameters of the Michaelis–Menten model through non-linear regression. The resulting fit is assessed using the coefficient of determination, and its accuracy is further confirmed by comparing the fitted curve against the analytical solution graphically, followed by the process of validation.

5.1. Fitting Procedure

Given the training set, the M-M parameters (V_{max}, K_{max}) of the M-M model are estimated by non-linear least squares regression, minimising the sum of the squared residuals between the analytically computed values and the model prediction values given by SS_{res}. One important observation regarding fitting the bulk-service queueing models behaviour to the M-M model is $V_{max} \approx N$. The optimisation was performed by using curve_fit of the sci.optimize module in Python 3.13. To ensure convergence, the initial parameter estimates were set heuristically, with V_{max} initialised at twice the maximum observed value of L in the dataset and K_{max} initialised at unity. The algorithm iteratively refined these estimates until convergence to the least-squares optimum is achieved. Parameter estimation was carried out without imposing bounds on V_{max} and K_{max}. Consequently, curve_fit employs its default Levenberg–Marquardt algorithm, with the Jacobian approximated by forward

finite differences and the default convergence tolerances of 10^{-8}. For the second model, ρ_i is replaced by ρ_i^2 in the Equation (1).

5.2. Goodness of Fit Assessment

The adequacy of the fitted model is evaluated using the coefficient of determination defined as [27]

$$R^2 = [1 - \frac{SS_{res}}{SS_{tot}}] \tag{37}$$

where $SS_{res} = \sum_{i=1}^{n} (\hat{L}_i - L_i)^2$ is the residual sum of squares between the analytical values and the fitted model predictions, and $SS_{tot} = \sum_{i=1}^{n} (L_i - \bar{L}_i)^2$ is the total sum of squares of the analytical values about their mean.

An R^2 value close to 100% indicates that the empirical M-M model closely reproduces the trend of the exact analytical solution across the sampled range of traffic intensities.

5.3. Prediction and Validation

For validation, the first step we have carried out is reducing the queueing model to known closed-form cases. By setting $b = 1$ and comparing it with the calculations and plots obtained in [3] for the standard queueing system $M/M/c/N$, the bulk-service queueing model outputs are validated. Monotonicity behaviour is verified for each configuration, finding L to be non-decreasing in ρ. Moreover, L always satisfies the condition $0 \leq L \approx N$.

6. Approximating the Mean Number of Customers L Using M-M Model

This section presents the approximation of the mean number in the system L, using the Michaelis–Menten model, organised into three subsections. Section 6.1 addresses the $M/M^b/1/N$ and $M/M^b/c/N$ models in detail, reporting the fitted parameters of both the first-order and second-order M-M models in separate tables, accompanied by two comparison plots for each order illustrating the fit against the analytical solution. Section 6.1 concludes with a table summarising the effect of buffer size N on the fitted parameters for both the first-order and second-order M-M models. Sections 6.2 and 6.3 provide the structure for the $M/M^b/c/N$, each reporting parameter tables for the first-order and second-order fits along with two comparison plots. Each plot compares the analytical solution (LU decomposition) with the M-M fit for both orders. Section 6.2 concludes with tables that capture the effect of multiple servers c on the fitted plot for L versus ρ for both first-order and second-order. Finally, Section 6.3 closes with tables summarising how the size of batch size b influences the fitted curve of L versus ρ, presenting the corresponding parameter values for both the first-order and second-order fits.

6.1. Effects of Buffer Size N

Following the formulation of the non-linear M-M model for the standard queueing system, a fitting procedure is carried out for a buffer size N to identify the best-fit regression model describing the relationship between traffic intensity and mean number in the system, for both the first-order and second-order variants.

6.1.1. For $M/M^b/1/N$ System

This section gives the parameter estimates of V_{max} and K_{max} in the M-M model, which best approximates the mean number of customers L in the bulk service queueing system $M/M^b/1/N$. Table 4 summarises the resulting coefficients V_{max} and K_{max}, corresponding to the M-M model that provides the closest match to the mean values of L derived for the $M/M^2/1/N$ bulk-service system, as illustrated in Figure 8. Additionally, Table 4 provides

the coefficient of determination R^2, which serves as a statistical measure of how well the M-M models fit the system data.

Table 4. Best-fit parameters of the first-order M-M model for $M/M^b/1/N$ systems.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
M/M^2/1/6	7.09	1.08	98.3	0.179	0.152
M/M^2/1/8	10.00	1.23	96.3	0.390	0.325
M/M^2/1/10	12.94	1.31	94.3	0.645	0.535
M/M^2/1/12	15.87	1.37	92.5	0.930	0.767

Figures 8 and 9 present a comparison between the best-fit first-order M-M model and the analytical solutions for the mean number of customers in the bulk-service queueing model, $M/M^2/1/8$ and $M/M^2/1/10$, respectively.

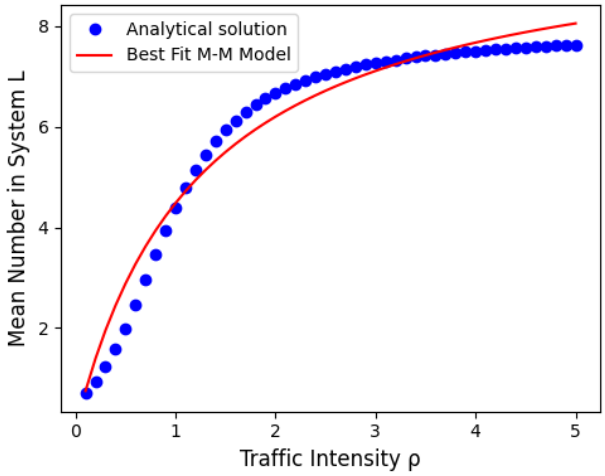


Figure 8. $M/M^2/1/8$ system: Comparison between analytical solution (LU) and best-fit first-order M-M model.

The findings indicate that the first-order M-M model effectively captures the mean number of customers L in the queueing systems of the bulk services discussed above, as shown in Figures 8 and 9.

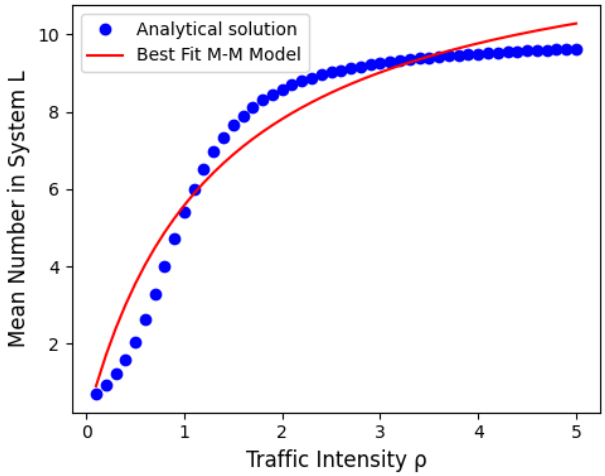


Figure 9. $M/M^2/1/10$ system: Comparison between analytical solution (LU) and best-fit first-order M-M model.

In Table 5, the second-order M-M model coefficients are listed that represent the best-fit for the mean number of customers L across the $M/M^2/1/N$ systems.

Table 5. Best-fit parameters for second-order M-M system for $M/M^2/1/N$ system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
$M/M^2/1/6$	5.69	0.64	98.5	0.176	0.105
$M/M^2/1/8$	7.87	0.75	99.5	0.149	0.067
$M/M^2/1/10$	10.08	0.82	99.4	0.216	0.168
$M/M^2/1/12$	12.30	0.86	98.9	0.369	0.295

Figures 10 and 11 give the second-order M-M model comparison with the analytical solution for $M/M^2/1/8$ and $M/M^2/1/10$, respectively. Table 4 shows that for the first-order M-M model of the $M/M^b/1/N$ system, V_{max} is approximately equal to N in all cases except for $N = 12$. For the second-order model, Table 5 shows that V_{max} is less than or approx to N for all four capacities, $N = \{6, 8, 10, 12\}$. The results for the second-order M-M model demonstrate generally higher goodness-of-fit R^2 as compared to the first-order M-M model, which can be compared from the given Tables 4 and 5.

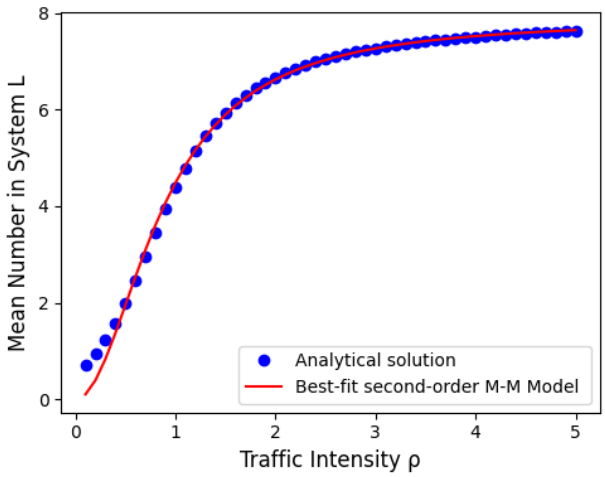


Figure 10. $M/M^2/1/8$ system: Comparison between analytical solution (LU) and best-fit second-order M-M model.

