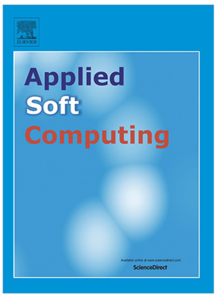


Integrating teletraffic theory with neural networks for quality-of-service evaluation in mobile networks

Yin-Chi Chan, Jingjin Wu, Eric W.M. Wong, Chi Sing Leung

PII: S1568-4946(23)01226-7
DOI: https://doi.org/10.1016/j.asoc.2023.111208
Reference: ASOC 111208

To appear in: *Applied Soft Computing*

Received date : 2 May 2023
Revised date : 14 October 2023
Accepted date : 21 December 2023

Integrating Teletraffic Theory with Neural Networks for Quality-of-Service Evaluation in Mobile Networks

Yin-Chi Chan[a], Jingjin Wu[b,c], Eric W. M. Wong[c,*], Chi Sing Leung[c]

[a]*Institute for Manufacturing, University of Cambridge, Cambridge CB3 0FS, United Kingdom*
[b]*Department of Statistics and Data Science, BNU-HKBU United International College, Zhuhai, Guangdong 519087, China*
[c]*Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China*

Abstract

In mobile cellular design, one important quality-of-service metric is the blocking probability. Using computer simulation for studying blocking probability is quite time-consuming, whereas existing teletraffic-based methods such as the Information Exchange Surrogate Approximation (IESA) only give a rough estimate of blocking probability. Another common approach, direct blocking probability evaluation using neural networks (NN), performs poorly when extrapolating to network conditions outside of the training set. This paper addresses the shortcomings of existing teletraffic and NN-based approaches by combining both approaches, creating what we call IESA-NN. In IESA-NN, an NN is used to estimate a tuning parameter, which is in turn used to estimate the blocking probability via a modified IESA approach. In other words, the teletraffic approach IESA still forms the core of IESA-NN, with NN techniques used to improve the accuracy of the approach via the tuning parameter. Simulation results show that IESA-NN performs better than previous approaches based on NN or teletraffic theory alone. In particular, even when the NN cannot produce a good value for the tuning parameter, for example when extrapolating to network conditions not experienced in the training set, the final IESA-NN estimate is generally still accurate as the estimate is primarily determined by the underlying teletraffic theory, with the NN determining the tuning parameter playing a supplementary role. The combination of

[*]Corresponding author
Email addresses: ycc39@cam.ac.uk (Yin-Chi Chan), jj.wu@ieee.org (Jingjin Wu), eeewong@cityu.edu.hk (Eric W. M. Wong), eeleungc@cityu.edu.hk (Chi Sing Leung)

Table 1: List of Key Abbreviations

Abbreviation	Meaning
BLS	Broad learning system (NN model)
BS	Base station
EFPA	Erlang Fixed Point Approximation
ELM	Extreme learning machine (NN model)
EEM-ELM	Enhancement of error-minimized ELM
IESA	Information Exchange Surrogate Approximation
IESA-CN	IESA for cellular networks
IESA-NN	IESA with neural networks
NN	Neural network
R^{3}S	Round robin with random start
SLFN	Single-hidden-layer feedforward network
QoS	Quality of service

the IESA framework with NN in a secondary role makes IESA-NN quite robust.

Keywords: Neural networks, quality of service, cellular networks, teletraffic, overflow loss systems

1. Introduction

Evaluating quality-of-service (QoS) metrics and meeting minimum QoS requirements form crucial components of many mobile cellular design and optimization problems, including base station (BS) sleeping [1, 2], BS deployment [3, 4], user association [5], dynamic routing [6], network resource allocation [7], and load balancing [8]. Accurate, robust, and computationally efficient algorithms for QoS evaluation are thus very important for obtaining practical solutions in such search-based optimization problems. This is especially true for optimization problems on large-scale networks, which require the QoS evaluation of a large number of candidate solutions.

In particular, the blocking probability of requests in the network is a widely-used QoS metric, defined as the long-term average proportion of mobile user requests not successfully completed. As there are generally no closed-form solutions for evaluating blocking probability and other QoS metrics in practical optimization problems, one traditional way to evaluate such metrics is via computer simulation, which can achieve a high level of accuracy under most circumstances but is considered time-consuming and not scalable for application scenarios with a large number of BSs and a high volume of user requests, e.g. a densely-populated area served by novel millimeter-wave BSs. This disadvantage prevents simulation from being effectively adopted in next-generation mobile applications. The problem of long running times is further exacerbated in new mobile communications applications with very high QoS requirements, for example Ultra-Reliable Low-Latency Communication (URLLC), where the blocking probability requirements are typically at or below 10^{-6}. Therefore, other approaches must be used.

A list of abbreviations used in this paper is given as Table 1.

1.1. Teletraffic theory-based approaches

Another method for blocking probability evaluation is using teletraffic theory-based approaches such as the classical Erlang Fixed Point Approximation (EFPA) [9]. Kelly [9] demonstrated that for networks with fixed routing, a fixed-point algorithm, using the means of the per-link offered traffic alone, is asymptotically exact as the number of channels per link tends to infinity (with the offered traffic increasing in proportion). Although Kelly [9] focused on wired telecommunication networks, Kelly also demonstrated how the method could be applied to channel assignment in a cellular radio network. Finally, Kelly also demonstrated the properties of EFPA when applied to networks with alternate routing. In such cases, EFPA may have *multiple* fixed points corresponding to different metastable states of the network, caused by feedback loops that arise when alternate-routed traffic consume more resources than direct traffic.

However, the two main simplifying assumptions of EFPA, namely that of Poisson traffic (including overflow traffic) and independence between base stations (BSs), result in large approximation errors in some cases [10]. In addition, EFPA tends to be especially inaccurate when the system blocking probability is low (e.g., below 10^{-3}) and when the system exhibits high levels of *mutual overflow*. Mutual overflow is a phenomenon that occurs when traffic offered to overloaded BSs overflows (is redirected) to neighboring BSs, in turn overloading those BSs and yielding overflow traffic to the original BS.

The performance evaluation of overflow loss systems with mutual overflow is a long-standing problem [11]. Over the past decade, a new approach has emerged to address this problem, by modifying EFPA to address its shortcomings for such networks (caused by its simplifying assumptions). This has led to what is known as Information Exchange Surrogate Approximation (IESA) framework [12]. IESA applies an EFPA-based decomposition approach on a fictitious *surrogate system* instead of the real system to be evaluated, aiming to capture features in systems with mutual overflow traffic that are ignored by EFPA. As a result, IESA exhibits significantly improved accuracy and robustness over EFPA, while maintaining EFPA's computational efficiency. To the best of our knowledge, there are **no** viable teletraffic theory-based approaches other than IESA that can effectively and efficiently evaluate blocking probability in overflow loss systems with mutual overflow.

A revised version of IESA, called IESA for Cellular Networks (IESA-CN) [13], was devised for cellular networks, which have unique locality and mobility features not present in previously-considered overflow loss models such as video-on-demand systems. By introducing a tuning parameter to reflect the extent of traffic overflow and mobility among different BSs, IESA-CN can capture the unique features of cellular networks and obtain highly accurate approximations in many cases. However, IESA-CN remains inaccurate in some cases, for example when the offered traffic is extremely low or when user mobility is extremely high. Improving the accuracy and robustness of IESA-CN thus forms the main objective of this paper.

1.2. Neural network-based approaches

Neural networks (NN) are now commonly used in many applications. In particular, single-hidden-layer feedforward network (SLFN) models, such as the broad learning system (BLS) [14] and extreme learning machine (ELM) [15], with a sufficient number of hidden nodes, provide *universal approximation ability*, namely the ability to approximate any continuous function with any desired precision. In the past decade, the NN approach has been adopted for traffic prediction and QoS evaluation in a number of telecommunications applications [16–20].

In particular, the SLFN approach has been used to evaluate blocking probability in optical networks [21]. However, this approach was shown to provide low accuracy when the blocking probability of the network is small (e.g., below 10^{-3}). As the range of blocking probabilities in practical telecommunications networks can span several orders of magnitude, this approach is not suitable for estimating blocking probability values. However, this issue can be readily resolved by first applying a logarithmic transformation to the blocking probability values [17, 18].

In addition, [17, 18] also employ ELM-based approaches for constructing an SLFN for blocking probability estimation in optical networks. Since the input weights and activation biases of the hidden layer nodes in ELM-trained SLFNs are randomly generated, only the output weights need to be computed, using an algorithm that is several orders of magnitude faster than backpropagation in traditional SLFN training algorithms. The output weights can be computed based on a matrix pseudoinverse or incrementally as each hidden node is added to the SLFN [22–24].

