ARTIFICIAL NEURAL NETWORK SIMULATION FOR MARKOVIAN QUEUING MODELS

Sivakami Sundari M
Department of BSH, GIET University, Gunupur, India,
Email: senthilsivakami@gmail.com

Senthil Kumar K*
Department of Mechanical Engineering, GIET University, Gunupur, India.
Email: vecsenthil@gmail.com

Yamini. S
Department of Mechanical Engineering, GIET University, Gunupur , India,
Email:yaminimech23@gmail.com

Palaniammal S.
Principal , Sri Krishna Adithya College of Arts and Science, Kovaipudur, Coimbatore, India.
Email: splvlbresearch@gmail.com

Abstract - Successful ranking of a website by Google or electric charging of vehicle, congestion is pervasive in all domains. This implies that the presence of queues everywhere or in various places simultaneously. Under this environment, a good understanding of the relationship between queueing and delay is essential in the design of mathematical queuing models. However, uncertainty is an unavoidable phenomenon in any decision-making process. Good number of mathematical approaches has been presented in the literature to the analysis of queuing. Uncertainty is usually considered as unidimensional in nature that can be handled with probability theory. The objective of queuing analysis is to offer a reasonably satisfactory service to waiting customers. Queuing theory is not an optimization technique. Rather, it determines the measure of performance of waiting lines, such as the average waiting time in the queue and the productivity of the service facility, which can then be used to design the service installation. Assumed systems and systems that are too complicated to be disturbed are often difficult to study by analytical techniques. Simulation is one technique that can be seen successfully utilized for analyzing such systems. Artificial neural networks (ANN) form a branch of artificial intelligence. Neural networks represent a connection of simple processing elements capable of processing information in response to external inputs. In this work, such a Markovian queue is simulated using ANN and presented the result. The result shows that the ANN is capable of solving complex queuing problems.

Keywords: *Markovian Queue, ANN, Simulation.*

1. Introduction

In any process entities waiting for service are called queues. Queues are formed because resources are limited. Such queues are analyzed, modeled and solved mathematically to optimize the performance, thereby increasing productivity. In every queuing process, the objective of queuing model is how with minimum resources and waiting maximum output can be achieved and hence maximize productivity. Queues are formed due to demand for service by arrivals and arrivals are with large uncertainty. It is possible to describe the probability of events only if entities attained the state. Queues are classified based on the nature of arrival and service. Arrivals of many real-time problems follow with Poisson distribution and services exponential distribution. However, both are Markovian. The number of servers increased if the arrival demands based on the analytical calculation which is very cumbersome. With the power of computers recently such a process has been simulated using Artificial Neural Network.

2. Markovian Queuing Model

Queuing models involve arrival rate, service rate and the number of servers and their types to solve analytically. AK Erlang in 1913[1] first used the concept of Queuing theory and analyzed to study telephonic calls and Kendall[2]-[3] applied those concepts in various operations. Little's formula has been widely used in solving queues. Kendal used certain mnemonics to represent queues like A/B/C/D/E/F. Queuing problems solved with the concept of probability theory using random phenomena. Such a phenomenon appears due to uncertainty involved in the arrival and service process. With both arrival and service follow with Markovian it is represented as M/M/C: ∞. C in this representation meant for the number of servers and the capacity is infinity[4]-[6]. In this work, a process involving Markovian arrival and Markovian service with single and multiple servers of infinite capacity is modeled and analytically solved to generate data sets for ANN. Generated data sets classified into training data, validation data and testing data. And these data sets used for modeling ANN for analyzing Morkovian queues.