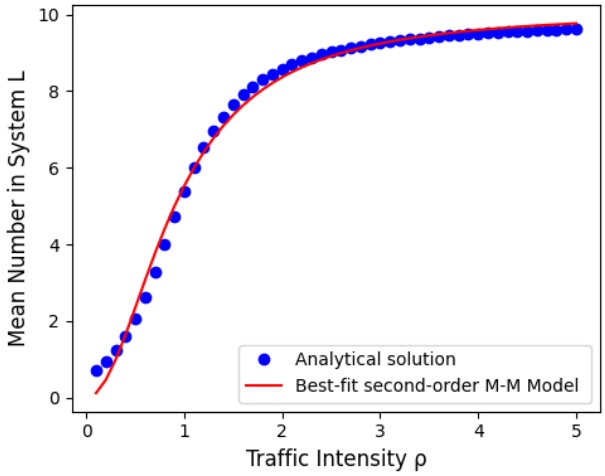


Figure 11. $M/M^2/1/10$ system: Comparison between analytical solution (LU) and best-fit second-order M-M model.

6.1.2. For $M/M^b/c/N$ Queueing Model

This section gives the best approximation of the mean number of the system L, in the bulk service $M/M^b/c/N$ queueing system, providing the M-M model parameter estimates

for each variant. Table 6 reports the best-fit parameters V_{max} and K_{max} and R^2 values for the first-order M-M Model applied to $M/M^b/c/N$ with $b = 3$ and $c = 2$ for different values of N.

Table 6. Best-fit parameters for first-order M-M system for M/M^3/2/N system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
M/M^3/2/10	11.18	0.72	98.8	0.229	0.195
M/M^3/2/12	14.01	0.86	97.7	0.411	0.348
M/M^3/2/14	16.89	0.96	96.4	0.638	0.533
M/M^3/2/16	19.80	1.04	95.1	0.895	0.745

The graphs in Figures 12 and 13 illustrate the comparison between the analytical solution and first-order M-M model results for the mean number of customers L in $M/M^3/2/12$ and $M/M^3/2/16$, respectively.

From Table 6, it can be observed that R^2 decreases slightly as N increases for fixed b, indicating that the first-order M-M model is a less perfect fit for large buffers. This feature is correctable by the second-order variant, as Table 7 shows, where R² rises from 97.0% to 99.2% as N increases from 10 to 16.

Table 7 reports the second-order M-M fits for the same system configuration $M/M^3/2/N$ for different values of $N = \{10, 12, 14, 16\}$.

Table 7. Best-fit parameters for second-order M-M system for M/M^3/2/N system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
M/M^3/2/10	9.47	0.38	97.0	0.374	0.233
M/M^3/2/12	11.62	0.48	98.4	0.355	0.180
M/M^3/2/14	13.82	0.57	99.0	0.342	0.144
M/M^3/2/16	16.04	0.63	99.2	0.378	0.241

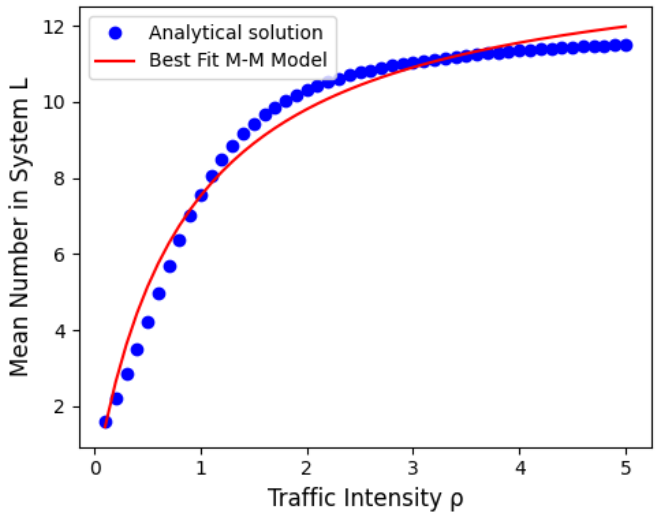


Figure 12. $M/M^3/2/12$ system: Comparison between analytical solution (LU) and first-order M-M model for mean number L.

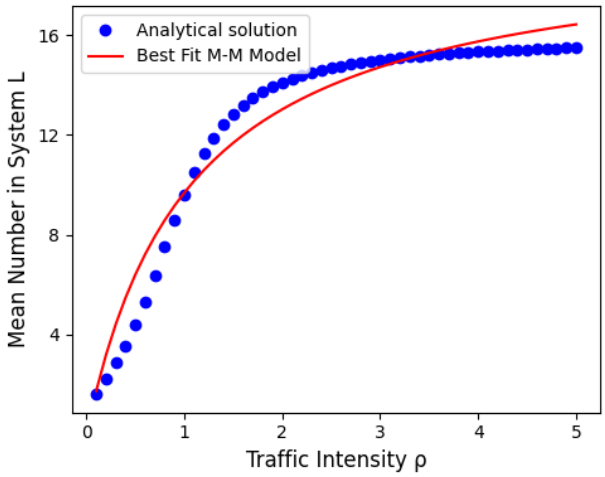


Figure 13. $M/M^3/2/16$ system: Comparison between the analytical solution (LU) and first-order M-M fit.

Figures 14 and 15 compare the analytical solution (LU) and the second-order M-M prediction for $M/M^3/2/12$ and $M/M^3/2/16$ respectively for $b = 3$ and $c = 2$. The two curves are visually indistinguishable throughout the range $\rho\epsilon\{0, 5\}$. As observed in Table 6, the first-order M-M approximation yields $Vmax \approx N$ for the $M/M^b/c/N$ system in all cases except for $N = 16$. The second-order approximation, on the other hand, retains this behaviour uniformly across $N = \{6, 8, 10, 12\}$, as shown in Table 3.

The second-order fits uniformly outperform their first-order counterparts exceptionally for the $M/M^3/2/10$ system, where first-order fits return 98.8% as compared to second-order with 97.1%. R^2 values range from 97.1% to 99.2%, with $b = 3$ yielding the best fit overall, except for $N = 10$. The K_{max} parameter for the second-order form is substantially smaller than for the first-order form. In practical terms, the second-order M-M model provides more exact predictions of L across the full operating range of ρ with only two fitted parameters.

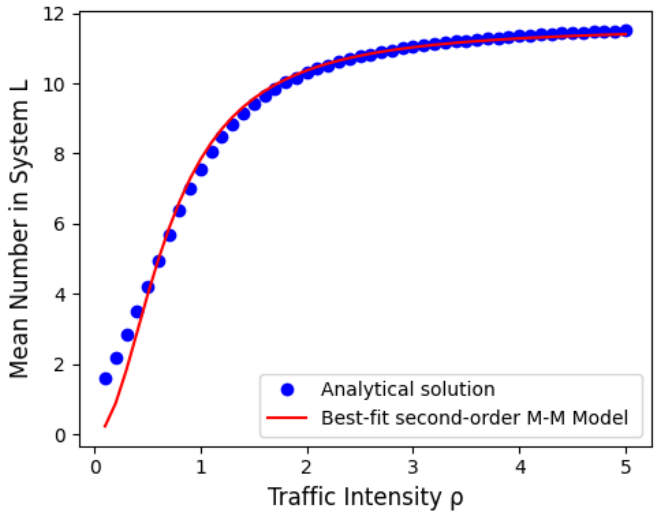


Figure 14. $M/M^3/2/12$ system: Comparison between the analytical solution (LU) and second-order M-M model for mean number.

Table 8 summarises the influence of buffer capacity N on the fitted curve, reporting the corresponding parameter values for both the first-order and second-order fits.

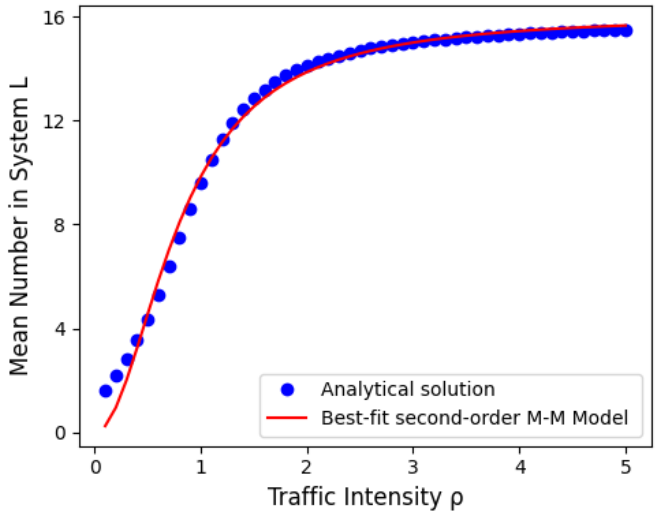


Figure 15. $M/M^3/2/16$ system: Comparison between the analytical solution (LU) and second-order M-M model for mean number L.

Table 8. Mean number in the system L (M-M first-order and second-order) for fixed $b = 3$, $c = 2$, $\mu = 8$ across different values of $N = \{10, 12, 14, 16\}$.

ρ	First-Order				Second-Order			
	N = 10	**N = 12**	**N = 14**	**N = 16**	**N = 10**	**N = 12**	**N = 14**	**N = 16**
0.10	1.3857	1.4775	1.6025	1.7434	0.0875	0.1024	0.1204	0.1401
0.50	4.6258	5.1909	5.8056	6.4456	1.5980	1.9511	2.3403	2.7521
0.90	6.2495	7.2023	8.1934	9.2038	3.1780	4.0657	4.9904	5.9423
1.30	7.2249	8.4636	9.7330	11.0170	4.1264	5.4351	6.7732	8.1342
1.70	7.8756	9.3285	10.8083	12.2999	4.6576	6.2381	7.8443	9.4692
2.10	8.3406	9.9585	11.6017	13.2554	4.9681	6.7200	8.4965	10.2888
2.50	8.6895	10.4378	12.2113	13.9947	5.1607	7.0238	8.9112	10.8127
2.90	8.9610	10.8148	12.6942	14.5837	5.2868	7.2248	9.1872	11.1626
3.30	9.1782	11.1190	13.0864	15.0640	5.3733	7.3637	9.3787	11.4058
3.70	9.3559	11.3696	13.4110	15.4631	5.4350	7.4632	9.5162	11.5808
4.10	9.5041	11.5797	13.6843	15.8001	5.4805	7.5367	9.6180	11.7105
4.50	9.6294	11.7584	13.9175	16.0883	5.5148	7.5924	9.6954	11.8091
4.90	9.7369	11.9122	14.1188	16.3377	5.5414	7.6356	9.7554	11.8857

From Table 8, it can be observed that as the values of N increase, L also increases, considering b = 3, c = 2 with $\mu = 8$ across different values of $N = \{10, 12, 14, 16\}$ for M-M fits for both orders, respectively.