Nevertheless, there are some well-known drawbacks of NN-based approaches. Among these, the most fundamental problem is the lack of explainability of NN output (i.e., the *black box* problem [25]), where there is no specific method for determining or interpreting the rationale behind decisions made by an NN. Furthermore, the NN output may be very poor for input values outside the range of the training set; in other words, NNs generally have poor extrapolation capabilities.

1.3. Contributions of this paper

This paper considers a cellular network model with user mobility and mutual overflow traffic among BSs due to dynamic user association mechanisms. We aim to develop an accurate evaluation method for request blocking probability across a wide range of network conditions. Instead of using a pure black-box NN approach, which heavily depends on the number and variability of training samples, this paper proposes a hybrid approach for generating an accurate estimate of blocking probability for a wide range of scenarios.

The fundamental model underlying our proposed approach is based on the IESA framework with an additional tuning parameter k. Whereas k was previously evaluated in IESA-CN using polynomial regression [13], in this paper the value of k is evaluated using an NN. We shall therefore call our proposed method *IESA with neural networks* (IESA-NN). In both IESA-CN and IESA-NN, k depends on network parameters including the offered load to each BS, the level of user mobility, the capacity (number of channels) of each BS, and the neighbor set of each BS cell in the network. We shall use an enhancement of error-minimized ELM (EEM-ELM) [18, 26] to train our NN.

The proposed IESA-NN method in this paper is based on a similar approach in [19] for a generic and highly simplified overflow loss system model. Despite its simplicity, the model considered in [19] still possesses the key element of mutual overflow and demonstrates the benefits of combining the IESA framework with NN to evaluate blocking probability in such networks. In this paper, we further develop IESA-NN by extending it to a much more realistic application/system model with additional features, namely locality (a request may only overflow to a neighboring cell) and user mobility (a user may move between cells during its request). In this paper, we show that despite these additional network features, our IESA-NN approach remains effective at evaluating blocking probability in cellular networks. This again demonstrates the versatility and power of the IESA approach with the help of NN.

Note that in the original IESA-CN [13], which used simple second-order polynomial regression rather than NNs to evaluate k, could not maintain accuracy across the full range of parameters considered. In this paper, we show that, using EEM-ELM, we can obtain improved values of k and in turn produce more accurate and robust ap-

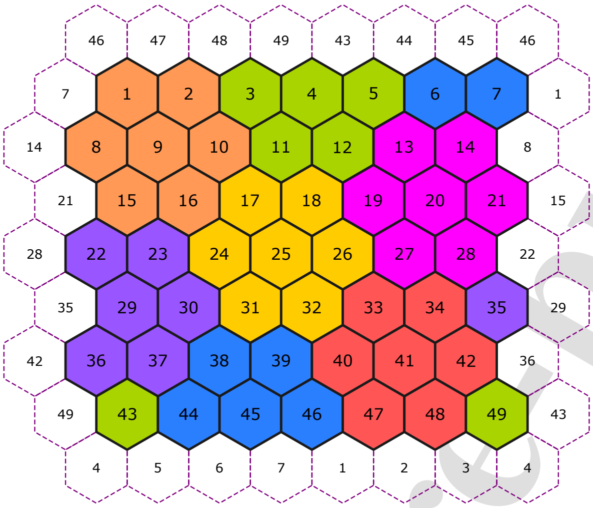


Figure 1: 49-cell wraparound (toroidal) network topology.

proximations of network blocking probability compared to not only IESA-CN but also methods based solely on NN for blocking probability evaluation ("direct NN").

2. Background

2.1. A modern cellular network and its mechanisms

2.1.1. Network Model

We consider a cellular network with $G = 49$ hexagonal cells, as shown in Fig. 1. Such hexagonal lattices are commonly used as simple cellular network models, e.g. [27, 28]. Note, however, that our proposed methodology in this paper can be applied to cellular networks with either regular or irregular topologies, as long as the neighbors of each cell are known. Each cell is associated with a single base station (BS).

There is a one-to-one correspondence between cells and BSs, and we will use the terms "cells" and "BSs" interchangeably. Each BS has a capacity of c channels. We assume negligible inter-cell interference, for example by using an orthogonal frequency-division multiple access (OFDMA) transmission scheme.

Table 2: Cellular Network Model Notations

Symbol	**Meaning**
c	Capacity of each BS (number of channels)
δ	Migration rate of requests between neighboring cells
θ	Probability that a request will undergo further handover before completion, equal to $\delta/(1+\delta)$
N	Number of neighbors of each cell in the network, in this case 6
Γ_i	Set of neighboring cells of cell i
$\gamma_{i,n}$	nth BS in some arbitrary ordering of Γ_i
G	Number of cells/BSs in the network
$g_{i,r,n}$	BS attempted by requests at cell i with n overflows, where r is a random starting index, as defined in (2)
λ_i	Offered load of fresh requests to cell i
$\overline{\lambda}$	Mean load of fresh requests to each cell
B_i	Non-completion probability for requests originating at cell i, i.e., the probability that such requests are blocked (immediately) or dropped (during handover)
Δ	In the fictitious IESA surrogate model, the set of previously attempted BSs of a given request (reset upon handover)
Ω	In the fictitious IESA surrogate model, the congestion estimate attribute of a given request (reset upon handover)

We define a *fresh* request as a request offered to the BS by an end user physically located within that BS's cell, which has not yet attempted access to any other BS. The offered load of fresh requests to each cell i, $i \in \{1, \ldots, G\}$, is denoted as λ_i. The set of neighbors of each cell i is denoted Γ_i, such that $\Gamma_i \subseteq \{1, \ldots, G\} \setminus i$. Note that for the model under consideration, $|\Gamma_i| = N = 6$ for all BSs i in the network.

Requests to the network have an exponentially distributed service time requirement with unit mean. Requests physically located in each cell i will migrate to a neighboring cell in Γ_i with rate δ, triggering a *handover event*. A request newly arrived (physically) in cell i, either a fresh request or a handover request, will first attempt service from BS i. If all channels in BS i are occupied, then the request will attempt to obtain service from each BS in Γ_i in random order. We say that this request *overflows* to the neighbors of BS i. Finally, if all BSs in Γ_i are also fully occupied, the request is *blocked* (for a fresh request) or *dropped* (for a handover request).

A list of notations for the cellular network model is given in Table 2. Hereafter:

- The term "fresh request" refers to a request which has not yet undergone any overflows or handovers.
- The term "handover request" refers to a request which has undergone handover, with no overflows since the most recent handover.
- The term "overflow request" refers to a request that has undergone overflow since its last handover (or since arrival if the request has not undergone any handovers).
- The term "origin cell" refers to the cell at which a fresh request originates.
- The term "starting cell" refers to the cell in which a fresh or handover request is physically situated, that is, the cell from which it first attempts service after arriving at a new physical location.
- The term "serving cell" refers to the cell that is currently serving the request, which may differ from the starting cell due to overflow.

2.1.2. *Handover and handover probability*

We assume that the time between handover events for a given request is exponentially distributed with mean $1/\delta$, and that the probability θ that a request will undergo further handover before completion is independent of elapsed time or the number of previous handovers. Therefore,

$$\theta = \frac{\delta}{1+\delta}. \tag{1}$$

Equation (1) can be explained by noting that a request will either complete at its current serving cell with rate 1, or undergo handover with rate δ, with no other options.

Note that:

- The total load to a BS includes the original traffic of fresh requests, handover traffic, and overflow traffic, which may be caused by both fresh and handover requests.

- A handover request may overflow **back** to the previous serving cell and continue to be served there, if it is a neighbor of the new starting cell. On the other hand, the old serving cell may have a maximum distance of **two** from the new starting cell and a maximum distance of **three** from the new serving cell, forming a straight line of adjacent cells: old serving cell, old starting cell, new starting cell, new serving cell..

2.1.3. *Round Robin with Random Start (R^{3}S)*

As allowing full random routing of overflow traffic results in intractable computational complexity, we shall approximate random routing with *round robin with random start* (R^{3}S) [12]. Recall that N is the number of neighbors of each BS in our model, and let $(\gamma_{i,1}, \ldots, \gamma_{i,N})$ be an arbitrary ordering of the neighbor set Γ_i for each BS i in the network. For convenience, define

$$[\![\chi]\!] = \chi - \left\lfloor \frac{\chi - 1}{N} \right\rfloor N.$$

In other words, $[\![\chi]\!]$ equals χ minus the largest multiple of N strictly less than χ, such that $[\![\chi]\!] \in \{1, \ldots, N\}$ for all positive integer χ.

The overflow count of a request is reset to zero upon a new handover attempt. Finally, the index of the BS that a request with starting cell *i* and *n* overflows is offered to, $n \in \{0, \dots, N\}$, is:

$$g_{i,r,n} = \begin{cases} i, & n = 0 \\ \gamma_{i,[\![j+n-1]\!]}, & n = 1, \dots, N, \end{cases} \tag{2}$$

where *r* is a random starting index. A graphical example depicting $g_{i,r,n}$ for a given cell *i* is provided in Fig. 2. We can see that $g_{i,1,n} = \gamma_{i,n}$ and $g_{i,4,n} = \gamma_{i,[\![n+3]\!]}$ for $n \in \{1, \dots, 6\}$, with $\gamma_{i,n}$ defined in a clockwise fashion around cell *i*.