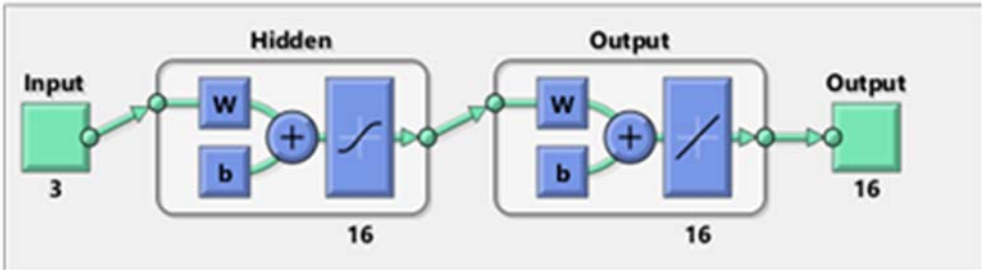

Figure 1 Architecture of an ANN model

3. Artificial Neural Network

Recently with the power of computers, ANN is very well used to solve complex problems [7]. ANN works based on the idea of the human neural system and successfully implemented in many applications. At the advent of machine learning, artificial intelligence, image processing, and virtual reality technologies ANN playing a critical role in the decision making process [8]. Basically ANN consists of three layers, an input layer, an output layer, and hidden layers. The input layer depends on input parameters generally used to solve the problems analytically or manually using mathematical equations [9]. And the output layer depends on the data arrived used a mathematical approach. Each layer is connected through neurons and the number of neurons on each layer depends on the number of input data and output data. Hidden layer neurons are arbitrarily set to model the ANN. Training and learning under supervised algorithms make these networks to appropriate tool to solve complex problems. Typical ANN developed is shown in fig.1

Table 1 Architecture of different ANN simulated

Model Developed	First layer Transfer function	Second layer Transfer Function
ANN1	Logsig	Pureline
ANN2	Pureline	Logsig
ANN3	Tansig	Logsig
ANN4	Logsig	Tansig
ANN5	Tansig	Pureline
ANN6	Pureline	Tansig

In this model as shown in fig.1 input layer is assigned with 3 neurons. Markovian queues are solved based on arrival rate, service rate and number of servers. Output layer consists of sixteen neurons based on the parameters on which the queues are analysed and is analytically calculated using Little's formula. However the number of neurons in the hidden layer is to be done on trial and error method [10].

