Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 135 (2018) 35–42

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Computer Science and Computational Intelligence 2018

Waiting-Time Estimation in Bank Customer Queues using RPROP Neural Networks

Rinda Parama Satya Hermanto[a], Suharjito[a] *, Diana[b], Ariadi Nugroho[a]

[a]*Computer Science Department, BINUS Graduate Program- Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*
[b]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

In daily banking customer queues, unknown waiting-time could lower customer experience. Little's Law formula in Queue Theory provides a generic formula for waiting-time, but it cannot be implemented directly to give finite wait-time estimation in real-life. This study aims to investigate predictive variables that explain waiting-time duration. This paper uses Fast Artificial Neural Network engine to implement Artificial Neural Networks method. To train Artificial Neural Networks, Resilient Propagation was used. Time-series approach and structural approach for input neuron was compared. Average duration from previous interval and number of server was proposed to increase structural variable like Queue Length and Head of Line Duration estimator variable. To determine the best configuration for number of neuron in input and hidden layer, experimental method was used. The results of this study show that structural approach provides better estimation than time-series approach. Furthermore, modified helper variable combination provides a more refined result.

Keywords: waiting-time; estimation; neural networks; time-series; queue

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: suharjito@binus.edu

10.1016/j.procs.2018.08.147

1. Introduction

Queuing is part of our everyday life. Sub-optimal processing of banking inquiries that cannot keep up with customer arrival rate will lead to long queues in branches. Random arrival distribution also contributes to the formation of long queues. Despite the importance of managing queues, waiting-time duration in queues is often difficult to estimate. Although stated in the in the psychology of waiting: "Principle 4: Uncertain Waits are Longer than Known, Finite Waits." [1], known finite waits still difficult to achieve. To determine finite wait, some studies started to estimate waiting-time in order to increase customer experience in queues.

Automated queue management system is widely used in the industry, including banking. However, most of these queue management systems do not have waiting-time estimation features. In this study, we use Mattel Queuing System (MQS) data log, one of computerized queue management system used in Indonesian banks. To estimate wait-time in queues, we need to refer to the Queuing theory. It is hard to implement queuing theory directly. Arrival rate and service rate cannot be acquired from queue management system. For wait-time estimation in this study, we need to generate variables by using simulation. Those variables are possibly used in real-life wait-time estimation.

A study to investigate wait-time in order to achieve quality of service was done by [2]. They calculated the probability of customer overflow by utilizing Little's Law. A significant study on wait-time estimation was published by [3]. Their methodology was based on statistical time-series forecasting. Unlike common time-series data, their collected data was not uniform nor equally spaced. Therefore, they need to fill missing data by using regression and average value for every corresponding 10 minutes interval. Their research showed a decrease in the mean absolute error of their service to less than 2-3 minutes. Another study was done by Kim and Whitt to estimate wait-time using time-varying little's Law (TVLL) [4]. They reduced bias on Little's Law. A similar study was conducted by Ibrahim and Whitt to predict wait-time for customer service. The authors utilized queue length (QL) and the elapsed waiting time of the customer at the head of the line (HoL) as predictor [5]. Their study used a parametric approach by deriving a formula to predict wait-time. As a result, several modified predictors based on those two predictors were tested.

There have been many studies on the usage Artificial Neural Networks (ANN) to estimate time-series data and prediction. Guresen Use expert systems with ANN to model time-series data, in particularly stock market data [6]. He describes this method as one of the best ways to model stock market, because it does not contain standard formulas and can easily adapt to changes in the market. A similar study on applying ANN to predict stock price showed satisfactory results as they achieved the objective to predict the closing price, following its behavior and tendency [7]. A comparative study discovered the fact that Neural Networks outperform Statistical technique in forecasting [8]. Another work to model the performance of an ANN with QUICKPROP and RPROP (improved version of traditional Back Propagation algorithm) on time-series forecasting. The study showed that the approach can generalize well and can be used in real problems [9]. The aforementioned studies show the applications of ANN to estimate waiting-time as series data in banking queue.

A related study in predicting waiting time in bank teller queues using various learning methods was conducted by [10]. The authors developed and tested four prediction models using Queuing Theory's formula, Deep Learning (DL), Gradient Boost Machine (GBM), and Random Forest (RF) respectively. Out of the four models, GBM model was found to be most efficient with 97% accuracy and an F1 measure of 75%. In this study, we use ANN to predict waiting time mainly for a reason of practicality and familiarity with the approach. We are aware that there are multiple methods that can be used to build data-driven prediction models as shown in the study conducted by [10]. Nevertheless, the objective of this paper is not to compare the effectiveness and accuracy of different prediction methods, but rather to investigate significant predictors of waiting time by utilizing RPROP Neural Networks.

