I.J. Intelligent Systems and Applications, **2021, 3, 11-22**
Published Online June 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2021.03.02

Point Based Forecasting Model of Vehicle Queue with Extreme Learning Machine Method and Correlation Analysis

Kasliono
Master of Computer Science, Universitas Gadjah Mada, Yogyakarta, Indonesia
E-mail: kasliono@gmail.com

Suprapto and Faizal Makhrus
Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia
E-mail: sprapto@ugm.ac.id, faizal_makhrus@ugm.ac.id

Received: 19 November 2020; Revised: 10 January 2021; Accepted: 19 February 2021; Published: 08 June 2021

Abstract: Traffic is a medium to move from one point to another. Therefore, the role of traffic is very important to support vehicle mobility. If congestion occurs, mobility will be hampered so that it gives influence to other sectors such as financial, air pollution and traffic violations. This study aims to create a model to predict vehicle queue at the traffic lights when its status is red. The prediction is conducted by using Neural Network with Extreme Learning Machine method to predict the length of the vehicle queue, and Correlation Analysis was used to measure the correlation between the connected roads. The conducted experiments use data of the length of the vehicle queue at the traffic lights which was obtained from DISHUB (Transportation Bureau) DI Yogyakarta. Several experiments were carried out to determine the optimum prediction model of vehicle queue length. The experiments found that the optimum model had an average MAPE value of 15.5882% and an average Running Time of 5.2226 seconds.

Index Terms: Forecasting, Correlation, Traffic, Vehicle queue, Extreme leaning machine (ELM), Neural network, Correlation coefficient.

1. Introduction

The problem of traffic congestion is a problem that needs special attention in big cities in Indonesia [1]. Congestion is a condition in which the flow of traffic that passes on the road segment is approaching 0 km / h and causing a queue. The losses suffered as a result of this congestion problem are very large when quantified in monetary units, namely losses due to the longer travel time, the greater vehicle operating costs and the increase of vehicle pollution [2]. In traffic jams the vehicles move at a very low speed hence the fuel consumption becomes very wasteful and inefficient. In such traffic conditions, drivers tend to be impatient which leads to doing undisciplined actions which are worsen the traffic conditions further.

Accordingly, if the drivers/riders know beforehand about the time when a particular road is congested, or the length of the vehicle queue at the traffic lights increased, then drivers/riders can avoid this particular road and choosing another alternative road. Therefore, the development of forecasting model is needed to predict the length of the vehicle queue at the traffic lights and see the correlation between the connected roads. To this end, the artificial neural network (ANN) with the ELM method and correlation analysis can be considered as the optimal solution to achieve this goal.

Artificial neural network (ANN) is one part of the branch of artificial intelligence (AI) that mimics the work patterns of neural networks in living things in the process of solving problems. Various methods have been developed in ANN and one of the methods developed is the Extreme Learning Machine (ELM). The ELM is an advanced learning method (Feedforward) that was developed by utilizing matrix theory in Mathematics. ELM offers learning solutions with relatively fast learning time on artificial neural networks [3]. ELM has a simple algorithm, very fast learning capability and training small error [4]. Correlation analysis attempts to measure the close relationship between two variables by using a number called the correlation coefficient [5]. The linear correlation coefficient is defined as the size of the linear relationship between two random variables independent (X) and dependent (Y) which is represented by r. So, r is a measure of the extent to which points cluster around a straight line. [6].

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

2. Related Work

Surjandy Anindra, Soeparno and Napitupulu [7] analyzed congestion in the pejompongan area (Jakarta) using the Cased-Based Reasoning (CBR) approach. The congestion data obtained from CCTV cameras via the website viamana.com. In this study, with the CBR algorithm, there are three cases to describe the conditions on the road, namely empty (low traffic), smooth (medium traffic) or solid (high traffic). The image files captured from CCTV (as case input) will be converted to binary and matched the binary pattern (dB CBM). When the case is low, which means that there is no congestion, then the traffic light is in a normal condition (for example, the green light is on for 90 seconds). In the medium case, it means that the traffic condition is under control then the traffic light is as usual (for example, Greenlight for 60 seconds). However, for the high case, when the traffic is high, it means a green light for only 30 seconds. The output of this result can be passed on to traffic management.

Chairan and Martiningtyas [8] created a system that will regulate the green time at each intersection that has CCTV using video processing and background subtraction methods based on the area covered by vehicles at the intersection area. The system will also discharge segments if important vehicles are passing by, where the emptying of these sections for important vehicles is done using infrared sensors. The test results with the background subtraction method show that the green time is around 8 seconds when traffic is quiet and around 40 seconds when in a solid-state. However, when the sensor detects an important vehicle, a green light will be prompted for the section that the important vehicle traversed when the sensor detects an infrared signal.