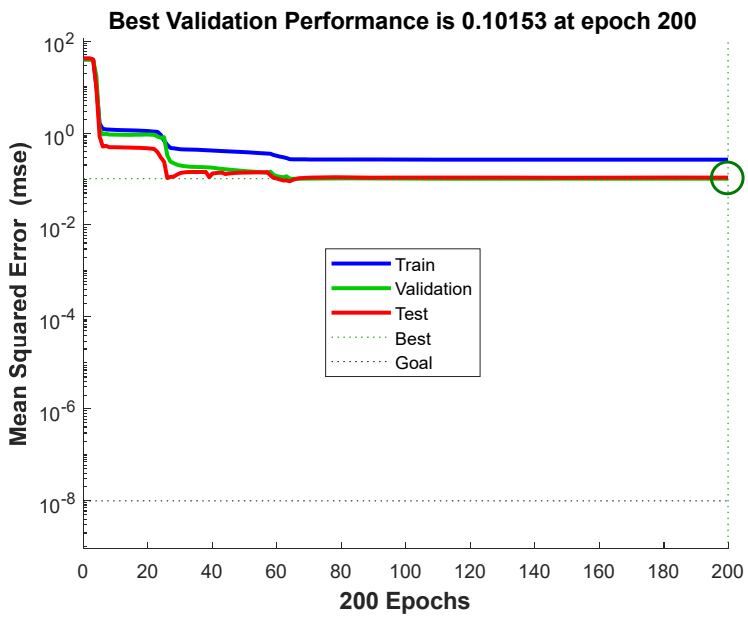

Figure 2 Training validation and testing curve of ANN4

ANN modeling is to be architected with suitable algorithm for better optimization. Which algorithm suits for the selected problem is again trial and error approximation [11]. In this work feed forward back propagation [12] algorithm is used for supervised learning. Training and learning functions plays a great role in simulation. There are different training and learning functions already available in the MATLAB platform which is used to train the ANN. Number of hidden layer is increased if the problem is highly complex and with single layer ANN along with different training and learning function not yielding expected accuracy. Transfer function between input layer to hidden layer and from hidden layer to output layer plays critical role in simulation and depends on the nature of the correlation between input data to output data combination of transfer function selected. Fig.1 shows combination of tansig transfer function in the first layer and pureline in the second layer. Table :1 shows the different ANN structure developed using combinations of transfer function with different number of neurons. Number of neurons in the hidden layer increased from one to sixteen and simulated using the same transfer function and checked for the simulation performance.

4. Result and Discussion

Fig.2 shows the training, validation and testing curve generated during the modeling. Whether the architecture of the ANN suits the problem or not could be decided based on the comparison of the performance of the simulations with different ANN models. The criteria for selecting the best ANN architecture depends on how faster the ANN converge to lower error value or meeting out the target error with minimum epochs. Fig.2 shows the training validation and testing curve of ANN4. It is observed from Fig.3 that within 60 epochs the ANN1 could reach the mean square error of 0.0843 wherein ANN4 could validate only up to 0.101 MSE and further iteration has no improvement in the accuracy. But ANN1 continue to converge with low MSE and increasing the epochs further the performance of the ANN is expected better. Based on this consideration, simulation performance of ANN1 shows better choice than ANN4.