In this paper, we will use $\mathrm{R}^3\mathrm{S}$ to approximate full random routing in EFPA, IESA-CN, and IESA-NN, while comparing against simulation results using full random routing. Numerical results in [12] demonstrated that $\mathrm{R}^3\mathrm{S}$ is a close approximation to full random routing in terms of blocking probability evaluation.

2.2. EFPA

EFPA is based on two simplifying assumptions:

1. All traffic in the network (including handover and overflow traffic) is Poisson.
2. The traffic streams to all BSs are mutually independent.

The two assumptions above lead to a set of fixed-point equations that can be solved via iterative substitution [29]. Furthermore, each BS can be modeled using a simple Erlang B queue model. A brief derivation of EFPA for the current cellular network model is given in Appendix A.

2.3. IESA and IESA-CN

IESA [12] is an EFPA-based approach that has been demonstrated to improve accuracy in blocking probability estimation compared to the original EFPA. IESA applies EFPA on a fictitious hierarchical *surrogate* system, where each request has two specifically designed attributes, denoted Δ and Ω. Specifically, Δ records the BSs that a request has attempted and been rejected from (due to lack of capacity), whereas Ω serves as an estimate of the number of busy (full) BSs in the network.

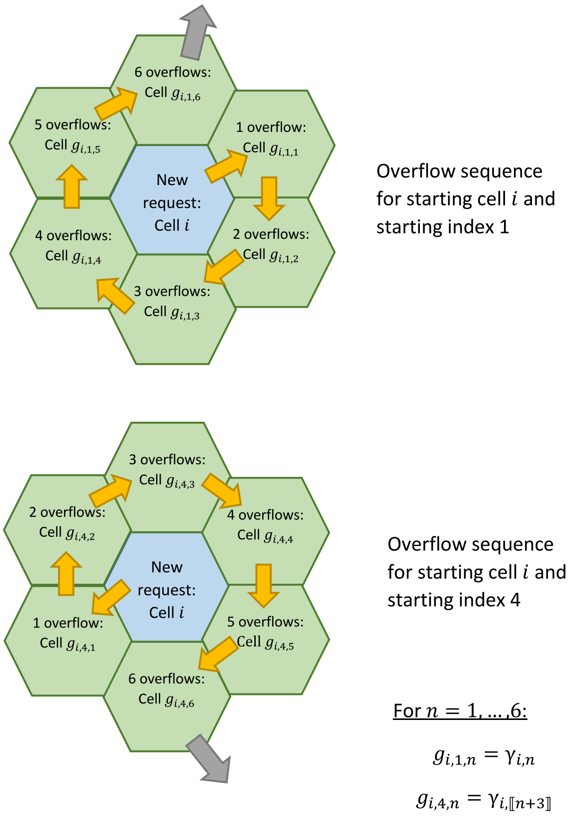

6 overflows: Cell $g_{i,1,6}$
5 overflows: Cell $g_{i,1,5}$
1 overflow: Cell $g_{i,1,1}$
New request: Cell i
4 overflows: Cell $g_{i,1,4}$
2 overflows: Cell $g_{i,1,2}$
3 overflows: Cell $g_{i,1,3}$
Overflow sequence for starting cell i and starting index 1
3 overflows: Cell $g_{i,4,3}$
2 overflows: Cell $g_{i,4,2}$
4 overflows: Cell $g_{i,4,4}$
New request: Cell i
1 overflow: Cell $g_{i,4,1}$
5 overflows: Cell $g_{i,4,5}$
6 overflows: Cell $g_{i,4,6}$
Overflow sequence for starting cell i and starting index 4
For $n = 1, \ldots, 6$:
$g_{i,1,n} = \gamma_{i,n}$
$g_{i,4,n} = \gamma_{i,[\![n+3]\!]}$

Figure 2: Routing sequences depicting $g_{i,r,n}$ for a given cell i and two different starting indexes r. In this example $\gamma_{i,n}$, $n = 1, \ldots 6$, is defined in a clockwise order. Gray arrows denote blocked/dropped requests after $N + 1 = 7$ overflows.

For an overflowing request with attributes $\Omega = j$ and $|\Delta| = n$, the probability that it will immediately abandon the network (i.e. blocking or dropping) without attempting the remaining BSs is:

$$P_{k,n,j} = \begin{cases} 0, & j < N \\ \frac{\binom{j-n}{N-n}}{\binom{k-n}{N-n}}, & j \geq N, \end{cases} \tag{3}$$

where k is a parameter denoting the maximum allowed Ω value of requests in the network. In addition to the abandonment policy defined by (3), the fictitious IESA surrogate system also introduces rules for updating Δ and Ω after each overflow, which are detailed in Appendix B. Note that during a handover event, a request's Δ and Ω are reset to $\emptyset$ (the empty set) and 0, respectively.

Based on the traffic hierarchy on Ω created by the abandonment policy, an EFPA-like process can be applied to estimate B_i for each BS i in the network. A full derivation is provided in Appendix B. Note that the abandonment policy defined by (3) creates bounds on the blocking probability estimated generated by IESA: $k = 1$ creates a system with no overflows, thus maximizing the blocking probability, while $k \to \infty$ disables the abandonment policy and causes IESA to converge to EFPA. In other words, the accuracy of IESA-based approximations depend heavily on the choice of tuning parameter k. In the original IESA [12], designed for video-on-demand systems, k was fixed to G, which is not appropriate for networks such as cellular networks with strong locality and mobility effects. To enhance the accuracy of IESA for cellular networks, in [13] the concept of IESA for cellular networks (IESA-CN) was proposed, in which the value of k is obtained via second-order polynomial regression, with training values obtained via simulation.

Figures 3 and 4 give the basic idea of the IESA and IESA-CN approaches. The inputs are defined as in Table 2, with $\overline{\lambda} = \sum_i \lambda_i / G$. In both cases, the IESA component takes the network parameters and the tuning parameter k as input and outputs the blocking probability B_i for each BS i. The IESA methodology itself, shown in the figures as "IESA steps", is desribed in detail in Appendix B.

The differences between IESA and IESA-CN can be described as follows. In IESA, k is a fixed input parameter, whereas in IESA-CN it is derived from the network pa-

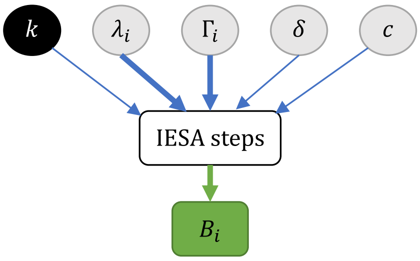


Figure 3: Conceptual depiction of IESA. See Table 2 for a list of notations. Bold arrows denote vector inputs/outputs.

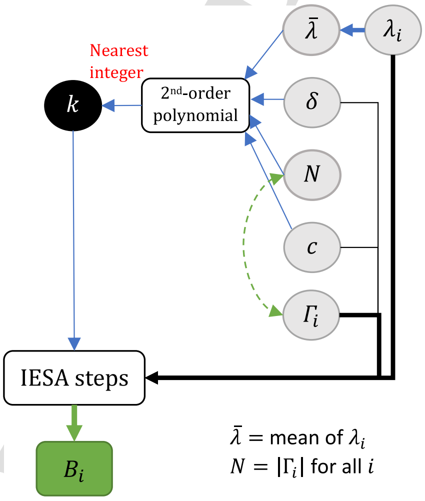


Figure 4: Conceptual depiction of IESA-CN. See Table 2 for a list of notations. Bold arrows denote vector inputs/outputs. Note that all inputs to the second-order polynomial are scalars.

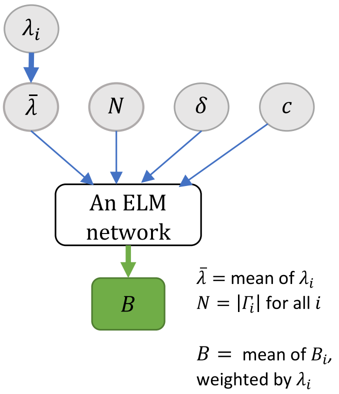

λ_i
$\bar{\lambda}$
N
δ
c
An ELM network
B
$\bar{\lambda}$ = mean of λ_i
$N = |\Gamma_i|$ for all i
$B =$ mean of B_i, weighted by λ_i

Figure 5: Conceptual depiction of Direct-NN. See Table 2 for a list of notations. Note that all inputs to the ELM network are scalars.

rameters. Note that k can be interpreted physically as the expected number of BSs a request is expected to visit during its service, due to both overflow and handover, and is therefore rounded to the nearest integer in IESA-CN.