6.2. Effects of Size of Multiple Servers

After establishing the non-linear M-M Model for the queueing model, the analysis proceeds to access how effect of the number of servers to determine the best-fit model for obtaining the plot between the traffic intensity ρ and mean number L for the first-order and second-order variants. Tables 9 and 10 summarise the coefficients V_{max}, K_{max}, and R^2 obtained for the first-order and second-order M-M model, respectively.

Furthermore, Figures 16 and 17 illustrate the curves obtained for the comparison between the analytical solution and best-fit M-M model for $M/M^3/5/30$ and $M/M^3/7/30$, respectively.

In Figures 16 and 17, we can observe that by adding server congestion indicating for the same ρ and higher values of c, the mean number L increases to a certain value of ρ and then decreases as c increases, as expected. According to Table 9, the first-order M-M approximation of the $M/M^3/c/N$ system yields poor saturation with $V_{max} > N$ in all cases.

The second-order approximation, as reported in Table 10, keeps $V_{max} \approx N$ across all values of multiple servers for $c = \{4, 5, 6, 7\}$. The second-order M-M fits for the multi-server case are uniformly excellent, with R^2 ranging from 98.2% to 99.0%.

Table 9. Best-fit parameters for first-order M-M system for M/M^3/c/30 system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
M/M^3/4/30	38.02	1.02	91.7	2.428	2.012
M/M^3/5/30	37.03	0.9	93.1	2.113	1.759
M/M^3/6/30	36.1	0.79	94.2	1.836	1.523
M/M^3/7/30	35.24	0.68	95.1	1.61	1.349

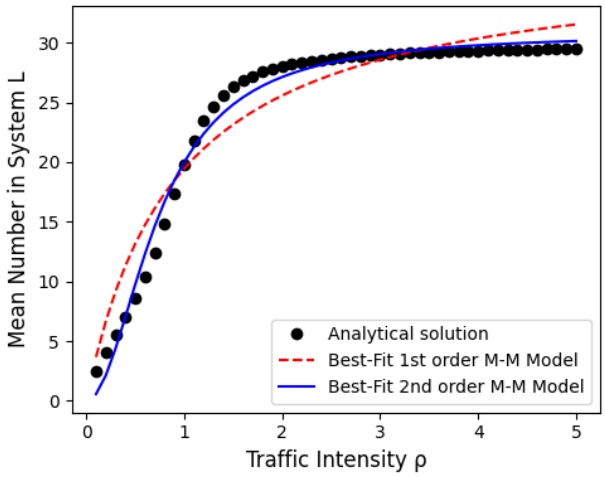


Figure 16. $M/M^3/5/30$ system: Comparison between the analytical solution (LU) and best-fit M-M model.

Table 10. Best-fit parameters for second-order M-M system for M/M^3/c/30 system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
M/M^3/4/30	31.06	0.63	98.1	1.16	0.939
M/M^3/5/30	30.75	0.54	98.6	0.938	0.758
M/M^3/6/30	30.46	0.45	99.0	0.775	0.591
M/M^3/7/30	30.2	0.37	99.2	0.67	0.448

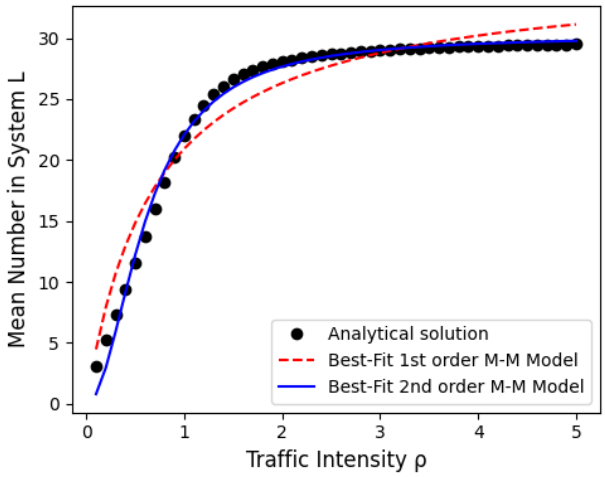


Figure 17. $M/M^3/7/30$ system: Comparison for mean number between the analytical solution (LU) and best-fit M-M model.

Table 11 discusses the effect of multiple servers c, displaying the values of the corresponding parameters for each order M-M fit. Table 11 demonstrates the influence of number of servers for each order M-M regression fit, representing the values obtained for the corresponding parameter values plotted on the fitted curve.

From Table 11, it can be observed that the fitted mean number in the system L increases with c at lower traffic intensities, i.e., up to about $\rho \approx 3.30$, and decreases with c at higher traffic intensities beyond about $\rho = 3.7$ considering fixed value of b = 3, N = 30 with $\mu = 8$ for first- and second-order M-M variants. The effect of the number of servers c is examined at fixed values of the traffic intensity ρ. Since arrival rate λ is scaled in proportion to c for each ρ, the tabulated results reflect the joint effect of number of servers c and λ.

Table 11. Mean number in the system L (M-M first-order and second-order) for fixed $b = 3$, $N = 30$, $\mu = 8$ across different values of $c = 4, 5, 6, 7$.

First-Order								
ρ	$c = 4$		$c = 5$		$c = 6$		$c = 7$	
	λ	L	λ	L	λ	L	λ	L
0.10	9.60	3.3891	12.00	3.7052	14.40	4.0792	16.80	4.5185
0.50	48.00	12.5048	60.00	13.2417	72.00	14.0556	84.00	14.9406
0.90	86.40	17.8347	108.00	18.5452	129.60	19.3003	151.20	20.0889
1.30	124.80	21.3317	156.00	21.9222	187.20	22.5343	218.40	23.1582
1.70	163.20	23.8026	204.00	24.2611	244.80	24.7279	285.60	25.1962
2.10	201.60	25.6412	252.00	25.9768	302.40	26.3136	352.80	26.6480
2.50	240.00	27.0627	300.00	27.2890	360.00	27.5133	420.00	27.7346
2.90	278.40	28.1945	348.00	28.3252	417.60	28.4527	487.20	28.5785
3.30	316.80	29.1171	396.00	29.1641	475.20	29.2082	554.40	29.2528
3.70	355.20	29.8835	444.00	29.8572	532.80	29.8290	621.60	29.8040
4.10	393.60	30.5303	492.00	30.4394	590.40	30.3481	688.80	30.2629
4.50	432.00	31.0834	540.00	30.9355	648.00	30.7887	756.00	30.6510
4.90	470.40	31.5619	588.00	31.3631	705.60	31.1674	823.20	30.9834
Second-Order								
ρ	$c = 4$		$c = 5$		$c = 6$		$c = 7$	
	λ	L	λ	L	λ	L	λ	L
0.10	9.60	0.4805	12.00	0.5564	14.40	0.6552	16.80	0.7836
0.50	48.00	8.7640	60.00	9.7029	72.00	10.8088	84.00	12.0800
0.90	86.40	17.4084	108.00	18.4316	129.60	19.5263	151.20	20.6595
1.30	124.80	22.5873	156.00	23.3051	187.20	24.0318	218.40	24.7448
1.70	163.20	25.4850	204.00	25.9250	244.80	26.3561	285.60	26.7679
2.10	201.60	27.1774	252.00	27.4214	302.40	27.6545	352.80	27.8741
2.50	240.00	28.2266	300.00	28.3369	360.00	28.4386	420.00	28.5341
2.90	278.40	28.9138	348.00	28.9315	417.60	28.9438	487.20	28.9562
3.30	316.80	29.3852	396.00	29.3371	475.20	29.2867	554.40	29.2412
3.70	355.20	29.7212	444.00	29.6252	532.80	29.5293	621.60	29.4423
4.10	393.60	29.9686	492.00	29.8367	590.40	29.7069	688.80	29.5891
4.50	432.00	30.1558	540.00	29.9963	648.00	29.8407	756.00	29.6995
4.90	470.40	30.3005	588.00	30.1197	705.60	29.9439	823.20	29.7845

6.3. Effects of Batch-Size b

In this section, analysis is performed for different values of $b = \{4, 5, 6, 7\}$ for a fixed value of c = 4 and N = 30. Tables 12 and 13 represent the values of V_{max}, K_{max} and R^2 obtained by plotting the graph against the mean number L in the system and traffic intensity ρ for the first-order and second-order M-M model, respectively.

Furthermore, Figures 18 and 19 illustrate the plot obtained for the comparison between the analytical solution and the best-fit M-M model for the queueing systems $M/M^4/4/30$ and $M/M^5/4/30$, respectively.