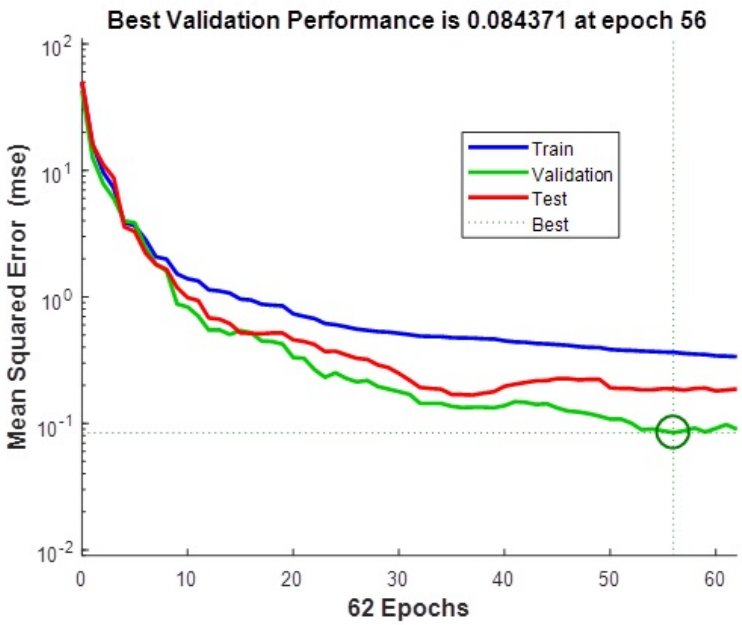

Figure 3: Training validation and Testing curve ANN1

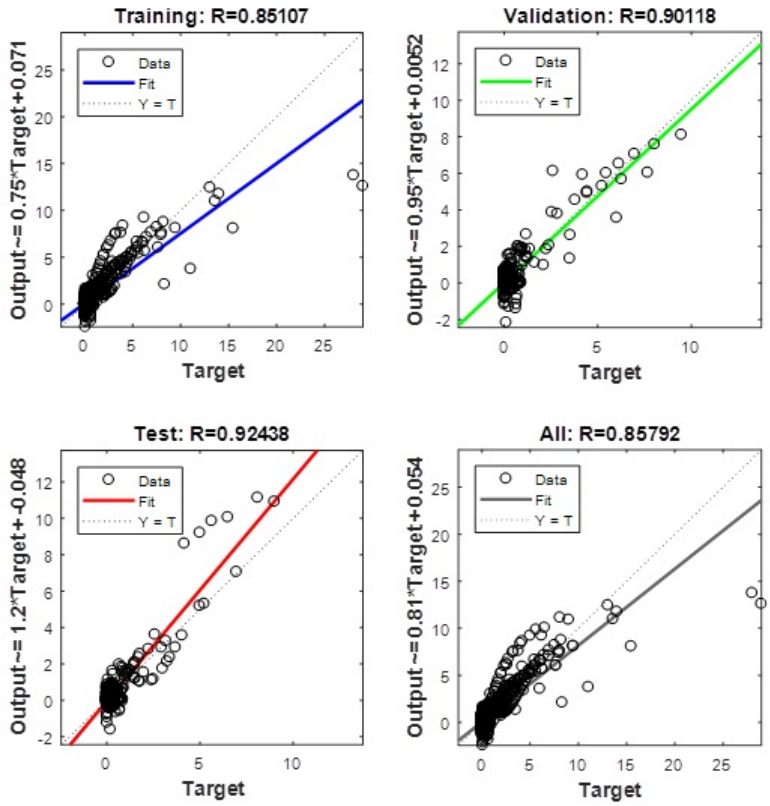

Figure 4 Regression analyses of training, validation and Testing ANN3

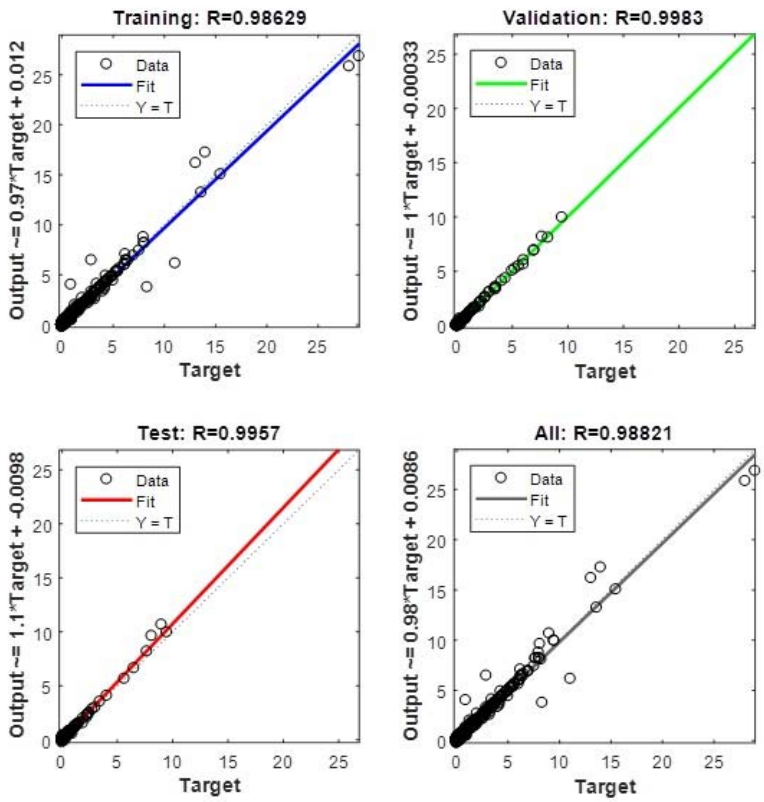

Figure 5: Regression analyses of training, validation and Testing ANN5

Regression analysis is a method used to predict the results of training, validation, testing and overall performance. The value R correlation coefficient lies between -1 and +1. R values closer to +1 specify a stronger positive linear relationship while R values closer to -1 indicate a stronger negative relationship. From the Fig.4 the overall R value is 0.85792 for the ANN3 developed. In all the process it is observed the the R value is not close to 1, which indicates that the simulated values are not matching with that of original test data which