2.4. *Direct NN*

Since SLFNs with sufficient hidden nodes provide universal approximation ability [14, 15], an SLFN was proposed in [16] for evaluating blocking probability in optical networks. This approach first uses computer simulation to obtain a training set $\{(\mathbf{x}_\ell, y_\ell) \mid \ell = 1, \ldots, s\}$, where $\mathbf{x}_\ell$ is a vector containing the network parameters of the ℓth training sample, y_ℓ is the corresponding blocking probability, and s is the number of training samples. However, direct use of the y_ℓ's as NN target values was shown to yield poor results when the blocking probability values are small.

To handle the large range issue when estimating blocking probability values, a logarithmic transformation can be applied [17, 18], such that the training set of the SLFN becomes $\{(\mathbf{x}_\ell, \log(y_\ell)) \mid \ell = 1, \ldots, s\}$. In addition, ELM-based learning [17, 18] provides a more computationally efficient incremental approach to constructing the SLFN [22–24] that is several orders of magnitude faster than backpropagation for traditional NNs. We can extend the approach of [17, 18] to handle the case of cellular networks, as shown in Figure 5. In this approach, the input vector $\boldsymbol{x}_\ell$ for a given sample network ℓ is its collection of network parameters $\overline{\lambda}$, N, δ, and c, as defined in Table 2, and the corresponding output, is the logarithm of the network probability. As the (logarithm of the) blocking probability is computed directly from the network parameters, we call this the Direct-NN approach.

The Direct-NN approach, as shown in Fig. 5, suffers from a major shortcoming known as the *black-box problem* [25], where the behavior of a trained NN cannot be readily explained. Additionally, NNs can only be trained based on the available data, and do not contain any intrinsic knowledge of the system to be approximated. Therefore, NNs generally do not posses any extrapolation ability to input parameter ranges not previously seen, as shown in Section 4.

In the remainder of this paper, we propose a novel hybrid approach for blocking probability evaluation in mobile cellular networks. In this hybrid approach, the fundamental evaluation step is based on the IESA framework with tuning parameter k, as in IESA-CN. However, this value k is now evaluated using a trained SLFN rather than via polynomial regression, as in IESA-CN. Finally, an analytic continuation of (3) is used to allow for non-integer values of k to be used within the IESA framework, based on the identities $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ and $n! = \Gamma(n+1)$. Numerical results in Section 4 demonstrate that the new hybrid approach, which we call IESA-NN, is more accurate and robust than both IESA-CN and Direct-NN, and has good extrapolation ability that is lacking in Direct-NN.

3. Hybrid Learning Approach: IESA-NN

3.1. Overview

This section considers a hybrid learning approach, namely IESA-NN, for estimating blocking probability in cellular networks. In hybrid learning [30, 31], machine learning and conventional models are combined to produce more accurate results than can be obtained via either method alone. In particular, the conventional model controls extrapolation in regions of input space that lack training data, while the neural network compensates for inaccuracies in the conventional model [30], or is used to estimate its

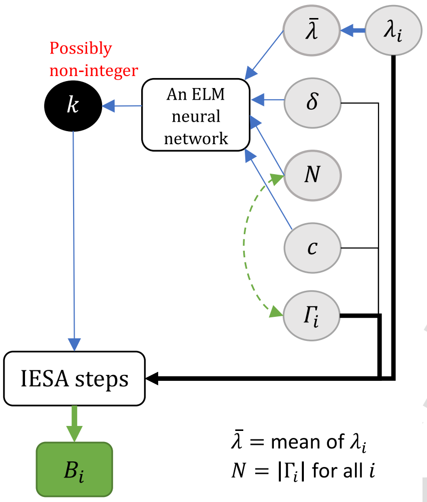


Figure 6: Conceptual depiction of IESA-NN. See Table 2 for a list of notations. Bold arrows denote vector inputs/outputs. Note that all inputs to the ELM network are scalars.

parameters, e.g., transmission and recovery rates in epidemiological models [32]. Such approaches can be thought of as a form of "theory-guided data science" [33]. Other applications of hybrid learning approaches include power systems [34, 35], oil and gas delivery [36], geology [37], and fluid mechanics [38].

A conceptual depiction of IESA-NN is shown in Fig. 6. The network parameters $\bar{\lambda}$, δ, N, and c, as defined in Table 2, are first inputted into an ELM network, which outputs an appropriate value of the tuning parameter k of the IESA algorithm. Then, the IESA algorithm gives an estimate of the network blocking probability as the final output. In the following subsections, we will describe the training procedure of the ELM network.

Numerical results in Section 4 demonstrate that IESA-NN is more accurate and robust than existing IESA approaches including the original IESA and IESA-CN. Although replacing the default k of IESA with a fitted value in IESA-CN improves accuracy, the performance of IESA-CN is still not ideal compared to IESA-NN, due to the limited approximation ability of the second-order polynomial in IESA-CN. Additionally, using an analytic continuation of (3) also improves performance of IESA-NN compared to IESA-CN.

IESA-NN is also shown in Section 4 to outperform Direct-NN. For Direct-NN, when the network parameters fall outside the range of the training set, the NN may not be able to handle those particular network conditions and will therefore yield inaccurate results. On the other hand, even when previously unobserved network parameters result in a poor choice of k by the NN, final estimation of the request blocking probability is still guided by a teletraffic model and thus does not deviate from the true value too much.

3.2. Preparing the training data

In order to train an ELM network to output appropriate values of k, we need a training set $\{(\mathbf{x}_\ell, k_\ell) \mid \ell = 1, \ldots, s\}$, where $\boldsymbol{x}_\ell$ is a vector containing the network parameters of the ℓth training sample, and k_ℓ is the corresponding value of k for IESA-NN. The following steps summarize the method for obtaining k_ℓ for each training sample ℓ:

1. Given network parameters $\boldsymbol{x}_\ell$, use computer simulation to obtain the corresponding blocking probability B_ℓ^{sim}.
2. Using the IESA algorithm, use bisection search to find a value $\hat{k}$ such that $B_\ell^{\text{sim}}\left(\hat{k}\right) = B_\ell^{\text{sim}}$ and assign $\hat{k}$ to k_ℓ. Use the best fit if no match is found within the search range – since $B_\ell^{\text{sim}}\left(\hat{k}\right)$ is monotonic in $\hat{k}$, this will be one of the search bounds. For this paper, the search bounds are 7 and 49, i.e. $N + 1$ and G.

3.3. ELM: notation

We consider an ELM network with a single hidden layer and N_h hidden nodes [15]. The output of the ELM network is given by

$$f_{N_h}(\mathbf{x}) = \sum_{h=1}^{N_h} w_h \phi_h(\mathbf{x}),$$

where $\boldsymbol{x}$ is a vector containing the network parameters for a given cellular network, w_h is the weight between the hth hidden node and the output node, and ϕ_h is the output of the hth hidden node. In this paper, we use the sigmoid function as the activation function for the hidden nodes, such that

$$\phi_h(\mathbf{x}) = \frac{1}{1 + 1/\exp\left\{\boldsymbol{\xi}_h^\intercal \mathbf{x} + \beta_h\right\}},$$

where $\boldsymbol{\xi}_h$ is the input weight vector of the hth hidden node, and β_h is the corresponding activation bias, which are randomly generated for each hidden node h [15].

Consider a training set with N_s samples, i.e. $\{(\mathbf{x}_\ell, k_\ell) \mid \ell = 1, \ldots, s\}$, where $\mathbf{x}_\ell$ and k_ℓ are the input vector and output value of the ℓth training sample, respectively. For each hidden node h, let

$$\boldsymbol{\varphi}_h = \begin{bmatrix} \phi_h(\mathbf{x}_1) \\ \vdots \\ \phi_h(\mathbf{x}_{N_s}) \end{bmatrix},$$

and let $\boldsymbol{\Phi}_{N_h} = \left[\boldsymbol{\varphi}_1 \mid \ldots \mid \boldsymbol{\varphi}_{N_h}\right]$ denote the output matrix of all the hidden nodes over the entire training set. The training objective is then

$$\arg\min_{\mathbf{w}} J_{N_h}(\mathbf{w}), \tag{4}$$

where $J_{N_h}(\mathbf{w})$ is the objective function

$$J_{N_h}(\mathbf{w}) = \sum_{\ell=1}^{N_s} \left(k_\ell - f_{N_h}(\mathbf{x}_\ell)\right)^2 = \left\|\mathbf{k} - \boldsymbol{\Phi}_{N_h}\mathbf{w}\right\|_2^2,$$

where $\mathbf{k} = [k_1 \ldots k_{N_s}]^\intercal$ is the vector of target outputs and $\mathbf{w} = [w_1, \ldots, w_{N_s}]^\intercal$ are the output weights of the hidden layer. The solution to (4) is simply $\mathbf{w}_{N_h} = \boldsymbol{\Phi}^\dagger \mathbf{k}$, where † denotes the Moore-Penrose pseudoinverse.