2. Research Method

In this study, a helper tool called Wait-Time Estimator is developed to study ANN usage in wait-time estimation. The overall architecture of Wait-Time Estimator is shown in **Error! Reference source not found.**. This tool use Fast Artificial Neural Network Library (FANN) [11] as a core engine to implement ANN. To handle UI and front-end data flow, Wait-Time Estimator uses AIR Application. Having the tooling implemented, it will be easier to replicate this study in the future in a real life queuing system. AIR Application communicates with FANN and reads queue data warehouse via ANE (AIR Native Extensions). As data source for this study, Matel Queuing System (MQS)'s data log was used. This queuing system handled M/M/c queuing model for banking's customer service. In queue simulation, time-varying service was detected in the data log. Matel is one of banking queue management system commonly used in Indonesia.

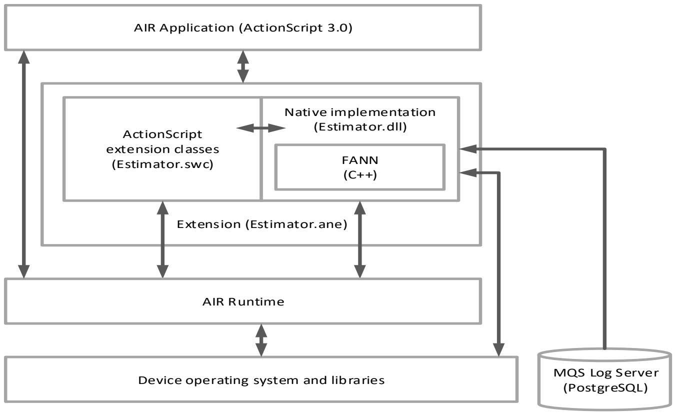


Fig. 1. Waiting Estimator Tool Architecture

2.1. ANN with FANN

In this study, Resilient Propagation (RPROP), a more advanced derivation of the dominating back-propagation training algorithm is used [11]. The essential advantage of the RPROP algorithm is based on the process of the ANN weights update (ANN training), where only the sign of error function gradient, instead of the value of the error, is utilized [12]. This modification of the ANN weight in the training process is done according to equation (1).

$$\Delta_{ij}^{(t)} \begin{cases} \eta^{+} * \Delta_{ij}^{(t-1)}, & if \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^{-} * \Delta_{ij}^{(t-1)}, & if \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)}, & else \end{cases} \quad (1)$$

Where $0 < \eta^{-} < 1 < \eta^{+}$

$\Delta_{ij}^{(t)}$ states individual weight update from neuron j to neuron i. t Defines iteration update from iteration $t-1$. Verbalized, every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_{ij} is decreased by the factor η^{-}. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions [12]. For implementation in this study, RPROP is supported by Fast Artificial Neural Network Library (FANN). FANN is a free open source neural network library [11] which implements multilayer artificial neural networks in C with support for both fully connected and sparsely connected networks. FANN is chosen because its libraries have been around since 2003 and it has been widely used in both research settings.

In this study, all ANN uses sigmoid activation function and activation steepness 0.5. Architecture used is multilayer network consists of input layer, hidden layer, and output layer. Single hidden layer is used with some variations on hidden neuron number. Time series approach will use previous consecutive time-series data as input neuron which will be tested in the experiment.

2.2. Data Collection

Data used in this study is a real industrial data gathered from one of the big banks in Indonesia. The bank has implemented a Queue Management System called Mattel Queuing System. We collected logs data containing waiting time across tellers. There are 22.082 data points obtained, of which 65% and 35% are used for training and testing respectively. This study compares two different approaches: time-series approach [3] and structural approach. In structural approach, input layer was determined by queue's environment condition variable.

2.3. Time-series approach

Bank customer Queuing data consist of customer data array, marked by their arrival time and wait-time duration (D). Every customer came in a random arrival time. Therefore, queue's waiting-time cannot be applied directly as time series data. It is different from stock data where we can take a value at a certain point in time. Wait-time is considered as *flow time-series* where its activities over a given period must be summarized. This is analogous to summarizing the number of new vehicle sales in a month as the sum of all new motor vehicles sold during each day of the month. We use 10 minutes as interval adapted from [3].