Table 12. Best-fit parameters for first-order M-M system for $M/M^b/4/30$ system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
$M/M^4/4/30$	36.21	0.84	94.9	1.729	1.438
$M/M^5/4/30$	34.58	0.67	96.9	1.219	1.015
$M/M^6/4/30$	33.24	0.53	98.0	0.889	0.731
$M/M^7/4/30$	32.3	0.42	98.5	0.711	0.574

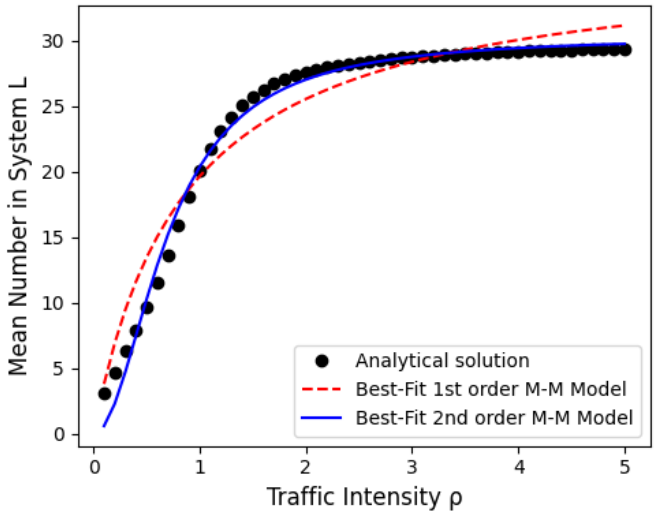


Figure 18. $M/M^4/4/30$ system: Comparison between the analytical solution (LU) and best-fit M-M model.

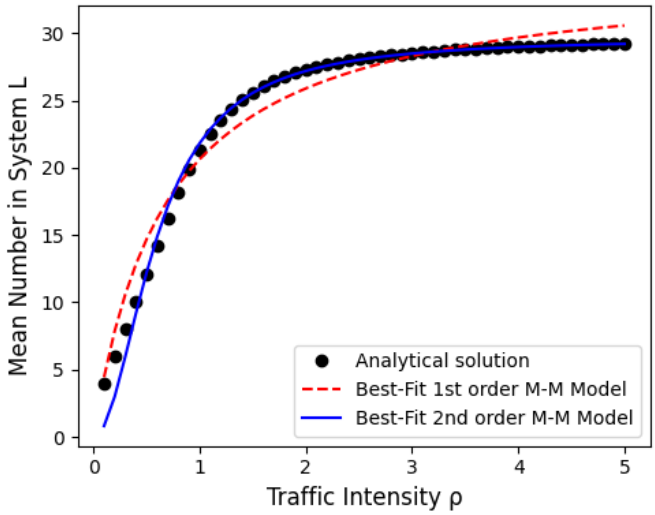


Figure 19. $M/M^5/4/30$ system: Comparison between the analytical solution (LU) and best-fit M-M model for mean number.

From Tables 12 and 13, we can observe that the second-order M-M model fits better as compared to the first-order M-M model, excluding the $M/M^4/4/30$ and $M/M^5/4/30$ systems. The second-order M-M fits for variable batch-size for the multi-server are uniformly excellent, with R^2 ranging from 97.2% to 99.0%. The first-order M-M model attains $Vmax > N$ for all values of b exploiting the saturation behaviour $V_{max} \approx N$ considered in Table 12, whereas the second-order model, as seen in Table 13, maintains $Vmax \approx N$ for all the values of $b = \{4, 5, 6, 7\}$.

Table 13. Best-fit parameters for second-order M-M system for $M/M^b/4/30$ system.

	V_{max}	K_{max}	R^2(%)	RMSE	MAE
$M/M^4/4/30$	30.3	0.49	99.0	0.772	0.54
$M/M^5/4/30$	29.63	0.36	98.7	0.785	0.323
$M/M^6/4/30$	29.11	0.25	97.9	0.917	0.482
$M/M^7/4/30$	28.84	0.18	97.0	1.012	0.582

Table 14 summarises the influence of batch size for each order M-M regression fit, representing the values obtained for the corresponding parameter values plotted on the fitted curve.

Table 14. Mean number in the system L (M-M first-order and second-order) for fixed $c = 4$, $N = 30$, $\mu = 8$ across different values of $b = \{4, 5, 6, 7\}$.

First-Order								
ρ	$b = 4$		$b = 5$		$b = 6$		$b = 7$	
	λ	L	λ	L	λ	L	λ	L
0.10	12.80	3.8619	16.00	4.5034	19.20	5.3246	22.40	6.2968
0.50	64.00	13.5433	80.00	14.8038	96.00	16.2152	112.00	17.6710
0.90	115.20	18.7722	144.00	19.8479	172.80	20.9839	201.60	22.1082
1.30	166.40	22.0460	208.00	22.8413	249.60	23.6601	291.20	24.4717
1.70	217.60	24.2884	272.00	24.8232	326.40	25.3734	380.80	25.9398
2.10	268.80	25.9205	336.00	26.2323	403.20	26.5641	470.40	26.9403
2.50	320.00	27.1617	400.00	27.2855	480.00	27.4397	560.00	27.6659
2.90	371.20	28.1373	464.00	28.1026	556.80	28.1107	649.60	28.2162
3.30	422.40	28.9244	528.00	28.7549	633.60	28.6413	739.20	28.6479
3.70	473.60	29.5728	592.00	29.2877	710.40	29.0714	828.80	28.9956
4.10	524.80	30.1162	656.00	29.7311	787.20	29.4271	918.40	29.2817
4.50	576.00	30.5781	720.00	30.1059	864.00	29.7261	1008.00	29.5211
4.90	627.20	30.9757	784.00	30.4269	940.80	29.9810	1097.60	29.7245
Second-Order								
ρ	$b = 4$		$b = 5$		$b = 6$		$b = 7$	
	λ	L	λ	L	λ	L	λ	L
0.10	12.80	0.6004	16.00	0.7871	19.20	1.0655	22.40	1.4649
0.50	64.00	10.1783	80.00	12.0234	96.00	14.1925	112.00	16.5135
0.90	115.20	18.8312	144.00	20.4220	172.80	22.0001	201.60	23.4546
1.30	166.40	23.4692	208.00	24.3809	249.60	25.2261	291.20	25.9944
1.70	217.60	25.9080	272.00	26.3318	326.40	26.7238	380.80	27.1167
2.10	268.80	27.2843	336.00	27.3959	403.20	27.5168	470.40	27.6972
2.50	320.00	28.1204	400.00	28.0300	480.00	27.9816	560.00	28.0331
2.90	371.20	28.6611	464.00	28.4351	556.80	28.2755	649.60	28.2439
3.30	422.40	29.0289	528.00	28.7085	633.60	28.4727	739.20	28.3846
3.70	473.60	29.2896	592.00	28.9012	710.40	28.6110	828.80	28.4830
4.10	524.80	29.4807	656.00	29.0420	787.20	28.7117	918.40	28.5545
4.50	576.00	29.6248	720.00	29.1478	864.00	28.7872	1008.00	28.6079
4.90	627.20	29.7360	784.00	29.2293	940.80	28.8453	1097.60	28.6490

It is clear that the mean number L decreases as b increases up to about $\rho \approx 2.50$, and then L increases as b increases beyond about $\rho \approx 2.90$ across various values of $b = 4, 5, 6, 7$, c = 4, N = 30 with $\mu = 8$, which can be seen from Table 14. We investigate how the batch size b affects system performance while keeping the traffic intensity ρ constant. For every fixed ρ, the arrival rate λ is adjusted with c, so the values in Table 14 represent the combined influence of b and λ.

7. Discrete-Event Simulation for Validating Data

To validate the analytical framework developed, a discrete-event simulation of bulk service systems was constructed to serve as an independent source of queue-length data, distinct from the exact balance-equation solutions employed for training in Section 4. The simulator generates customer arrivals according to a Poisson process with rate λ and admits them into the system only while the buffer capacity remains below the capacity N, with any arrival encountering a full system being discarded. Service is allocated in fixed batches of size b: whenever a free server observes at least b customers waiting, it draws exactly b of them into service and processes the batch as a single unit following an exponential distribution with rate μ, after which all b customers depart simultaneously, and the server becomes free to form a new batch. For the multi-server, up to c such servers operate independently, each following the same batch-formation rule, so that the number of active servers at any instant is governed by the current queue population in the manner described in Section 4. Each simulation run is initialised in the empty state and allowed to progress through a warm-up interval of length value 1000 time units, discarded prior to data collection, so that the time-averaged queue length is measured only once the system has settled into statistical steady state. The warm-up length was fixed at 1000 time units, and the same warm-up was applied to every parameter configuration. With $\mu = 8$, the mean service time is 0.125, so the warm-up spans about 8000 mean service times, which is far longer than the time needed for the queue to settle from the empty initial state. For every parameter configuration (b, c, N, ρ), a batch of independent replications is executed with distinct random-number streams, and the resulting sample means of the time-averaged number in the system are aggregated to obtain a point estimate of L together with a 95% confidence interval, computed from the replication variance. To assess the reliability of the simulation results, a 95% confidence interval was constructed using multiple independent simulation replications. If $\hat{L}_R$ denotes the time-averaged number in the system over the post-warm-up horizon of replication R $(R = 1, \ldots, m)$, the point estimate is $\bar{L} = \frac{1}{m}\sum_{R=1}^{m} \hat{L}_R$, the sample variance is $s^2 = \frac{1}{m-1}\sum_{R=1}^{m}(\hat{L}_R - \bar{L})^2$, and the 95% confidence interval is $\bar{L} \pm 2.093 \cdot \frac{s}{\sqrt{R}}$, with R independent replications. The blocking probability is estimated as the proportion of arrivals discarded in each replication, and its confidence interval is constructed in the same way as illustrated in the Figure 20.