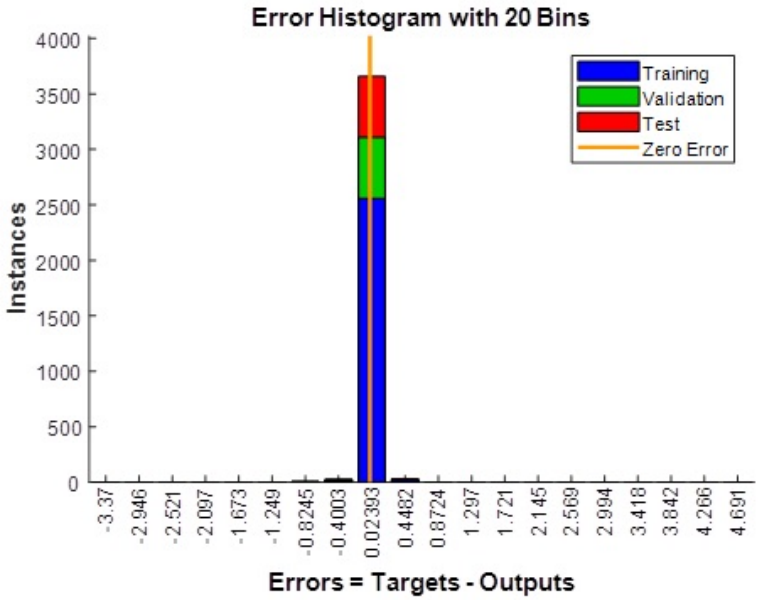

Figure 6 Error histogram of ANN5

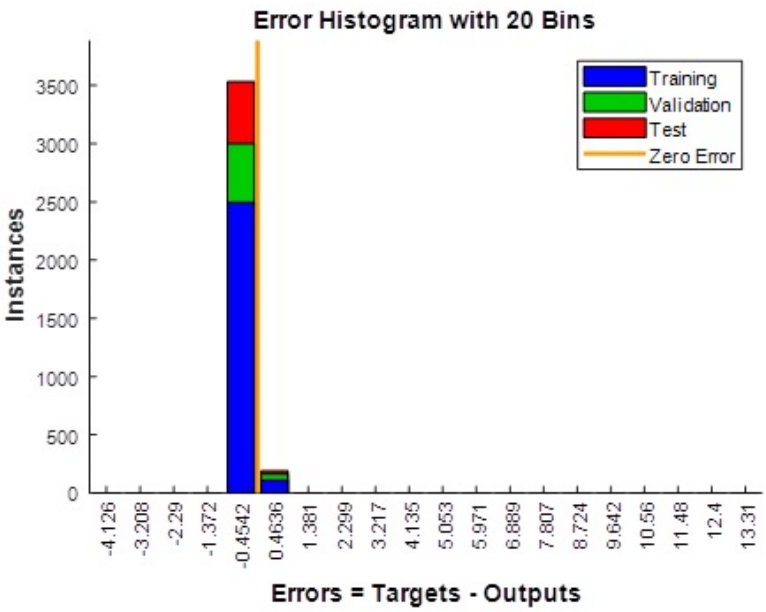

Figure 7 Error histogram of ANN3

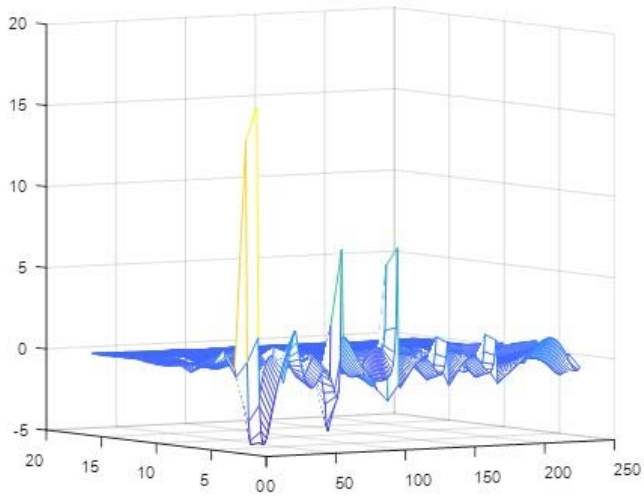

Figure 8 Error mesh diagram of ANN3

was obtained from analytical method. The supervised training outputs shows larger deviation and fit is not perfect which indicates the error generated is more and values obtained with error of nearly 25%. Fig.5 shows the regression analysis of ANN5. It is observed that the R value is very close to 1 and all three process shows best fit with target and output values. Though the training R value is low compared with validation and testing, ANN5 expected to yield better R value for testing data set.