When it comes to estimate a new customer waiting-time, we cannot simply take previous customer's waiting-time as input data for time-series approach. Previous customer's waiting-time might not have recorded in the system, as shown in **Error! Reference source not found.**, because they are still in queue when the new customer arrives. To solve this, Wait-Time Estimator needs to estimate the previous interval's waiting time first.

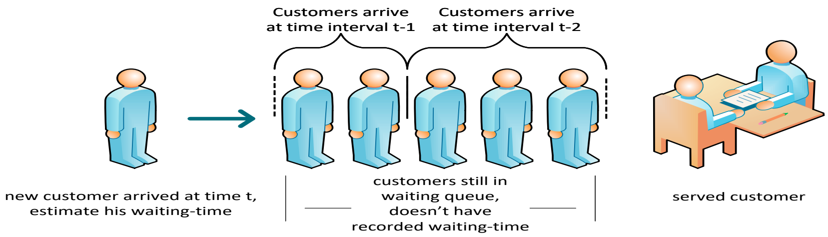


Fig. 2. Problem unrecorded series data input

To accommodate this solution, *iterative ANN* [13] as shown in **Error! Reference source not found.** is needed. The output of this function will be used in another estimation as an input neuron. Thus, basically, unrecorded interval will be filled with an estimation result of the corresponding previous interval using the same estimation method recursively.

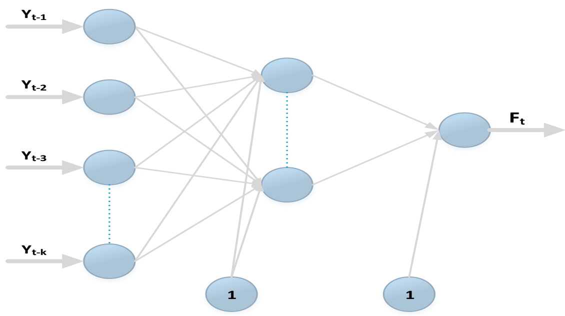


Fig. 3. Iterative ANN architecture for time-series approach

2.4. Structural approach

Structural approach for ANN's input in waiting-time estimation is easier than the time series approach. Structural approach uses the recorded value as an input neuron. However, in this study we have partial access to the required recorded values because our log data only record arrival time and waiting time of a customer. Queue length (QL) and number of server (c) are not recorded. Therefore, before using this approach, we need to generate QL data and number of server's data through a simulation. In this simulation, queuing flow will be simulated by using all recorded customer data. Thus, Wait-Time Estimator can count QL and c at every arrived customer in the virtual queues.

- *Queue Length (QL)*

  To generate QL data, every arrived customer will be counted in d length variable. This variable will record QL and reset to 0 when the simulation reaches another day. This process is illustrated in **Error! Reference source not found.**(a).

- *Number of Server (c)*

  Because our study was on M/M/c queuing model, we utilize service server number (c) to be the input for the estimator. Simulation process to generate the number of servers in this study is similar with QL process. The difference is Waiting-Time Estimator tool hold the list of server pool for every arrived customer. Then for every arrived customer, Wait-Time Estimator counts the current number of servers in this pool list. However, before this counting, idle servers (not serving a customer) in server pool list will be deleted. In other words, inactive server will be discounted. This process is illustrated in Fig 4(b).