7.1. Simulation Verification for $M/M^b/c/N$ Model

Table 15 presents the outcome of simulation verification for the configuration $b = 3, N = 8$, with traffic intensity ρ sampled from 0.1 through 3.0 in steps of 0.4, covering the system's behaviour from lightly loaded to moderately congested conditions. At the lowest traffic intensity, the analytical and simulated values of L differ by less than 0.0024, and this close correspondence is maintained as ρ increases, with the largest observed gap across the sampled range remaining under 0.0038 in absolute terms. The blocking probability follows the same pattern. It rises gradually from an effectively negligible value at $\rho = 0.10$ to just over 0.0005 at $\rho = 2.90$, with the simulated estimate matching the analytical figure to within the reported confidence bound at every sampled point. Every row in the table is flagged as a match, meaning the analytically computed value of L falls inside the 95% confidence interval produced by the 20 independent simulation replications, each run over a horizon of 20,000 time units following a warm-up period of 1000 time units sufficient to eliminate initial transient bias. Because the discrete-event simulation was constructed independently of the balance-equation solver, this consistent agreement across the full sampled range of ρ provides direct evidence that the exact analytical formulation of the $M/M^b/c/N$ system used elsewhere in this paper correctly reproduces the true stationary

behaviour of the queue, rather than being validated only indirectly through the fit quality of the M-M model.

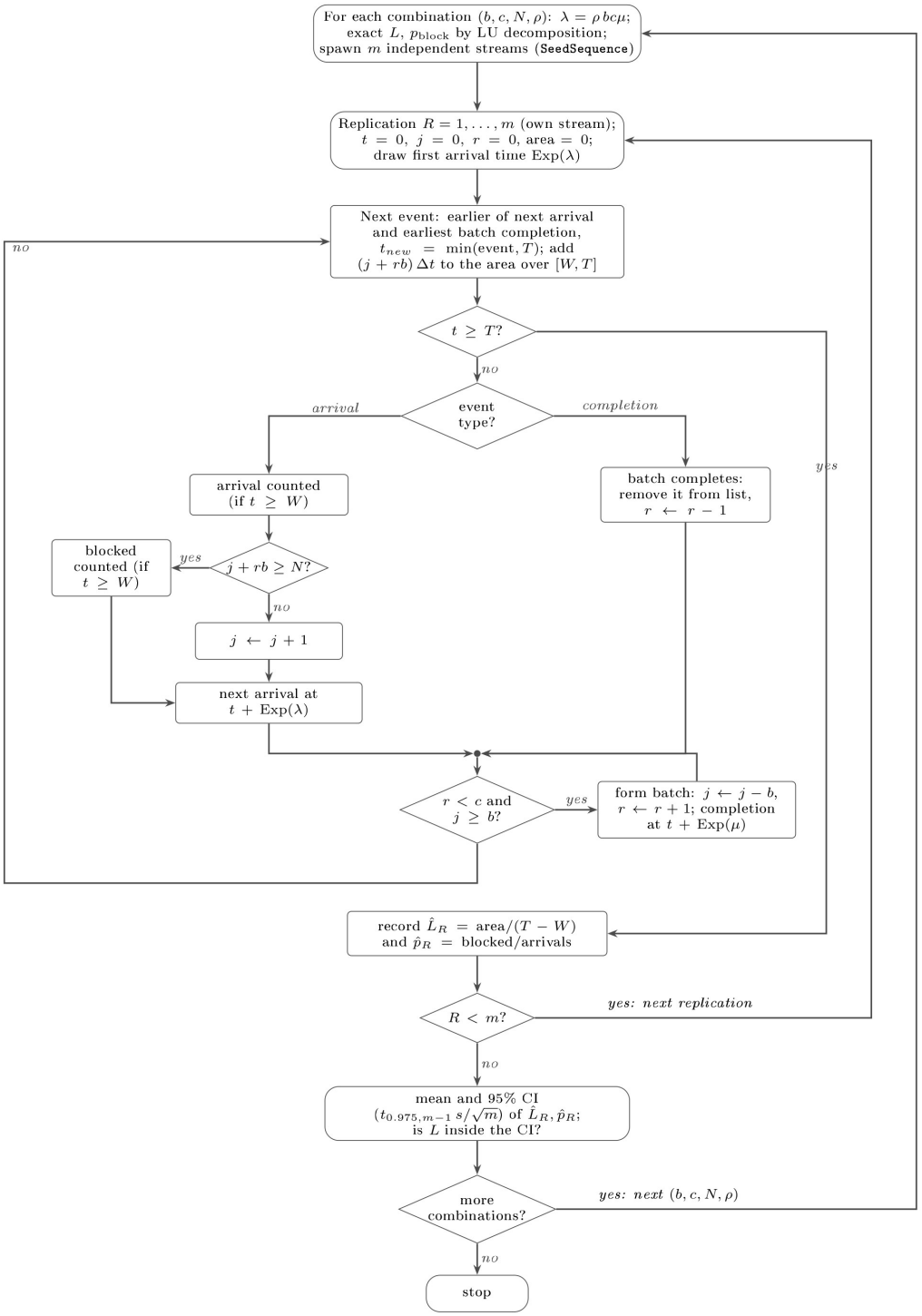


Figure 20. Flowchart of the discrete-event simulation used to validate the analytical results. For each combination (b, c, N, ρ), m independent replications are run, and the simulated L and p_{block} are compared with the exact LU solution.

Table 15. M/M^3/2/8: Analytical (LU) versus simulated mean number in system and blocking probability comparison for different values of ρ with 20 replications.

ρ	λ	L	$\hat{L}_R \pm 95\%$CI	p_{block}	$\hat{p}_R \pm 95\%$CI
0.10	1.600	1.2000	1.2012 ± 0.0030	0.0000	0.0000 ± 0.0000
0.50	8.000	2.0000	2.0025 ± 0.0026	0.0025	0.0024 ± 0.0001
0.90	14.400	2.7840	2.7833 ± 0.0031	0.0199	0.0198 ± 0.0002
1.30	20.800	3.5079	3.5084 ± 0.0038	0.0576	0.0577 ± 0.0003
1.70	27.200	4.1378	4.1377 ± 0.0035	0.1093	0.1093 ± 0.0005
2.10	33.600	4.6653	4.6655 ± 0.0036	0.1671	0.1671 ± 0.0004
2.50	40.000	5.0992	5.0985 ± 0.0024	0.2252	0.2250 ± 0.0004
2.90	46.400	5.4540	5.4534 ± 0.0029	0.2804	0.2804 ± 0.0005

7.2. Simulation Validation over Combinations of b, c, N and ρ

To check that the agreement in Table 16 does not depend on a single configuration, the simulation was repeated for four (b, c, N) combinations at $\rho \in \{0.5, 1.0, 1.5\}$, with $\mu = 8$ and $\lambda = \rho\, bc\mu$. For every combination, $m = 20$ independent replications were run, each with a horizon of $T = 2000$ time units after a warm-up of $W = 100$ time units. Table 16 compares the simulated mean number in the system and blocking probability with the exact solution of Section 4.2.

The exact value of L lies inside the 95% confidence interval in all 12 cases. Over all 12 cases, the largest relative difference in L is below 0.33%, and the largest difference in the blocking probability is 0.0013. The simulation therefore confirms the analytical solution across different batch sizes, numbers of servers, capacities, and traffic intensities.

Table 16. Analytical (LU) versus simulated L and p_{block} for combinations of b, c, N, and ρ ($\mu = 8$, 20 replications, simulated values given as mean ± 95% CI half-width).

(b, c, N)	ρ	λ	L	$\hat{L}_R \pm$CI	p_{block}	$\hat{p}_R \pm$CI
(3,2,8)	0.5	24.0	3.8356	3.8370 ± 0.0139	0.0822	0.0818 ± 0.0012
(3,2,8)	1.0	48.0	5.5322	5.5260 ± 0.0114	0.2936	0.2931 ± 0.0015
(3,2,8)	1.5	72.0	6.3525	6.3523 ± 0.0059	0.4559	0.4556 ± 0.0012
(3,2,12)	0.5	24.0	4.2244	4.2385 ± 0.0246	0.0174	0.0176 ± 0.0006
(3,2,12)	1.0	48.0	7.5701	7.5703 ± 0.0251	0.1832	0.1837 ± 0.0020
(3,2,12)	1.5	72.0	9.4240	9.4274 ± 0.0153	0.3783	0.3781 ± 0.0016
(3,4,30)	0.5	48.0	7.1720	7.1639 ± 0.0238	0.0001	0.0001 ± 0.0000
(3,4,30)	1.0	96.0	18.6858	18.6915 ± 0.1169	0.0811	0.0815 ± 0.0023
(3,4,30)	1.5	144.0	26.2034	26.2167 ± 0.0232	0.3362	0.3363 ± 0.0011
(5,4,30)	0.5	80.0	12.0932	12.0809 ± 0.0343	0.0047	0.0045 ± 0.0002
(5,4,30)	1.0	160.0	21.3192	21.3279 ± 0.0238	0.1493	0.1495 ± 0.0011
(5,4,30)	1.5	240.0	25.6033	25.6210 ± 0.0223	0.3624	0.3637 ± 0.0014

8. Validation of M-M Approximation

To examine the approximation itself, the fitted curves were therefore evaluated at traffic intensities that took no part in the estimation. The parameters $V_{\max}$ and $K_{\max}$ were estimated by nonlinear least squares from the exact mean number in the system at $m = 25$ equally spaced values $\rho_1, \ldots, \rho_{25}$ in [0.1, 5.0]. The validation set consists of the $m_v = 24$ midpoints $\rho_i^* = (\rho_i + \rho_{i+1})/2$, $i = 1, \ldots, 24$, which range from 0.202 to 4.898, so no validation point coincides with a fitting point. At each ρ_i^*, the exact value $L(\rho_i^*)$ was obtained from the analytical solution, and the fitted curve $\hat{L}(\rho_j^*)$ was evaluated with the parameters estimated

on the fitting grid, without re-estimation. The agreement is measured by the root mean square error and the mean absolute error, defined by

$$\mathrm{RMSE} = \sqrt{\frac{1}{m_v}\sum_{i=1}^{m_v}\left(L(\rho_i^*) - \hat{L}(\rho_i^*)\right)^2}, \qquad \mathrm{MAE} = \frac{1}{m_v}\sum_{i=1}^{m_v}\left|L(\rho_i^*) - \hat{L}(\rho_i^*)\right|, \tag{38}$$

both expressed in customers. The RMSE gives more weight to large deviations than the MAE, so reporting both shows whether the error is spread evenly over the range of ρ or concentrated at a few values. The same two measures were computed on the fitting grid for comparison.