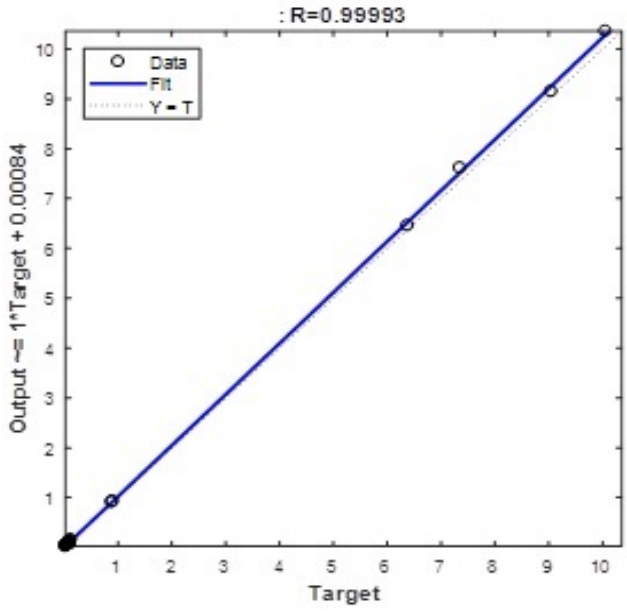

Figure 9 Regression analysis of test data comparison with target and output

Error histogram analysis is another tool to decide the accuracy level of the ANN simulated. From Fig.6 the error histogram developed it is observed that, for the entire data range the error value ranging between -3.37 to 4.691 and maximum instances occurred at zero error. Other error values instances are observed less instances and the maximum error found to be within the range ±0.4 in most of the instances. Fig.7 shows the error histogram of ANN3 and the error ranges between -4.1 to 13.3 which is higher than that of ANN5 which indicates that ANN5 perform better than the ANN3. The 3D mesh profile of the error values obtained is shown in the Fig.8. From the Fig.8 error analysis with respect to input data set given for simulation and output could be identified. Although histogram is useful in identifying the error concentration magnitude it is not telling what data is observed wrong and this could be observed using data mesh profile accordingly decision could be taken. Fig.10 shows the original output diagram and Fig.11 shows the simulated output mesh data diagram. By comparing profile of the original output and simulated output the selection of the ANN model for further simulation of unknown data can be simulated for output. Regression analysis of original and test data shown in Fig.9. R value 0.9999 means the simulation is exactly matching with output required and the error is minimum.