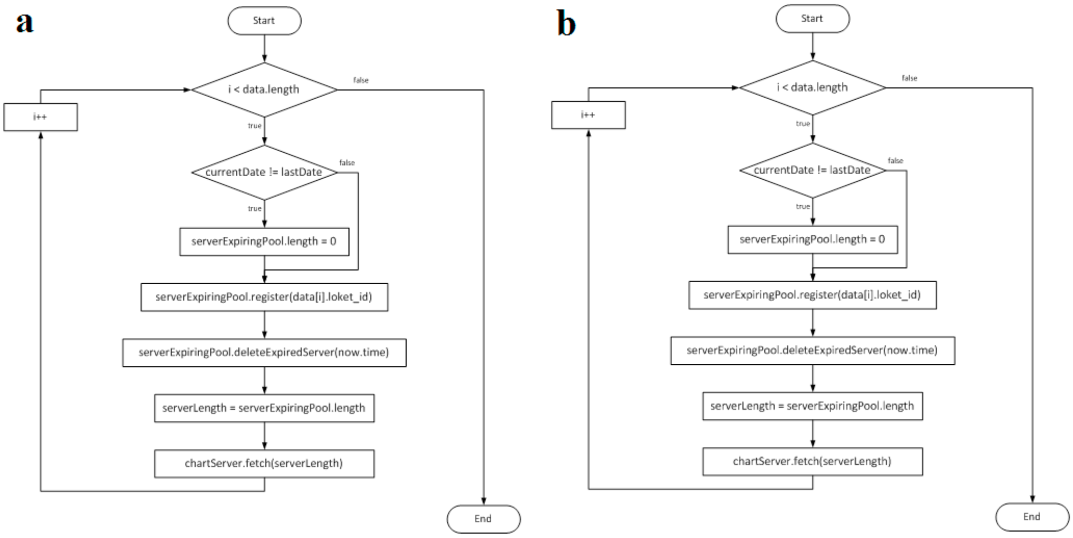


Fig. 4. (a) QL calculation process in simulation ; (b) Number of server calculation process in simulation.

- *Head of Line (HoL)*

  In this study, we use elapsed waiting time of the customer at the head of the line (HoL) based on Ibrahim's study [5]. In a customer queue, HoL is illustrated by Fig 5. Just like QL, HoL data does not exist in the log data for this study. Therefore, a similar simulation which generates the required data was conducted.

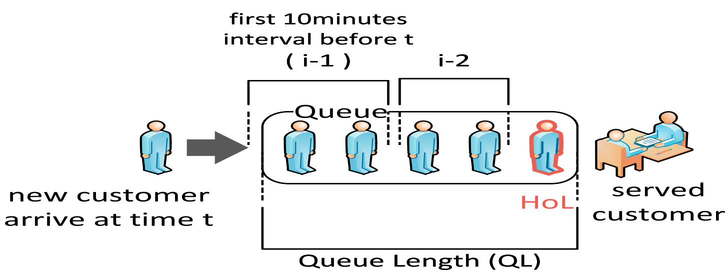


Fig. 5. Head of Line (HoL) in a customer queue

In Fig 5, we can see that we just need QL to find HoL from arrived customer at time t. This mean HoL data of customer t is customer t's arrival time subtracted by customer t-QL's arrival time.

3. Results and Analysis

In this section, we compare time-series approach and structural approach. Before that, we need to find the best configuration for time-series approach. To seek the best number of input neuron configuration, we need to state the number of hidden neuron first. Initially we use 17 hidden neurons, based on the best result reported in [14]. Other criteria will also be used to determine the correct number of hidden neurons to use, namely the most common rule-of-thumb [15]. We also use n+1 as a criterion, where n is the number of input.

There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following:

- The number of hidden neurons should be in the range between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

This study uses experimental methods to choose number of input neuron. We started to use nearest interval from predicted series: 1, 2, 3, 4, and 5. We then added 10, 20, and 30 to check higher number of input neuron's result. **Error! Reference source not found.** shows the result which roughly demonstrates that larger number of input neuron will generate higher error. To assess the error rate of the predictions we use Mean Squared Error (MSE), which measures the squared mean difference between the predicted waiting time and the actual waiting time. As such, MSE is a non-negative and values closer to zero are better. Note that the values of MSE are unites – the square root of SME test (RMSE) will indicate values measured in seconds.

As shown in Fig 6, a higher MSE group is cut to see the difference with the lower groups. The result shows that 3 inputs with rule-of-thumb, 2 inputs with n+1 hidden, and 4 inputs with n+1 hidden is the lowest among the results. Those best results will be compared to another Wait-Time Estimator's predictor variables, such as QL and HoL