Table 17 summarises the results. For the first-order model, the validation RMSE lies between 0.71 and 2.42 customers and the MAE between 0.57 and 2.01. Relative to the capacity $N = 30$, the RMSE corresponds to 2.5 to 6.7 %. In every case, the validation errors exceed the corresponding fitting errors by less than 8.5 %, so the accuracy of the fit does not depend on the particular grid used for estimation. The errors decrease as the number of servers c or the batch size b increases.

For the second-order model, the validation RMSE lies between 0.19 and 1.16 customers (0.6 to 3.9 percent of N) and the MAE between 0.28 and 0.90, and in none of the eight cases do the validation errors exceed the fitting errors. The second-order form gives the smaller validation RMSE in seven of the eight cases. The exception is $M/M^7/4/30$, where the first-order model has the lower RMSE (0.742 against 0.803), although the MAE still favours the second-order model (0.496 against 0.600). The advantage of the second-order model is uniform in all cases except for $M/M^7/4/30$(RMSE). Within the fitted range, the M-M curves reproduce L at unseen traffic intensities with errors of the same size as on the fitting grid. The mean absolute error is between 0.5 and 2.0 customers for the first-order form and below one customer for the second-order form, against a capacity of 30 customers.

Table 17. Fitting and validation errors (in customers) of the first- and second-order M-M approximations for $N = 30$ and $\mu = 8$. The validation errors are computed at 24 values of ρ that were not used for fitting.

	First-Order				Second-Order			
	Fitting		Validation		Fitting		Validation	
Queue	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
$M/M^3/4/30$	2.428	2.012	2.462	2.029	1.160	0.939	1.156	0.905
$M/M^3/5/30$	2.113	1.759	2.143	1.787	0.938	0.758	0.921	0.707
$M/M^3/6/30$	1.836	1.523	1.862	1.564	0.775	0.591	0.742	0.535
$M/M^3/7/30$	1.610	1.349	1.634	1.363	0.670	0.448	0.19	0.393
$M/M^4/4/30$	1.729	1.438	1.754	1.473	0.772	0.540	0.715	0.492
$M/M^5/4/30$	1.219	1.015	1.240	1.041	0.785	0.323	0.676	0.279
$M/M^6/4/30$	0.889	0.731	0.912	0.755	0.917	0.482	0.766	0.424
$M/M^7/4/30$	0.711	0.574	0.742	0.600	1.012	0.582	0.803	0.496

9. Computational Efficiency

Table 18 compares the average runtime for two approaches for computing L at a range of equally spaced traffic intensities varying from $0.1 \leq \rho \leq 5.0$, i.e., 25 points for four $M/M^b/c/K$ system of increasing size. The approaches are (a) generating training data via LU decomposition (25 LU solves) in (s) and (b) fitting the curve via M-M model (s). As reported in Table 18, all timings were obtained from Python implementations on a standard laptop (Intel Core i3) and averaged over 10 replications.

Table 18. Computational runtime comparison for 25 values of ρ.

System(b, c, N)	LU Decomposition (s)	M-M Prediction (s)
(6, 4, 30)	0.056	0.00039
(30, 9, 300)	2.505	0.00020
(40, 11, 500)	7.197	0.00017
(50, 18, 1000)	31.76	0.00013

Table 18 compares the computation time of LU decomposition and the M-M prediction for four $M/M^b/c/N$ systems of increasing size computing L at a range of equally spaced traffic intensities varying from $0.1 \leq \rho \leq 5.0$, i.e., 25 points. The cost of LU decomposition grows steeply with the system dimension, from 0.056 s for $(6, 4, 30)$ to 31.76 s for $(50, 18, 1000)$, because the size of the linear system to be solved increases with N. The M-M prediction is faster than LU decomposition, and its runtime stays on the order of 10^{-4} s for every system and does not increase with size, which makes it roughly four to five orders of magnitude faster than LU decomposition for the largest system. These results show that once the M-M model has been fitted, it gives near-instantaneous estimates of L, and its advantage over LU decomposition methods grows as the system becomes larger.

10. Practical Applications

In this section, we will discuss the cost function related to the $M/M^b/c/N$ model. In real-world scenarios, $M/M^b/c/N$ systems are applicable to airport shuttle services, amusement park rides, elevator systems in buildings, bank/ATM cash replenishment, manufacturing line batching, cargo loading at ports, etc., with batch size b having multiple servers c and buffer capacity N. Consider a scenario of an airport shuttle service operating between a terminal and transit hub, where passengers arrive individually but are transported in batches. Each shuttle van departs only once it has collected a fixed number of passengers b, indicating the fixed seating capacity per vehicle. The service is provided by c identical shuttle vans operating in parallel, each capable of carrying one full batch of b passengers per trip. Given the constrained seating or standing capacity of the pickup zone, only a finite number of passengers, N, can be accommodated in the queue at any given time. Passengers arriving when the waiting area is full are unable to join the queue and are consequently lost to the system, corresponding to blocking in the $M/M^b/c/N$ framework.

Cost Function

The economic performance of the batch-service queueing system is evaluated through a composite cost function that aggregates every recurring expense generated by the operating policy. Two decision variables that govern the system are batch size b, which determines how many customers are grouped before a service cycle begins, and c, the number of parallel servers allocated to process the arriving traffic [30]. The total cost rate is defined as:

$$T(C) = \underbrace{C_{s0}\, c}_{\text{initial cost}} + \underbrace{C_v\, b\, c}_{\text{capacity cost}} + \underbrace{C_h\, L_q}_{\text{holding cost}} + \underbrace{C_{s1}\, L_s}_{\text{service cost}} + \underbrace{C_d\, \frac{\lambda_{eff}}{b}}_{\text{dispatch cost}} + \underbrace{C_\ell\, \lambda\, p_{block}}_{\text{blocking cost}} \tag{39}$$

where the cost parameters and their numerical values used in the experiments are summarized in Table 19.

Table 19. Cost parameters and their numerical values used in the numerical experiments.

Term	Definitions	Value
First	initial cost per server per unit time, independent of server's batch capacity	$C_{s0} = 10$
Second	cost of provisioning c servers each of capacity b	$C_v = 4$
Third	holding cost for customers waiting in the queue	$C_h = 5$
Fourth	service cost while customers are actually serving	$C_{s1} = 4$
Fifth	fixed cost incurred every time a batch is dispatched	$C_d \frac{\lambda_{eff}}{b} = 10$
Sixth	penalty incurred for each customer, using the blocking probability	$C_l = 3$

Since L_q, L_s, and P_{block} are all obtained from the same finite-capacity batch-service queueing model, the six cost terms are dependent on each other. A change in b or c simultaneously reshapes the congestion, the blocking, and the dispatch frequency, which is precisely why T(C) must be evaluated jointly rather than by optimising each cost component in isolation. In the numerical study that follows, this cost function is evaluated over an integer grid of batch sizes and server counts to locate the configuration that minimises total expected cost. The location of the optimum depends on the relative magnitudes of these coefficients mentioned in the above table, so the operational conclusions below should be read as specific to this cost structure.

The cost coefficients used to generate Figures 21–26 are $C_{s0} = 10$, $C_v = 4$, $C_h = 5$, $C_{s1} = 4$, $C_d \frac{\lambda_{eff}}{b} = 10$, and $C_l = 3$, per unit time. These values are held fixed in all numerical experiments reported in this Section; hence, every result can be reproduced from the stated parameters.

Figure 21 represents the graph of the total system cost T(C) against batch size b for $c = 5, 7, 9$ servers under a traffic load of $\lambda = 100, \mu = 3$ and a buffer capacity of $N = 90$. Each curve first falls, reaches a minimum, and then rises again, since small batches raise waiting and dispatch costs while overly large batches raise capacity and service costs. Overall cost also rises with c at every b, reflecting the fixed cost of server provisioning. Thus, batch size and server count together determine cost efficiency and should be optimised jointly. Figure 22 shows T(C) against the number of servers c for batch sizes $b = 5, 7, 8$, with all three curves following a U-shaped pattern. Cost falls sharply at low c due to congestion and blocking from insufficient capacity, then rises as added servers bring provisioning costs that outweigh the benefit of shorter waits. The optimal server count drops from c = 8 at b = 5 to c = 5 at b = 8, showing that larger batches reduce dependence on extra servers. The flatter minimum for larger batch sizes also suggests the cost becomes less sensitive to the exact server count once batching is used effectively.