3.4. Building the ELM network incrementally: EEM-ELM

In EEM-ELM [23, 26], hidden nodes are added to the NN incrementally. For each hidden node h, the input weights $\boldsymbol{\xi}_h$ and bias term β_h are generated randomly and then fixed as additional hidden nodes are added. EEM-ELM provides a method of updating the output weights $\mathbf{w}$ as each hidden node is added without retraining the entire network. This reduces the complexity and running time of the training process.

Suppose the current ELM network contains N_h hidden nodes. Adding an $(N_h + 1)^{\text{th}}$ hidden node yields $\widehat{\mathbf{\Phi}}_{N_h+1} = [\mathbf{\Phi} \mid \widehat{\varphi}_{N_h+1}]$. The following recursive relationship can be used to compute $\widehat{\mathbf{w}}_{N_h+1}$ efficiently [23, 26]:

$$\begin{aligned}
\mathbf{Q}_{N_h+1} &= \left(\left(\mathbf{I} - \mathbf{\Phi}_{N_h}\mathbf{\Phi}^{\dagger}_{N_h}\right)\boldsymbol{\varphi}_{N_h+1}\right)^{\dagger} \\
\mathbf{T}_{N_h+1} &= \mathbf{\Phi}^{\dagger}_{N_h}\left(\mathbf{I} - \boldsymbol{\varphi}_{N_h+1}\mathbf{Q}_{N_h+1}\right)^{\dagger} \\
\widehat{\mathbf{w}}_{N_h+1} &= \begin{bmatrix} \mathbf{T}_{N_h+1} \\ \mathbf{Q}_{N_h+1} \end{bmatrix}\mathbf{k}.
\end{aligned}$$

Finally, at each iteration in EEM-ELM, j candidate hidden nodes are generated and only the candidate yielding the best estimation error is permanently added to the ELM network; thus

$$\varphi_{N_h+1} = \arg\min_{\widehat{\varphi}_{N_h+1}} J_n = \arg\min_{\widehat{\varphi}_{N_h+1}} \left\| \mathbf{k} - \widehat{\mathbf{\Phi}}_{N_h+1}\widehat{\mathbf{w}}_{N_h+1} \right\|_2^2$$

4. Numerical Results

In this section, we demonstrate and compare the performance of four different approaches for blocking probability evaluation in cellular networks, namely EFPA, IESA-CN, Direct-NN, and our proposed IESA-NN.

4.1. Settings and datasets

We consider a 49-cell wraparound model as defined in Section 2.1. As in [13], we designate a seven-cell cluster, denoted $\mathbf{H}$, as the "hot cluster", in which the arrival rate to the BSs differs from the rest of the network by a ratio of α. In other words, for all BS i, $i \in \{1, \ldots, G\}$,

$$\lambda_i = \begin{cases} \lambda, & i \notin \mathbf{H} \\ \alpha\lambda, & i \in \mathbf{H}. \end{cases}$$

Note that $\mathbf{H}$ is defined to contain one cell and its six neighbors, i.e. one of the colored groups in Fig. 1.

To demonstrate that our proposed IESA-NN methodology can accurately and efficiently evaluate blocking probabilities in mobile networks, in this section we consider

Table 3: Summary of Training and Test Set Parameters

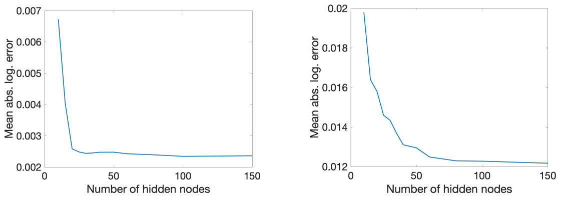
Scenario	Training samples	Test samples	Training set blocking prob. range	Test set blocking prob. range
In-Sample-1	2337	1002	1.36×10^{-4} to 0.075	
In-Sample-2	1377	386	1.36×10^{-4} to 0.01	
Out-Sample-1	2088	1251	0.001 to 0.075	1.36×10^{-4} to 0.075
Out-Sample-2	1983	356	0.001 to 0.01	1.36×10^{-4} to 0.001

Table 4: Cellular Network Parameters for the Four Scenarios

Parameter	Value(s)
$\lambda_i,\ i \notin \mathbf{H}$	7 to 10
α	0.8 to 2
c	10
δ	1
N	6 (49-cell wraparound topology)

four scenarios, as described in Table 3. The training and test datasets for each scenario are generated by regular sampling of the parameter space as shown in Table 4, with the "true" blocking probability for each parameter setting found via computer simulation. The training and test datasets are then filtered according to the blocking probability ranges given in Table 3, giving the number of training and test samples shown. Note that in Scenarios In-Sample-1 and In-Sample-2, the blocking probabilities for the training and test sets cover the same range, for Out-Sample-1 the ranges partially overlap, and for Out-Sample-2 they are disjoint.

Figure 7 shows the training error (mean absolute logarithmic error) of Direct-NN and IESA-NN as applied to In-Sample-1. The results show that both algorithms pro-



(a) Direct-NN

(b) IESA-NN

Figure 7: Mean absolute logarithmic errors for Direct-NN and IESA-NN as applied to the In-Sample-1 scenario.

duce near-optimal error results within 100 hidden nodes. Therefore, we shall use $N_h = 100$ hidden nodes for all remaining results in this paper. Note, however, that since the NN in Direct-NN estimates the blocking probability directly whereas the NN in IESA-NN estimates the tuning parameter k, the training errors of the two methods are not comparable.

The accuracy of out-of-sample testing (e.g., scenarios Out-Sample-1 and Out-Sample-2) is particularly useful and important in network design and optimization problems, especially in 5G URLLC (ultra-reliable low-latency communication) applications where the target blocking probability is very low due to strict QoS requirements. In such applications, it is difficult and time-consuming to obtain accurate simulation results for such low blocking probabilities. This can be avoided if the blocking probability of such networks can be accurately estimated based on training data with higher blocking probabilities, which are easier to simulate.

The approximation results are shown in Figs. 8–12 using scatter plots, where the horizontal axes represent simulated blocking probabilities and the vertical axes represent blocking probabilities obtained by each of the approximation methods. For Figs. 9–12, the distributions of the relative errors are also shown in histogram form.

4.2. In-sample scenarios

In this subsection, we examine the approximation results for scenarios where the training and test sets are sampled from the same parameter range. First, we consider

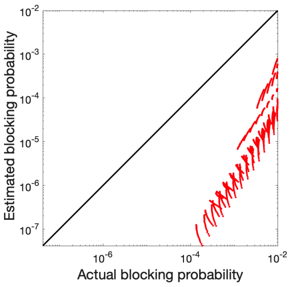


Figure 8: EFPA results for Scenario In-sample-1, as defined in Table 3.

the classical EFPA method as a separate case. Note that EFPA is purely teletraffic-based and does not require a training set. However, the results, shown in Figure 8, demonstrate that EFPA underestimates blocking probability by at least one order of magnitude across the entire parameter range considered. Therefore, EFPA is not a viable approximation method for network design and optimization.

Therefore, we shall hereafter focus on the other three approaches only, namely IESA-CN, Direct-NN, and IESA-NN. The results of these three approaches for the in-sample scenarios are shown in Figures 9 and 10. It is shown that the performance of IESA-CN is the worst among the three methods, with a tendency to overestimate blocking probability for small values. This is because due to the limited approximation ability of the second-order polynomial and the restriction of k to integer values only. On the other hand, the Direct-NN and IESA-NN approaches yield similar results for both scenarios, with close to zero relative error in the large majority of cases, as shown by the histograms in the bottom parts of Figures 9 and 10.

4.3. Out-of-sample scenarios

In this subsection, we examine the approximation results for scenarios where the training and test sets are sampled from different parameter ranges. Results are shown in Figures 11 and 12. The results demonstrate that IESA-NN is the most accurate and robust among the three methods shown. In particular, Fig. 11 demonstrates that Direct-NN, while accurate for the portion of the test set that overlaps with the training

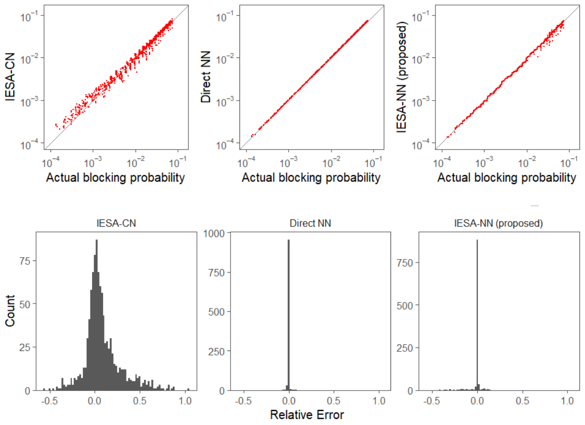


Figure 9: Results for Scenario In-Sample-1, as defined in Table 3.