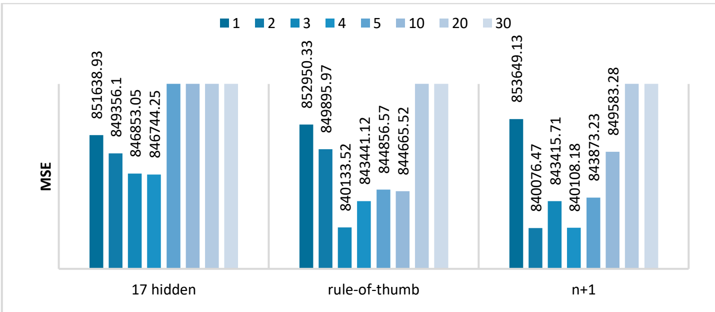
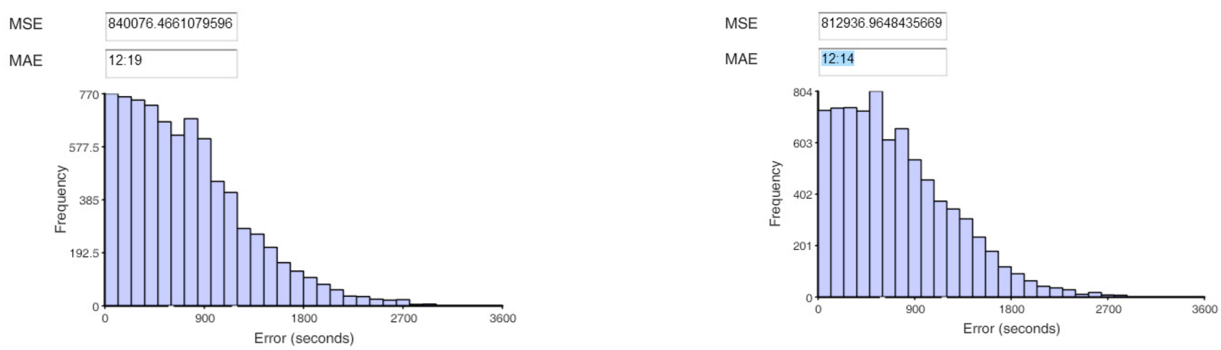


Fig. 6. Time-Series Approach's lower MSE Comparison Histogram

For a comparison between time-series approach and structural approach, we show the distribution error in histograms in Fig 7. It indicates that both approaches have higher density on lower error. Structural approach has better MSE 812936.96 (QL, c, HoL) compared to the series approach with MSE 840076.47 (2 input, n+1 hidden). However, our combined approach (D_{i-7}, QL, c, and HoL) has a slightly better result with MSE 811858.59.

a b



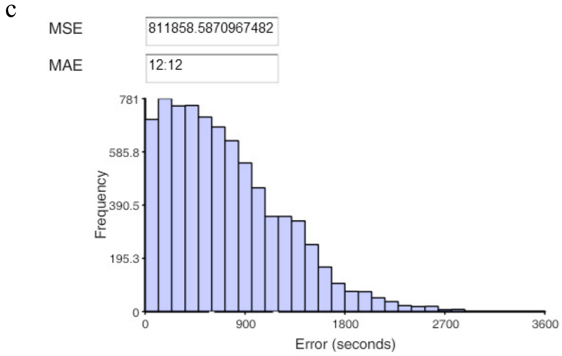


Fig 7 MSE histogram of best approach (a) time-series approach (b) structural approach, and (c) structural approach with helper (Di-7, QL, c, and HoL)

Fig 8 shows all MSE result comparison across all combinations. Time-series approach shows a moderate result, while the structural approach shows a better result. With the addition of time-series imitative variable, the structural approach has improved its result to be the most superior. This best result is shown by Di-7, QL, c, and HoL as an input layer for Wait-Time Estimator.

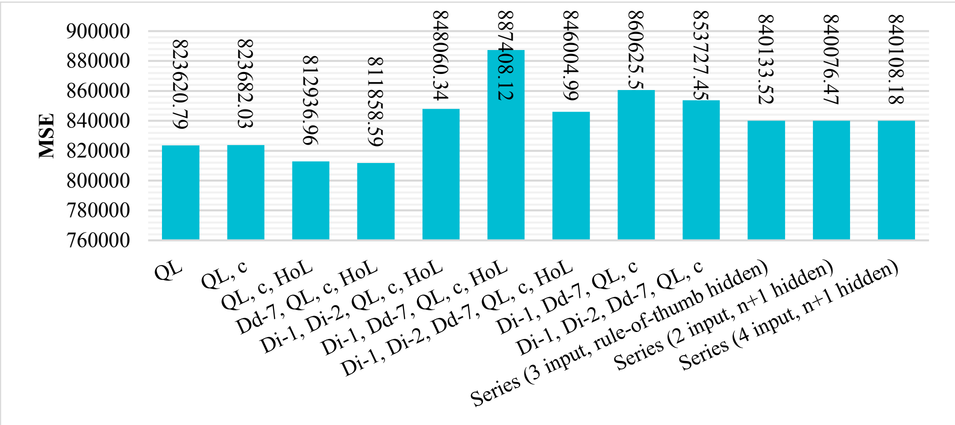


Fig. 8. MSE Result Comparison

Comprehensive awareness of environment variable makes ANN easier to perform an estimation. However, complex queues have more variables to be considered. With little help of past wait-time factor like D_{d-7}, the accuracy of an estimate can be increased.