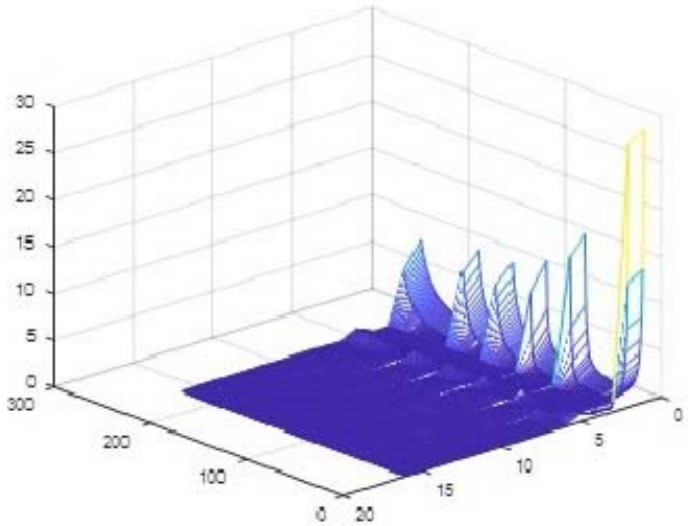

Figure 10 Mesh diagram of original output data

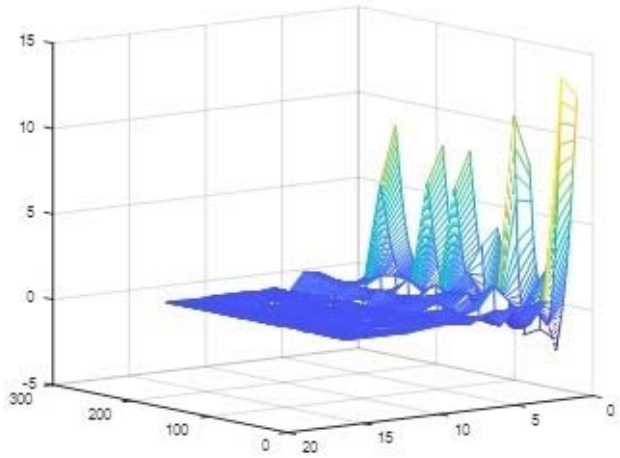

Figure 11 Mesh diagram of ANN simulated output data

5. Conclusions

It is seen that very good predictive accuracy is achieved by the neural network with the different training function of test patterns. Based on this successful performance, the neural network can be used to predict future simulation outputs without actually running the simulation process using the same architecture. This will facilitate prompt decision making in deciding number of servers and the queue performance as required for the process.

References

[1] Erlang, A.K (1917). Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephone Exchanges, Elektrotkeknikeren, vol 13.

[2] Kendall, D.G. (1951).Some Problems in theory of queues, Journal of the Royal Statistical Society(B) Vol. 13 No. 2, pp. 151-157 and 184-185.

[3] Kendall, D.G. (1953).Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain, Annals of Mathematical Statistics Vol. 24 No. 3, pp. 338-354.

[4] Abdelkader, Yousry H., Wohaibi, Maram Al. Computing the Performance Measures in Queueing Models via the Method of Order Statistics, Journal of Applied Mathematics, Volume 2011, Article ID 790253, 12 pages,

[5] Andriansyah, R., Van Woensel, T., Cruz, F.R.B., Duczmal, L (2010).Performance optimization of open zero-buffer multi-server queueing networks, Computers and Operations Research Volume 37, Issue 8, pp. 1472-1487

[6] Aras, A.K., Chen, X., Liu, Y. (2018).Many-server Gaussian limits for overloaded non-Markovian queues with customer abandonment. Queueing Syst 89, 81–125 .

[7] Aveklouris, A., Vlasiou, M.,Zwart, B. (2019).Bounds and limit theorems for a layered queueing model in electric vehicle charging, Queueing Syst Vol. 93, Issue 1–2, pp 83–137.

[8] Azadeh, A., Faiz, Z.S., Asadzadeh, S.M., Tavakkoli-Moghaddam, R. (2011).

[9] An integrated artificial neural network-computer simulation for optimization of complex tandem queue systems, Mathematics and Computers in Simulation, Vol.82, Issue 4, pp.666-678.

[10] Chambers, M., Mount-Campbell, C.A. (2002).Process optimization via neural network metamodeling, International Journal of Production Economics, Vol. 79, Issue 2, pp. 93-100.

[11] Emami, A., Sarvi, M., Asadi Bagloee, S. J. (2019).A neural network algorithm for queue length estimation based on the concept of k-leader connected vehiclesJournal of Modern Transportation , Volume 27, Issue 4, pp 341–35.

[12] Hagan, M. T., Menhaj, M. (1994).Training feedforward networks with the Marquardt algorithm, IEEE Transactions on Neural Networks, Vol. 5, No.6, pp. 989- 993.