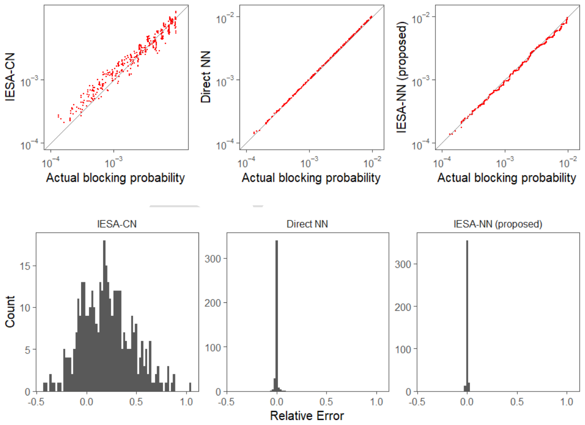


Figure 10: Results for Scenario In-Sample-2, as defined in Table 3.

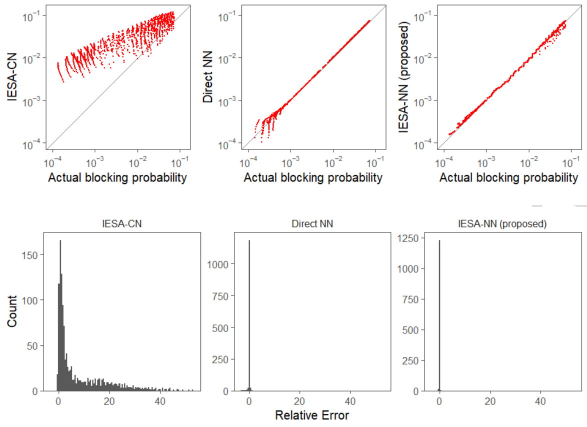


Figure 11: Results for Scenario Out-Sample-1, as defined in Table 3.

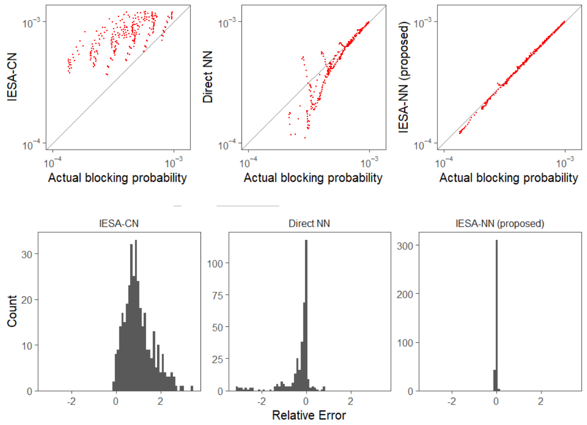


Figure 12: Results for Scenario Out-Sample-2, as defined in Table 3. Note negative Direct-NN estimates cannot be shown in the scatter plot, but are visible in the histogram (relative error less than -1).

set, becomes increasingly inaccurate as the blocking probability decreases beyond the lower limit of the training set. Furthermore, Direct-NN can even produce negative blocking probability estimates, as shown by some cases having less than negative one relative error. This is not possible with IESA-NN.

On the other hand, the combination of an NN and teletraffic theory yields more accurate and robust results than either approach alone. Recall that the abandonment policy defined by (3) creates bounds on the blocking probability estimated generated by IESA, thus minimizing the effect of suboptimal choice of k when the blocking probability is low. The robustness of IESA-NN compared to Direct-NN is best demonstrated in Fig. 12, where IESA-NN is shown to have a much tighter error distribution than Direct-NN.

4.4. Discussion

From the results in this section, it is demonstrated that while Direct-NN gives the best performance of the three methods under consideration when the test set parameter ranges fall within those of the training set (Figs. 9 and 10), IESA-NN is the most robust method when extrapolating to new parameter ranges not in the training set (Figs. 11 and 12). The reason for this is because of nature of NN – its purpose is to fit a regression to the data seen, without any interpretation of underlying structures. In contrast, in IESA-NN, the NN component is only used to fit a tuning parameter, with the underlying teletraffic theory supporting IESA-NN providing some protection against wildly inaccurate results. Additionally, whereas the range of possible blocking probabilities in our training/test data spans several orders of magnitude, the search range for our IESA-NN tuning parameter is much narrower, namely 7 to 49. This makes it easier for the NN to fit the training data more accurately. Finally, while IESA-CN is similar to IESA-NN in that a tuning parameter is used to adjust the IESA result, the polynomial-regression-based method used in IESA-CN is less flexible than the NN-based method in IESA-NN, thus leading to less accurate results for IESA-CN.

Note that although our generated training and test datasets in this section are artificial, the parameter space outlined in Table 4 covers a wide range of scenarios that may appear in real situations. Therefore, our proposed method is expected to obtain accurate approximations in a computationally efficient manner even if real datasets are used.

5. Concluding Remarks

In this paper, we proposed the IESA-NN approach for approximating blocking probability in cellular mobile networks with user mobility, by combining classic teletraffic theory with neural network techniques. Blocking probability forms an important metric in such networks. Specifically, IESA-NN adopts a neural network approach to estimate a key parameter in the IESA framework. The results demonstrate that IESA-NN significantly outperforms both direct-NN and pure teletraffic-based approaches, especially when extrapolating beyond the parameter range of the training data. This is because the NN portion of IESA-NN can compensate for inaccuracies in base IESA, while the teletraffic theory underlying IESA controls extrapolation of IESA-NN in regions that lack training data for the NN.

The improvement in accuracy of IESA-NN over previous approaches is important for application scenarios such as 5G URLLC, where the request blocking probability may be extremely low. Such cases cause accurate simulation of such networks to become especially time-consuming, especially in optimization scenarios where blocking probability results are required for a large number of parameter settings.

Appendix A. Derivation of EFPA

In addition to the notation defined in Table 2, we also define:

- $a_{i,r,n}$ — Offered traffic with starting cell i, n overflows and starting index r, including both non-handover and handover requests. The BS receiving this traffic is $g_{i,r,n}$ as defined by (2).
- $e_{i,n}$ — Total offered traffic to BS i with n overflows.
- A_i — Total offered traffic to BS i.
- $\underline{A}_i$ — Total carried traffic of BS i, including traffic that may undergo further handover.

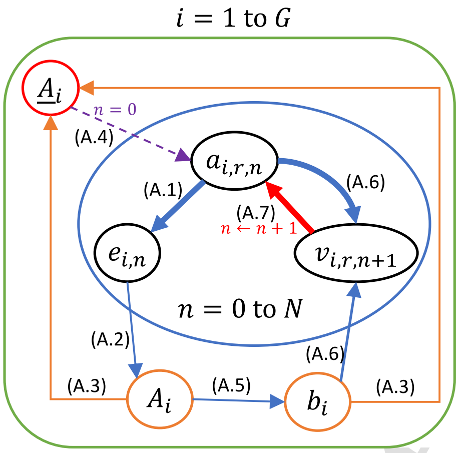


Figure A.1: Update sequence for EFPA. Thick lines indicate updates for each iteration of the inner (n) loop, while thin lines indicate updates for each iteration of the outer (i) loop. Note that multiple fixed-point iterations are required for convergence. Numbers in parentheses refer to equation numbers in this paper.

- b_i — Probability that all channels in BS i are busy.
- $v_{i,r,n}$ — Overflow traffic with starting cell i, n overflows, and starting index r. The BS from which this traffic overflows is $g_{i,r,n-1}$ as defined by (2).
- B_i — Non-completion probability for requests originating at cell i (before any handovers), i.e., the probability that such requests are blocked (immediately) or dropped (during handover).

We obtain

$$e_{i,n} = \sum_{(q,r):g_{q,r,n}=i} a_{q,r,n} \tag{A.1}$$

$$A_i = \sum_{n=0}^{N} e_{i,n} \tag{A.2}$$

$$\underline{A}_i = A_i\,(1 - b_i) \tag{A.3}$$

We also have

$$a_{i,r,0} = \frac{\lambda_i}{N} + \frac{1}{N^2} \sum_{q:i\in\Gamma_q} \underline{A}_q \theta, \tag{A.4}$$

where the first term of the sum represents fresh traffic and the second term represents handovers from the neighboring cells of cell i. To explain (A.4), note that the handover traffic $\underline{A}_q\theta$ is first divided among all N neighbors of cell q, then among all N possible starting indexes for R^{3}S (corresponding to the "j" index in $a_{i,j,0}$).