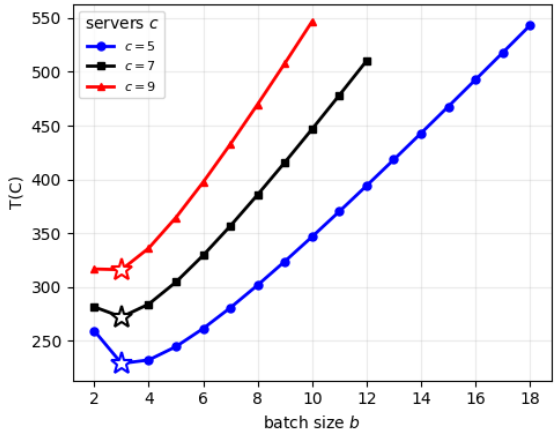


Figure 21. T(C) versus batch size b for different values of c = 5,7,9.

Figure 23 shows T(C) against system capacity N for a fixed server count, comparing batch sizes $b = 5, 6, 7$. Cost rises with N for all cases, but b = 5 climbs almost linearly throughout, while b = 6 and b = 7 rise sharply at first and then flatten once N exceeds about 80–90, as extra capacity yields little further reduction in blocking beyond that point. The optimum values show the cost-minimising N occurs at comparatively low capacity for each batch size, indicating that over-provisioning capacity adds cost without real benefit. Larger batch sizes also achieve lower overall cost at high N, confirming that suitable batching reduces the burden on queue capacity. Figure 24 shows T(C) against system capacity N for a fixed batch size, comparing server counts c = 4, 5, 6. Cost rises with N in all cases, with c = 5 and c = 6 flattening beyond N = 100, while c = 4 keeps climbing steadily due to greater congestion from fewer servers as capacity grows. The optimal N occurs at relatively low capacity for all three, shifting slightly higher as the server count decreases. This confirms that systems with fewer servers need a larger buffer to limit blocking, while more servers allow efficient operation even at modest capacity.

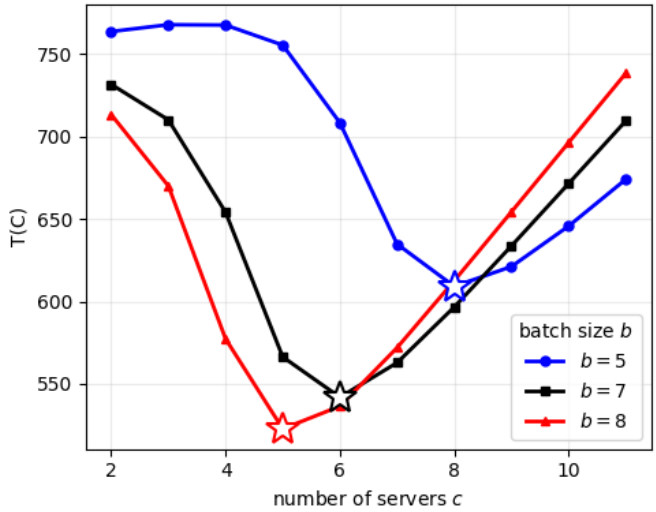


Figure 22. T(C) versus servers c for different values of b = 5,7,8 $\lambda = 100, \mu = 3, N = 90$).

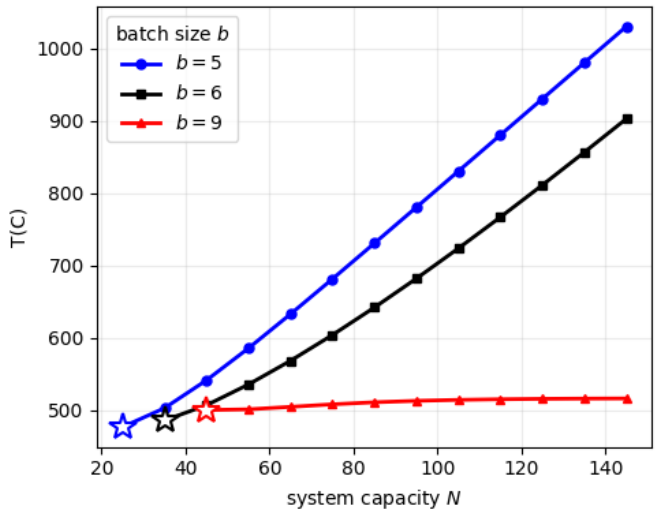


Figure 23. T(C) versus server N for different values of b = 5,6,9.

Figure 25 shows T(C) against traffic intensity ρ for batch sizes $b = 5, 7, 9$, with cost rising steadily in all cases as higher arrival pressure increases congestion and blocking. The

b = 5 curve stays lowest throughout, while b = 9 gives the highest cost, and the gap between the curves widens noticeably at higher ρ. This shows smaller batches handle heavy traffic more cost-effectively, dispatching more frequently in response to demand. The star markers confirm the most cost-efficient region lies at lighter traffic levels for all batch sizes, so the batch size should be chosen based on the expected traffic regime. Figure 26 shows T(C) against traffic intensity ρ for server counts $c = 3, 5, 7$ with cost rising steadily in all cases as higher arrival pressure increases congestion and blocking. The c = 3 curve stays lowest throughout, while c = 7 gives the highest cost, and the gap between the curves widens noticeably at higher ρ. This shows that fewer servers remain more cost-effective as traffic builds, since the fixed provisioning cost of extra servers outweighs their benefit under this load range. The optimal confirms the most cost-efficient region lies at lighter traffic levels for all server counts, so the number of servers should be chosen based on the expected traffic regime.

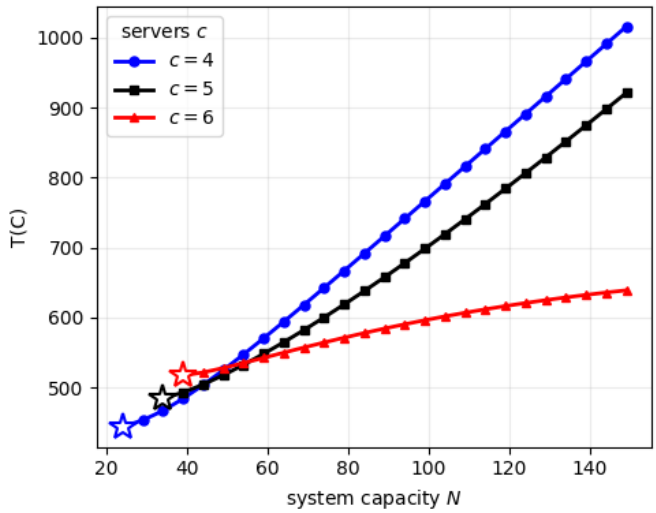


Figure 24. T(C) versus server N for different values of c = 4,5,6.

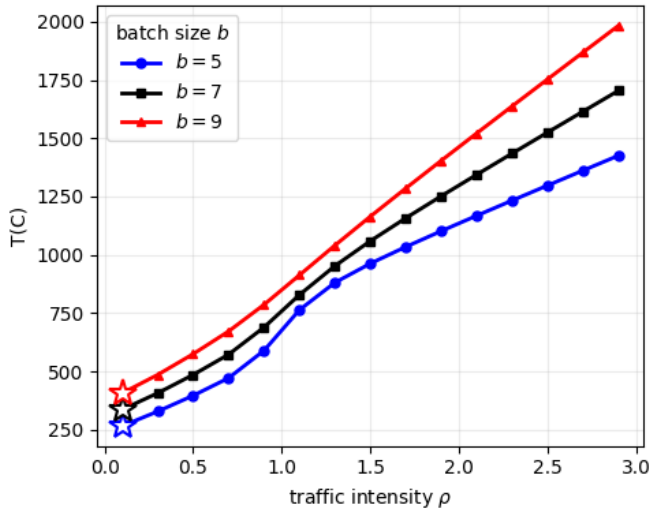


Figure 25. T(C) versus servers ρ for different values of b = 5,7,9.

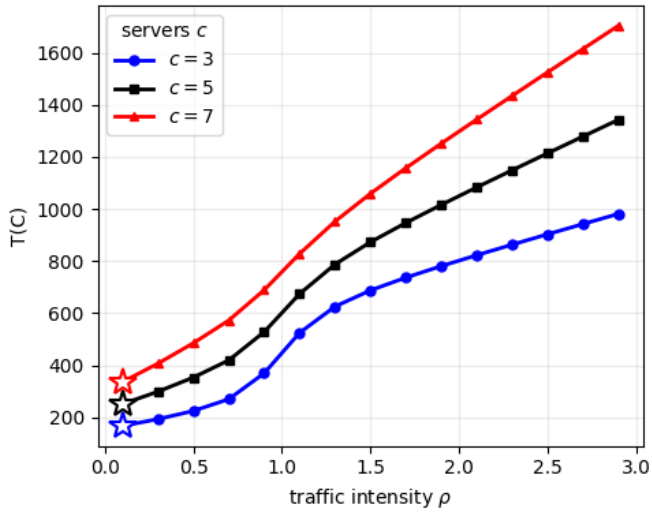


Figure 26. T(C) versus servers ρ for different values of c = 3,5,7.

11. Conclusions

This study presented a comprehensive numerical and computational analysis of a finite-capacity bulk-service queueing system operating under the $M/M^b/1/N$ and $M/M^b/c/N$ framework. By formulating the system as a finite, continuous-time Markov chain and constructing its block-structured generator matrix, the steady-state joint probability distribution was obtained through LU decomposition. A range of performance measures, including the expected queue length, expected number in the system, blocking probability, and effective arrival rate, were derived directly from the stationary distribution. The numerical results demonstrated that system performance is highly sensitive to the choice of a batch size b and server capacity c. For $b = 10$, $c = 7$, $N = 90$, $\lambda = 50$, and $\mu = 1$, the joint distribution shows that four to six servers are busy for 63% of the time, that $L = 54.35$ with $L_q = 5.35$, and that 2% of the arrivals are blocked ($p_{block} = 0.0199$). The sensitivity study showed that at light traffic, L is governed by batch formation and increases with b; that at heavy traffic, it saturates at 88–97% of the capacity N; and that the effect of N and c is strongest near $\rho \approx 1$. On the computational side, the LU solution grows from 0.056 s for $(b, c, N) = (6, 4, 30)$ to 31.76 s for $(50, 18, 1000)$, while the fitted Michaelis–Menten model needs values of the order of 10^{-4} s, and the simulation replications agreed with the exact solution within the 95% confidence intervals.