The other research conducted by Mahatmaputra, Permata, and William [9] was to detect congestion using a modified Pin Hole algorithm and apply it to highways by using an IP camera for data retrieval. An evaluation performed to detected traffic jams in 78 different cases. The Traffic jam detection trial resulted in 55 accurate detection results with a total of 78 detections. So, the 70% level of accuracy of the detection results is obtained. Wibawa, Arida, and Rahning [10] also did the same thing to detect congestion by providing a mechanism to check road conditions at a particular location in Bali, which was previously manual to automatic. "Moving Object Detection" and "Background Subtraction" methods used to detect vehicle objects on the highway automatically through digital video media. Moving Object Detection (MOD) works by analyzing the movement of objects on digital video, and Background Subtraction applied to analyzes the movement by applying a wedge operation to the movement of the image with the background image.

Setiadi and Yeffry [11] designed a system to detect traffic density and traffic conditions using image processing and background subtraction methods which apply to separate objects from the background. As an implementation of the system in this study, a traffic light controller simulation is made and expected to be used to a larger system, namely the Traffic Control and Monitoring System (SIKOMOLINTAS). In this study, the results that have been achieved from the system design are producing data on density and traffic status, controlling the traffic light simulation timers based on existing density data, and integrating with SIKOMOLINTAS. On the other hand, the research on traffic density conducted by Wibowo and Soesanti [12] analyzed by forecasting the number of motorized vehicles during the day with time series data analysis to predict the number of motorbikes and VCR analysis to classify the road conditions. Based on time series data analysis using the trend method, the number of motorbikes passing Jalan Adisucipto at 12.30-14.40 tends to decline. Based on VCR analysis, the number of vehicles peaked at 15.30-17.00 and were included in class F (very bad), namely forced current conditions, low speed, volume overcapacity and long queues (jammed).

The Research conducted by Atmojo, Pulungan and Syahputra [13] regarding forecasting focused on the forecasting model for the needs of resellers using the Extreme Learning Machine (ELM) method aimed at prototyping the Intelligent Warehouse Management System (IWMS), especially in the Intelligent Forecasting System sub-system. (IFS).

3. Methods

3.1. *Artificial Neural Network (ANN)*

Artificial neural networks were first introduced by McCulloh and Walter Pitts in 1943 [14]. Artificial neural networks are information processing systems that have characteristics similar to biological neural networks. Artificial neural networks that have been trained are expected to have good generalization capabilities, generalization capabilities show the ability of artificial neural networks to produce the correct output (close to true) to enter from test data taken by the same population as training data but has never been used before in the process training [15].

3.2. *Extreme Learning Machine (ELM) Algorithm*

The ELM method was created to overcome the weaknesses of feedforward artificial neural networks, especially in learning speed because it uses a gradient descent algorithm for the learning process. Gradient descent learning algorithm requires iteration to find the weight values on the right network that requires a long training time. Some parameters needed for training, such as the pace of learning and stop conditions, must be determined precisely through several trials to get good training results. The selection of learning rate values that are too small will require a long time to reach convergence, but if the value is too large causing the algorithm to become unstable, improper stopping conditions can also cause over-fitting conditions that cause the ability of network generalization to decrease [16].

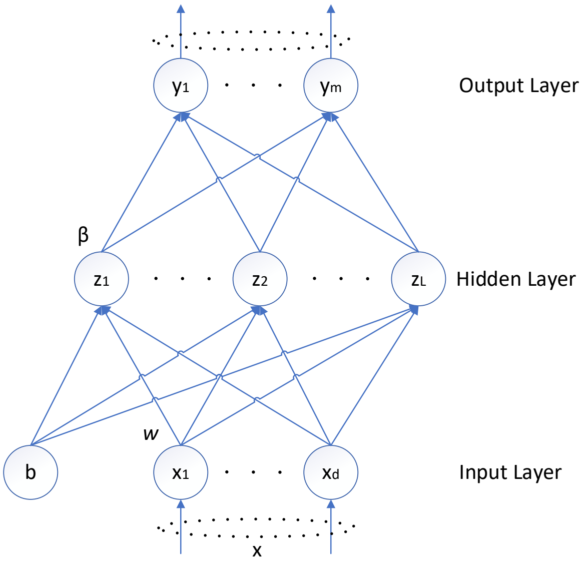


Fig.1. Single hidden layer feedforward network architecture [17]

ELM can conduct training in a much faster time (extreme learning) compared to the decent gradient algorithm because it does not require iteration. The parameters on the hidden node (weights and biases) can be determined at the beginning randomly and the value does not need to be changed during training through an iteration process. Furthermore, the weights from the hidden layer to the output layer are determined analytically using the inverse moore penrose matrix. ELM is also able to produce good generalization capabilities because ELM does not only try to minimize errors, but also norm of weight $||\beta||$ [3,17].