Applying the Poisson assumption, we obtain

$$b_i = E(A_i, c) \tag{A.5}$$

and

$$v_{i,r,n} = a_{i,r,n-1} b_i, \tag{A.6}$$

where $E(A_i, c)$ denotes the Erlang B formula with A_i Erlangs of traffic and c channels. Based on the R^{3}S routing policy, we obtain for all $n \in \{1, \ldots, N\}$:

$$a_{i,r,n} = v_{i,r,n}. \tag{A.7}$$

The relationships between (A.1)–(A.7) form a system of fixed-point equations, as illustrated in Fig. A.1. The equations can be solved via iterative substitution, with initial values $a_{i,r,0} \leftarrow \lambda_i/N$ and all other initial values set to zero. Finally, the overall non-completion probability for requests with origin cell i is defined as

$$B_i = (1-\theta) b_i + \frac{\theta}{N} \sum_{q \in \Gamma_i} B_q. \tag{A.8}$$

Appendix B. Derivation of IESA

As IESA, IESA-CN, and IESA-NN differ only in the method by which k is chosen in equation (3) (see Figs. 3, 4, and 6), in this section we present a generic set of equations encompassing all three approximation methods.

In addition to the notation defined in Table 2, we also define:

- $a_{i,r,n,j}$ — Offered traffic with starting cell i, n overflows, a congestion estimate of j, and starting index r, including both non-handover and handover requests. The BS receiving this traffic is $g_{i,r,n}$ as defined by (2).

- $e_{i,n,j}$ — Total offered traffic to BS i with n overflows and a congestion estimate of j.

- $\overline{a}_{i,r,n,j}$ — Total offered traffic to BS $g_{i,r,n}$ with starting cell i, n overflows, a congestion estimate of *j or less*, and starting index r.

- $\overline{e}_{i,n,j}$ — Total offered traffic to BS i with n overflows and a congestion estimate of *j or less*.

- $A_{i,j}$ — Total offered traffic to BS i at level j of the IESA hierarchy, consisting of requests with congestion estimates of at most j.

- $\underline{A}_i$ — Total carried traffic of BS i, including traffic that may undergo further handover.

- $v_{i,r,n,j}$ — Overflow traffic with starting cell i, n overflows, a congestion estimate of j, and starting index r, including both non-handover and handover requests. The BS from which this traffic overflows is $g_{i,r,n-1}$ as defined by (2).

- $z_{i,r,n,j}$ — Blocked or dropped traffic with starting cell i, n overflows, a congestion estimate of j, and starting index r, including both non-handover and handover requests. The BS from which this traffic overflows is $g_{i,r,n-1}$ as defined by (2).

- $b_{i,j}$ — Probability that all channels in BS i are busy at level j of the IESA hierarchy, i.e., all channels are serving requests with congestion estimates of at most j.

By definition, we have:

$$e_{i,n,j} = \sum_{(q,r):g_{q,r,n}=i} a_{q,r,n,j} \tag{B.1}$$

$$\overline{a}_{i,r,n,j} = \sum_{p=n}^{j} a_{i,r,n,p} \tag{B.2}$$

$$\overline{e}_{i,n,j} = \sum_{p=n}^{j} e_{i,n,p} \tag{B.3}$$

$$A_{i,j} = \sum_{n=0}^{N} \overline{e}_{i,n,j} \tag{B.4}$$

$$a_{i,r,0,0} = \frac{\lambda_i}{N} + \frac{1}{N^2} \sum_{q:i\in\Gamma_q} \underline{A}_q \theta. \tag{B.5}$$

Using the Erlang B formula, we obtain

$$b_{i,j} = E\left(A_{i,j}, c\right). \tag{B.6}$$

To obtain the overflow traffic from BS i for a given congestion estimate j and n overflows, we consider two scenarios:

- In the first scenario, a request with $n - 1$ overflows and a congestion estimate of $j - 2$ or less finds, with probability $B_{i,j-1} - B_{i,j-2}$, that all channels at BS i are busy and the most senior (highest congestion estimate) request at BS i has a congestion estimate of $j - 1$. The incoming request exchanges congestion estimates with the senior request and overflows with a new congestion estimate of j (note that the congestion estimate is incremented upon overflow regardless of whether exchange of congestion information occurs).

- In the second scenario, a request with $n - 1$ overflows and a congestion estimate of exactly $j - 1$ find, with probability $B_{i,j-1}$,that all channels at BS i are busy and all requests in service have congestion estimates of $j - 1$ or less. No information exchange occurs and the incoming request simply increments its congestion estimate by one upon overflow.

Thus we obtain

$$\begin{aligned} v_{i,r,n,j} &= \overline{a}_{i,r,n-1,j-2}\left(b_{i,j-1} - b_{i,j-2}\right) \\ &\quad + a_{i,r,n-1,j-1} b_{i,j-1} \\ &= \overline{a}_{i,r,n-1,j-1} b_{i,j-1} \\ &\quad + \overline{a}_{i,r,n-1,j-2} b_{i,j-2}. \end{aligned} \tag{B.7}$$

By definition, values with negative indices are all zero. Applying the abandonment policy, we obtain

$$z_{i,r,n,j} = v_{i,r,n,j} P_{k,n,j} \tag{B.8}$$

$$a_{i,r,n,j} = v_{i,r,n,j}\left(1 - P_{k,n,j}\right). \tag{B.9}$$

where $P_{k,n,j}$ is defined in (3). Note also that $P_{k,n,j} = 1$ if Ω reaches k or $|\Delta|$ reaches N. Finally, note that the highest level of the IESA hierarchy, containing all offered traffic, is level $k-1$. Therefore, the total carried traffic by BS i is

$$\underline{A}_i = A_{i,k-1}\left(1 - b_{i,k-1}\right). \tag{B.10}$$

Note that in our fictitious IESA surrogate model of the cellular network, the Δ and Ω attributes of a request are reset to $\emptyset$ and 0, respectively, upon handover. This is because congestion information about the neighborhood set of one cell may be irrelevant to the neighborhood set of another cell. This creates the scenario where the amount of handover traffic to each BS with $\Omega = 0$ depends on the amount of overflow traffic in the network with $\Omega > 0$. Therefore, unlike previous applications of IESA such as video-on-de mand networks that do not contain handovers, fixed-point iteration is required to solve equations (B.1)–(B.10). A diagram of the relationships between (B.1)–(B.10) is provided in Fig. B.1. The initial values for the fixed-point iteration are $a_{i,r,0,0} \leftarrow \lambda_i/N$ with all other values initialized to zero. Finally, the total blocked or dropped traffic for requests with starting cell i (including handover requests) is

$$Z_i = \sum_{r=1}^{N}\sum_{n=1}^{N+1}\sum_{j=N}^{k} z_{i,r,n,j}, \tag{B.11}$$

and the overall non-completion probability for requests with origin cell i is defined as

$$B_i = (1-\theta)\frac{Z_i}{\lambda_i} + \frac{\theta}{N}\sum_{q\in\Gamma_i} B_q. \tag{B.12}$$

Acknowledgements

This work was supported by the Health and Medical Research Fund of Hong Kong (grant number 16171921), and the Research Grants Council (RGC) of Hong Kong under the General Research Fund (grant numbers 11104620, 11102421, and 11101422).

References

[1] A. Alnoman, A. S. Anpalagan, Computing-aware base station sleeping mechanism in H-CRAN-Cloud-Edge networks 9 (3) (2019) 958–967. doi:10.1109/TCC.2019.2893228.

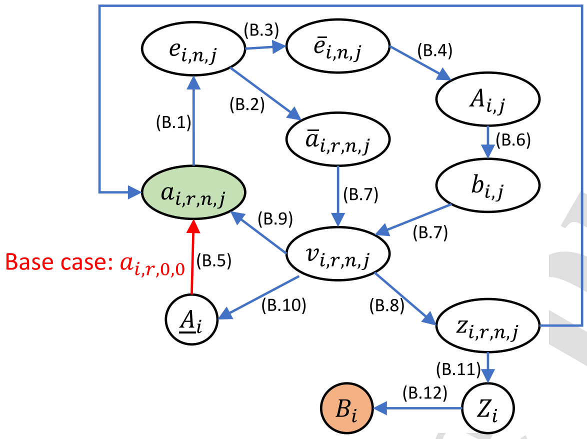


Figure B.1: Diagram showing fixed-point relationships between the various equations in the IESA algorithm (also applies to IESA-CN and IESA-NN). Numbers in parentheses refer to equation numbers in this paper.

[2] J. Wu, E. W. M. Wong, Y. Chan, M. Zukerman, Power consumption and GoS tradeoff in cellular mobile networks with base station sleeping and related performance studies, IEEE Trans. on Green Commun. and Netw. 4 (4) (2020) 1024–1036. doi:10.1109/TGCN.2020.3000277.

[3] J. Liu, T. Kou, Q. Chen, H. D. Sherali, Femtocell base station deployment in commercial buildings: A global optimization approach 30 (3) (2012) 652–663. doi:10.1109/JSAC.2012.120414.