4. Conclusion & Future Work

In this study, we adapt time-series approach from a previous study conducted by Bulut et al. [3] to estimate bank's customer queue wait-time. We use the same method to convert queue data as time-series data. Artificial Neural Networks is used to evaluate multiple predictor variables. With Resilient Propagation NN, the training process goes faster. Result shows that time-series approach is considered good to recognize wait-time pattern for this estimation. We also compare the result with a structural approach using predictors of Queue Length (QL), Number of Server (c), and Head of Line (HoL). A helper variable (D_{d-7}) also added in this study to be considered.

In this study, the structural approach result (QL, c, HoL) outperforms the time-series approach. Furthermore, the use of imitative time-series approach variable (D_{d-7}) makes the result even better. The improved result is due to increased predictive capability after accounting for wait-time in the same day the week before.

Both time-series approach and structural approach have good pattern recognition for wait-time estimation, but for banking customer queue with complex causative factor, further research is needed in the future. Queue Length, Number of Server, and Head of Line factors should be recorded for this purpose. Additionally, the type of services should be considered to estimate service rate variable in queue theory.

References

1. Jensen KB. The Psychology of Waiting. In ; 2011: Best Practices Inc.
2. Patel JJ, Chaudhary RM, Patel PA, Makwana P. Minimize the Waiting Time of Customer and Gain more Profit in Restaurant using Queuing Model. International Journal of Engineering Research and Applications (IJERA). 2012; 2(3): p. 1422-1425.
3. Bulut MF, Yilmaz YS, Demirbas M, Ferhatosmanoglu N, Ferhatosmanoglu H. Lineking: Coffee shop wait-time monitoring using smartphones. IEEE Transactions on Mobile Computing. 2015; 14: p. 2045-2058.
4. Kim SH, Whitt W. Estimating Waiting Times with the Time-Varying Little’s Law. Probability in the Engineering and Informational Sciences. 2013; 27(04): p. 471-506.
5. Ibrahim R, Whitt W. Wait-Time Predictors for Customer Service Systems with Time-Varying Demand and Capacity. Operations Research. 2011; 59(5): p. 1106-1118.
6. Guresen E, Kayakutlu G, Daim TU. Using Artificial Neural Network Models in Stock Market Index Prediction. Expert Systems with Applications. 2011; 38(8): p. 10389-10397.
7. de Oliveira FA, Nobre CN, Zárate LE. Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index--Case study of PETR4, Petrobras, Brazil. 2013; 40: p. 7596-7606.
8. Maliki OS, Agbo AO, Maliki AO, Ibeh LM, Agwu CO. Comparison of Regression Model and Artificial Neural Network Model for the Prediction of Electrical Power Generated in Nigeria. Advances in Applied Science Research. 2011; 2(5): p. 329-339.
9. Su´arez RR, Graff M. Modeling ANNs Performance on Time Series Forecasting. Power, Electronics and Computing (ROPEC), 2013 IEEE International Autumn Meeting on. 2013: p. 1-6.
10. Carvalho RS, Carvalho RN, Ramos GN, Mour�o RN. Predicting Waiting Time Overflow on Bank Teller Queues. In 16th IEEE International Conference on Machine Learning and Applications (ICMLA); 2017; Mexico: IEEE. p. 842-847.
11. Nissen S. Implementation of a Fast Artificial Neural Network Library (FANN). Report, Department of Computer Science University of Copenhagen (DIKU). 2003; 31: p. 29.
12. Riedmiller M, Braun H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In Neural Networks, 1993., IEEE International Conference on; 1993: IEEE. p. 586-591.
13. Hamzaçebi C, Akay D, Kutay F. Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. Expert Systems with Applications. 2009; 36(2): p. 3839-3844.
14. Valipour M, Banihabib ME, Behbahani SMR. Comparison of the ARMA, ARIMA, and the autoregressive artificial Neural Network Models in Forecasting the Monthly Inflow of Dez Dam Reservoir. Journal of Hydrology. 2013;: p. 433-441.
15. Panchal FS, Panchal M. Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network. International Journal of Computer Science and Mobile Computing. 2014; 3(11): p. 455-464.