Huang et al. (2011) [17] developed an ELM for Generalized Single Hidden Layer Feedforward Network (SLFN), so that ELM can be applied to various types of SLFN. Fig. 1 shows the SLFN generalized architecture. The SLFN consists of d nodes in the input layer, L nodes in the hidden layer and m nodes in the output layer.

The data used for SLFN training consists of pairs of input and output data $(\boldsymbol{x_i}, \boldsymbol{y_i})$ for the number of different N samples, where $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \dots, x_{id}]^T \in R^d$,, and $\boldsymbol{y_i} = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$. The SLFN standard with L number of hidden nodes and the activation function $g(x)$ is represented mathematically as follows.

$$f(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i g(\boldsymbol{w_i}.\boldsymbol{x_j} + b_i) = \boldsymbol{y_j} \qquad j = 1, \dots, N \tag{1}$$

where $\boldsymbol{w_i} = [w_{i1}, w_{i2}, \dots, w_{iL}]^T$ is the weight vector that connects the i-th hidden node with the input node. $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is a weight vector that connects the i-th hidden node with the output node, and b_i is the bias of the i-th hidden node. $w_i.x_j$ is denoted as the inner product of w_i and x_j.

The input layer, hidden node, and output layer of ELM were suggested to us in “genersalized” single- hidden layer feedfordward networks (SLFN). The hidden layer does not to be neuron alike in this networks [18,19,20]. The output of ELM function for generalized SLFN is represented by the following equation [21,22,23].

$$f(x) = \sum_{i=1}^{L} \beta_i g(\boldsymbol{w_i}, \boldsymbol{x_j}, b_i) = \boldsymbol{h}(\boldsymbol{x}) \cdot \beta_i \tag{2}$$

Where the value of the vector $h(x)$ is calculated for all existing data so that it will get a matrix H which is the output matrix of the hidden layer, as in equation (3). For the weight matrix β that connects the hidden layer with the output layer as in equation (4). Whereas Matrix Y is the target matrix of training data as in equation (5).

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(w_1.x_1 + b_1) & \cdots & g(w_L.x_1 + b_L) \\ \vdots & \vdots & \vdots \\ g(w_1.x_N + b_1) & \cdots & g(w_L.x_N + b_L) \end{bmatrix}_{N \times L} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times m} \tag{4}$$

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} \tag{5}$$

In ELM the input weight and bias are determined randomly, then to find the value of the output weight connecting the hidden layer with the output layer is determined by equation (6).

$$\beta = H^{\dagger}Y \tag{6}$$

Where the inverse matrix $H^{\dagger}$ is an output matrix H whose inverse value is obtained by the inverse matrix equation of moore penrose generalized inverse.

$$H^{\dagger} = (H^T H)^{-1} H^T \tag{7}$$

The ELM algorithm for SLFN can be stated [16,24,25] as follows:

Input: Training set $(x_i, y_i)$where $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T \in R^d$ dan $y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$. L the number of hidden node/neuron and the activation function g(x).

1. For $i = 1,2,\dots,L$ randomly assign the input weight vector w_i and bias b_i.s
2. Determine the matrix H defined by the equation (3).
3. Calculate the invers moore penrose generalized matrix $H^{\dagger}$ by the equation (7).
4. Calculate the output weights matrix β which is the weight towards the output node by equation (6), where Y is obtained by equation (5).

Output: A Single hidden layer Feedforward neural Network (SLFN) with an output weight vector β and a weight vector w_i and b_i bias that are randomly determined.

For each input $x_i \in R^d$ the output value of y can be calculated using equation (1). Where w_i, b_i and the activation function $g(x)$ are inputs, and the weight vector β is the output of the ELM algorithm.