A nonlinear regression approach based on the Michaelis–Menten function, fitted using the Levenberg–Markquardt algorithm, was employed to characterise the relationship between traffic intensity and mean number in the system. The resulting fits attained a high degree of accuracy, as reflected in the reported goodness of fit statistics. It concludes that the Michaelis–Menten form captures the saturating behaviour typical of queueing performance measures as the system approaches heavy-traffic conditions. Furthermore, the second-order M-M model fits better compared to the first-order fit. A limitation is that $V_{\max}$ was not constrained to be at most N, so the M-M fits can exceed the capacity, and the fitted curves should be used within the fitted range of ρ. Computationally, the M–M model prediction is faster than LU decomposition, and the advantage grows with system size. Taken together, the data generation, regression, validation, and prediction steps form a supervised machine learning algorithm (Algorithm 1) for analysing and simulating the waiting behaviour of bulk-service systems.

Overall, the findings of this study offer practical insights for the design and operation of bulk-service systems in which servers process customers in fixed-size groups and buffer

capacity is inherently limited, making it relevant to numerous real-world applications, including public transportation, batch manufacturing, and dynamic resource allocation. The proposed framework provides a systematic means of balancing service efficiency for decision-makers against operational cost, while the regression-based approximation offers a computationally efficient alternative to repeated exact numerical solutions for large-scale parameter studies. Future work may extend this framework to incorporate heterogeneous servers, server vacations or breakdowns, impatient customers with reneging behaviour, or a hybrid queueing discipline that relaxes the fixed batch-size assumption. Applying alternative machine learning techniques for parameter estimation also represents a promising direction for further research.

Author Contributions: Conceptualisation, P.G.J. and G.K.G.; methodology, P.G.J.; software, P.G.J.; validation, P.G.J. and G.K.G.; formal analysis, P.G.J., G.K.G., M.L.S. and S.V.; investigation, P.G.J. and G.K.G.; resources, P.G.J.; data curation, P.G.J.; writing—original draft, P.G.J.; writing—review and editing, P.G.J.; visualisation, G.K.G.; supervision, G.K.G.; project administration, G.K.G.; funding acquisition, M.L.S. and S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data supporting the findings of this study are available within the paper. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Efrosinin, D.; Vishnevsky, V.; Stepanova, N.; Sztrik, J. Use cases of machine learning in queueing theory based on a GI/G/K system. *Mathematics* **2025**, *13*, 776. [CrossRef]
2. Hochendoner, P.; Ogle, C.; Mather, W.H. A Queueing Approach to Multi-Site Enzyme Kinetics. *Interface Focus* **2014**, *4*, 20130077. [CrossRef] [PubMed]
3. Sule, A.A.; Ghazaleh, H.A. Machine Learning Tool for Analyzing Finite Buffer Queueing Systems. *Mathematics* **2025**, *13*, 346. [CrossRef]
4. Bailey, N.T.J. On Queueing Processes with Bulk Service. *J. R. Stat. Soc. Ser. B (Methodol.)* **1954**, *16*, 80–87. [CrossRef]
5. Chaudhry, M.L.; James, G.C.T. *A First Course in Bulk Queues*; Wiley: Hoboken, NJ, USA, 1983
6. Bhat, U.N. *An Introduction to Queueing Theory: Modeling and Analysis in Applications*; Springer: New York, NY, USA, 2008; Volume 36.
7. Medhi, J. *Stochastic Models in Queueing Theory*; Elsevier: Amsterdam, The Netherlands, 2002.
8. Klimko, E.M.; Neuts, M.F. The Single Server Queue in Discrete Time-Numerical Analysis II. *Nav. Res. Logist. Q.* **1973**, *20*, 305–319. [CrossRef]
9. Neuts, M.F. The Single Server Queue in Discrete Time-Numerical Analysis I. *Nav. Res. Logist. Q.* **1973**, *20*, 297–304. [CrossRef]
10. Sule, A.A. Discrete Time Queues and Matrix-Analytic Methods. *Top* **2002**, *10*, 147–185. [CrossRef]
11. Banerjee, A.; Gupta, U.C. Reducing Congestion in Bulk-Service Finite-Buffer Queueing System Using Batch-Size-Dependent Service. *Perform. Eval.* **2012**, *69*, 53–70. [CrossRef]
12. Bar-Lev, S.K.; Parlar, M.; Perry, D.; Stadje, W.; Van der Duyn Schouten, F.A. Applications of Bulk Queues to Group Testing Models with Incomplete Identification. *Eur. J. Oper. Res.* **2007**, *183*, 226–237. [CrossRef]
13. Dieter, C.; Walraevens, J.; Laevens, K.; Bruneel, H. A Queueing Model for General Group Screening Policies and Dynamic Item Arrivals. *Eur. J. Oper. Res.* **2010**, *207*, 827–835. [CrossRef]
14. Veena, G.; Samanta, S.K.; Laxmi, P.V.; Gupta, U.C. Analyzing a Multiserver Bulk-Service Finite-Buffer Queue. *Appl. Math. Model.* **2008**, *32*, 1797–1812. [CrossRef]
15. Yu, M.; Alfa, A.S. Algorithm for Computing the Queue Length Distribution at Various Time Epochs in DMAP/G(1,a,b)/1/N Queue with Batch-Size-Dependent Service Time. *Eur. J. Oper. Res.* **2015**, *244*, 227–239. [CrossRef]
16. Behrokh, K.; Parisay, S. Machine Learning and Simulation: Application in Queuing Systems. *Simulation* **1993**, *61*, 294–302. [CrossRef]
17. Chen, M.-S.; Yen, H.-W. Applications of Machine Learning Approach on Multi-Queue Message Scheduling. *Expert Syst. Appl.* **2011**, *38*, 3323–3335. [CrossRef]
18. Peng, W.; Zhao, J.; Gao, Y.; Sotelo, M.A.; Li, Z. Lane Work-Schedule of Toll Station Based on Queuing Theory and PSO-LSTM Model. *IEEE Access* **2020**, *8*, 84434–84443. [CrossRef]

19. Samson, T.; Teoh, T.T. AI-Driven Optimization of Sanitary Facilities in Office Buildings: A Machine Learning Approach Using LSTM Neural Networks. *Appl. Sci.* **2025**, *15*, 2499. [CrossRef]
20. Tan, C.; Yao, J.; Tang, K.; Sun, J. Cycle-Based Queue Length Estimation for Signalized Intersections Using Sparse Vehicle Trajectory Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 91–106. [CrossRef]
21. Yuan, Z.; Du, P.; Wang, J.; Ba, T.; Ding, R.; Xin, N. Resource Scheduling for Delay Minimization in Multi-Server Cellular Edge Computing Systems. *IEEE Access* **2019**, *7*, 86265–86273. [CrossRef]
22. Pornthep, A.; Limlawan, V. Design of Advanced Queue System Using Artificial Neural Network for Waiting Time Prediction. *Rev. Espac.* **2020**, *798*, 1015.
23. Hassan, H.; Olawoyin, R. Predicting Patient Waiting Time in the Queue System Using Deep Learning Algorithms in the Emergency Room. *Int. J. Ind. Eng.* **2021**, *3*, 33–45. [CrossRef]
24. Elisheva, C.; Cohen, I.; Feigin, P. Delay Prediction for Managing Multiclass Service Systems: An Investigation of Queueing Theory and Machine Learning Approaches. *IEEE Trans. Eng. Manag.* **2022**, *71*, 4469–4479. [CrossRef]
25. Li, X.; Liu, W.; Kong, W.; Zhao, W.; Wang, H.; Tian, D.; Jiao, J.; Yu, Z.; Liu, S. Prediction of Outpatient Waiting Time: Using Machine Learning in a Tertiary Children's Hospital. *Transl. Pediatr.* **2023**, *12*, 2030. [CrossRef] [PubMed]
26. Elisabetta, B.; Aloini, D.; Squicciarini, N. Towards a Real-Time Prediction of Waiting Times in Emergency Departments: A Comparative Analysis of Machine Learning Techniques. *Int. J. Forecast.* **2023**, *39*, 192–208. [CrossRef]
27. Al-Mousa, A.; Al-Zubaidi, H.; Al-Dweik, M. A Machine Learning-Based Approach for Wait-Time Estimation in Healthcare Facilities with Multi-Stage Queues. *IET Smart Cities* **2024**, *6*, 333–350. [CrossRef]
28. Wu, Z.; Wu, H.; Zhao, H.; He, Q.M. Training Neural Networks for the GI/G/K Queue. *Perform. Eval.* **2026**, *172*, 102555. [CrossRef]
29. Meena, A.; Eswari, A.; Rajendran, L. Mathematical modelling of enzyme kinetics reaction mechanisms and analytical solutions of non-linear reaction equations. *J. Math. Chem.* **2010**, *48*, 179–186. [CrossRef]
30. Lata, P.; Banerjee, A.; Gupta, U.C. Batch-size-sensitive queueing analysis with binomial group joining in bulk optional service system under MAP with metaheuristic cost optimization. *Comput. Oper. Res.* **2026**, *194*, 107579. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.