[4] M. Dong, T. Kim, J. Wu, W. M. E. Wong, Millimeter-wave base station deployment using the scenario sampling approach 69 (11) (2020) 14013–14018. doi:10.1109/TVT.2020.3026216.

[5] W. Teng, M. Sheng, X. Chu, K. Guo, J. Wen, Z. Qiu, Joint optimization of base station activation and user association in ultra dense networks under traffic uncertainty 69 (9) (2021) 6079–6092. doi:10.1109/TCOMM.2021.3090794.

[6] T. Zhang, H. Li, S. Zhang, J. Li, H. Shen, STAG-based QoS support routing strategy for multiple missions over the satellite networks 67 (10) (2019) 6912–6924. doi:10.1109/TCOMM.2019.2929757.

[7] S. F. Abedin, M. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, C. S. Hong, Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network 67 (1) (2019) 489–502. doi:10.1109/TCOMM.2018.2870888.

[8] M. Jia, W. Liang, Z. Xu, M. Huang, Y. Ma, QoS-aware cloudlet load balancing in wireless metropolitan area networks 8 (2) (2020) 623–634. doi:10.1109/TCC.2017.2786738.

[9] F. P. Kelly, Blocking probabilities in large circuit-switched networks, Advances in Applied Probability 18 (2) (1986) 473–505. doi:10.2307/1427309.

[10] E. W. Wong, A. Zalesky, Z. Rosberg, M. Zukerman, A new method for approximating blocking probability in overflow loss networks, Computer Networks 51 (11) (2007) 2958–2975. doi:10.1016/j.comnet.2006.12.007.

[11] E. W. M. Wong, Y.-C. Chan, A century-long challenge in teletraffic theory: Blocking probability evaluation for overflow loss systems with mutual overflow, IEEE Access 11 (2023) 61274–61288. doi:10.1109/access.2023.3283803.

[12] E. W. M. Wong, J. Guo, B. Moran, M. Zukerman, Information exchange surrogates for approximation of blocking probabilities in overflow loss systems, in: Proc. International Teletraffic Congress (ITC), IEEE, 2013. doi:10.1109/itc.2013.6662932.

[13] J. Wu, E. W. M. Wong, J. Guo, M. Zukerman, Performance analysis of green cellular networks with selective base-station sleeping, Perform. Eval. 111 (2017) 17–36. doi:10.1016/j.peva.2017.03.002.

[14] C. P. Chen, Z. Liu, Broad learning system: An effective and efficient incremental learning system without the need for deep architecture, IEEE transactions on neural networks and learning systems 29 (1) (2017) 10–24.

[15] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1-3) (2006) 489–501. doi:10.1016/j.neucom.2005.12.126.

[16] D. R. B. de Araujo, C. J. A. Bastos-filho, J. F. Martins-filho, Methodology to obtain a fast and accurate estimator for blocking probability of optical networks, IEEE/OSA Journal of Optical Communications and Networking 7 (5) (2015) 380–391. doi:10.1364/JOCN.7.000380.

[17] H. C. Leung, C. S. Leung, E. W. M. Wong, S. Li, Extreme learning machine for estimating blocking probability of bufferless OBS/OPS networks, Journal of Optical Communications and Networking 9 (8) (2017) 682. doi:10.1364/jocn.9.000682.

[18] S. Li, H. C. Leung, E. W. M. Wong, C. S. Leung, Enhancement of extreme learning machine for estimating blocking probability of OCS networks with fixed-alternate routing, IEEE Access 7 (2019) 52319–52330. doi:10.1109/access.2019.2907752.

[19] Y.-C. Chan, E. W. M. Wong, C. S. Leung, Evaluating non-hierarchical overflow loss systems using teletraffic theory and neural networks 25 (5) (2021) 1486–1490. doi:10.1109/lcomm.2021.3052683.

[20] A. Chen, J. Law, M. Aibin, A survey on traffic prediction techniques using artificial intelligence for communication networks, Telecom 2 (4) (2021) 518–535. doi:10.3390/telecom2040029.

[21] D. R. de Araújo, C. J. Bastos-Filho, J. F. Martins-Filho, Methodology to obtain a fast and accurate estimator for blocking probability of optical networks, Journal of Optical Communications and Networking 7 (5) (2015) 380–391.

[22] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes 17 (4) (2006) 879–892. doi:10.1109/tnn.2006.875977.

[23] G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Transactions on Neural Networks 20 (8) (2009) 1352–1357.

[24] H.-T. Wong, H.-C. Leung, C.-S. Leung, E. Wong, Noise/fault aware regularization for incremental learning in extreme learning machines, Neurocomputing 486 (2022) 200–214.

[25] W. J. von Eschenbach, Transparency and the black box problem: Why we do not trust AI, Philosophy & Technology 34 (4) (2021) 1607–1622. doi:10.1007/s13347-021-00477-0.

[26] Y. Lan, Y. C. Soh, G.-B. Huang, Random search enhancement of error-minimized extreme learning machine, in: Proc. European Symposium on Artificial Neural Networks (ESANN), 2010, pp. 327–332.

[27] A. N. Njoya, C. Thron, M. N. Awa, A. A. A. Ari, A. M. Gueroui, Power-saving system designs for hexagonal cell based wireless sensor networks with directional transmission, Journal of King Saud University - Computer and Information Sciences 34 (10) (2022) 7911–7919. doi:10.1016/j.jksuci.2022.07.008.

[28] S. R. Das, G. K. Audhya, K. Sinha, Channel assignment in hexagonal cellular networks in presence of device-to-device communication, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2019. doi:10.1109/wimob.2019.8923294.

[29] F. E. Browder, W. V. Petryshyn, The solution by iteration of nonlinear functional equations in Banach spaces, Bulletin of the American Mathematical Society 72 (3) (1966) 571–575. doi:10.1090/S0002-9904-1966-11544-6.

[30] M. L. Thompson, M. A. Kramer, Modeling chemical processes using prior knowledge and neural networks, AIChE Journal 40 (8) (1994) 1328–1340. doi:10.1002/aic.690400806.

[31] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating scientific knowledge with machine learning for engineering and environmental systems, ACM Computing Surveys 55 (4) (2022) 1–37. doi:10.1145/3514228.

[32] A. Bousquet, W. H. Conrad, S. O. Sadat, N. Vardanyan, Y. Hong, Deep learning forecasting using time-varying parameters of the SIRD model for covid-19, Scientific Reports 12 (1) (feb 2022). doi:10.1038/s41598-022-06992-0.

[33] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data, IEEE Transactions on Knowledge and Data Engineering 29 (10) (2017) 2318–2331. doi:10.1109/tkde.2017.2720168.

[34] B. Huang, J. Wang, Applications of physics-informed neural networks in power systems - a review, IEEE Transactions on Power Systems 38 (1) (2023) 572–588. doi:10.1109/tpwrs.2022.3162473.

[35] J. Du, J. Zheng, Y. Liang, Q. Liao, B. Wang, X. Sun, H. Zhang, M. Azaza, J. Yan, A theory-guided deep-learning method for predicting power generation of multi-region photovoltaic plants, Engineering Applications of Artificial Intelligence 118 (2023) 105647. doi:10.1016/j.engappai.2022.105647.

[36] J. Du, J. Zheng, Y. Liang, N. Xu, Q. Liao, B. Wang, H. Zhang, Deeppipe: Theory-guided prediction method based automatic machine learning for maximum pitting corrosion depth of oil and gas pipeline, Chemical Engineering Science 278 (2023) 118927. doi:10.1016/j.ces.2023.118927.

[37] N. Wang, Q. Liao, H. Chang, D. Zhang, Deep-learning-based upscaling method for geologic models via theory-guided convolutional neural network, Computational Geosciences (Aug. 2023). doi:10.1007/s10596-023-10233-2.

[38] P. Sharma, W. T. Chung, B. Akoush, M. Ihme, A review of physics-informed machine learning in fluid mechanics, Energies 16 (5) (2023) 2343. doi:10.3390/en16052343.

Highlights

- We propose an improved method for efficient cellular network performance evaluation.
- Combining teletraffic theory with neural networks boosts accuracy and robustness.
- The neural network part improves accuracy over a pure teletraffic approach.
- The teletraffic part aids robustness when extrapolating to new parameter ranges.

Yin-Chi Chan: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - Original Draft, Writing - Review & Editing.

Jingjin Wu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - Review & Editing.

Eric W. M. Wong: Conceptualization, Methodology, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Chi Sing Leung: Conceptualization, Methodology, Writing - Review & Editing.

Declaration of Interest Statement

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Eric W. M. Wong reports financial support was provided by Hong Kong Health and Medical Research Fund. Eric W. M. Wong reports financial support was provided by Hong Kong Research Grants Council.