3.3. *Moore Penrose Generalized Inverse*

Moore Penrose inverse is one type of inverse matrix denoted by $\boldsymbol{H}^{\dagger}$. The $\boldsymbol{H}^{\dagger}$ matrix is said to be the Moore Penrose inverse of an $\boldsymbol{H}$ matrix of size $(m \times n)$ if it satisfies one of these properties [26,27]:

1. $H\,H^{\dagger}H = H$s
2. $H^{\dagger}\,H\,H^{\dagger} = H^{\dagger}$
3. $(H\,H^{\dagger})^T = H\,H^{\dagger}$
4. $(H^{\dagger}\,H)^T = H^{\dagger}\,H$

Equation (7) can be used to fulfill one of the 4 conditions above to figure out if a matrix can fulfill one of the 4 condition. For example, there is a matrix H

$$H = \begin{bmatrix} 10 & 5 & 7 & 6 & 8 \\ 9 & 8 & 7 & 10 & 11 \\ 3 & 6 & 8 & 5 & 21 \\ 10 & 15 & 13 & 62 & 12 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 10 & 9 & 3 & 10 \\ 5 & 8 & 6 & 15 \\ 7 & 7 & 8 & 13 \\ 6 & 10 & 5 & 62 \\ 8 & 11 & 21 & 12 \end{bmatrix}$$

$$H^T.H = \begin{bmatrix} 10 & 9 & 3 & 10 \\ 5 & 8 & 6 & 15 \\ 7 & 7 & 8 & 13 \\ 6 & 10 & 5 & 62 \\ 8 & 11 & 21 & 12 \end{bmatrix} \times \begin{bmatrix} 10 & 5 & 7 & 6 & 8 \\ 9 & 8 & 7 & 10 & 11 \\ 3 & 6 & 8 & 5 & 21 \\ 10 & 15 & 13 & 62 & 12 \end{bmatrix} = \begin{bmatrix} 290 & 290 & 287 & 785 & 362 \\ 290 & 350 & 334 & 1070 & 434 \\ 287 & 334 & 331 & 958 & 457 \\ 785 & 1070 & 958 & 4005 & 1007 \\ 362 & 434 & 457 & 1007 & 770 \end{bmatrix}$$

$$(H^T.H)^{-1} = \begin{bmatrix} 0.0147 & -0.0378 & 0.0232 & 0.0023 & -0.0025 \\ -0.0378 & 0.2864 & -0.1542 & -0.0285 & -0.0147 \\ 0.0232 & -0.1542 & 0.0840 & 0.0148 & 0.0071 \\ 0.0023 & -0.0285 & 0.0148 & 0.0035 & 0.0016 \\ -0.0025 & -0.0147 & 0.0071 & 0.0016 & 0.0043 \end{bmatrix}$$

$$H^{\dagger} = [(H^T.H)^{-1}.H^T]$$

$$H^{\dagger} = \begin{bmatrix} 0.1139 & -0.0126 & -0.0388 & -0.0066 \\ -0.3138 & 0.4249 & -0.0803 & -0.0310 \\ 0.1951 & -0.2102 & 0.0397 & 0.0134 \\ 0.0176 & -0.0514 & 0.0058 & 0.0220 \\ -0.0051 & -0.0273 & 0.0596 & -0.0005 \end{bmatrix}$$

Let see if matrix H fulfill one of 4 conditions from moore penrose matrix by using one of 4 conditions for moore penrose matrix.

$H\,H^{\dagger}H = H$
$H.\,H^{\dagger} = I$

$$H.\,H^{\dagger} = \begin{bmatrix} 10 & 5 & 7 & 6 & 8 \\ 9 & 8 & 7 & 10 & 11 \\ 3 & 6 & 8 & 5 & 21 \\ 10 & 15 & 13 & 62 & 12 \end{bmatrix} \times \begin{bmatrix} 0.1139 & -0.0126 & -0.0388 & -0.0066 \\ -0.3138 & 0.4249 & -0.0803 & -0.0310 \\ 0.1951 & -0.2102 & 0.0397 & 0.0134 \\ 0.0176 & -0.0514 & 0.0058 & 0.0220 \\ -0.0051 & -0.0273 & 0.0596 & -0.0005 \end{bmatrix}$$

$$I = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

$$I.H = H$$

$$I.H = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \times \begin{bmatrix} 10 & 5 & 7 & 6 & 8 \\ 9 & 8 & 7 & 10 & 11 \\ 3 & 6 & 8 & 5 & 21 \\ 10 & 15 & 13 & 62 & 12 \end{bmatrix}$$

$$H = \begin{bmatrix} 10.0000 & 5.0000 & 7.0000 & 6.0000 & 8.0000 \\ 9.0000 & 8.0000 & 7.0000 & 10.0000 & 11.0000 \\ 3.0000 & 6.0000 & 8.0000 & 5.0000 & 21.0000 \\ 10.0000 & 15.0000 & 13.0000 & 62.0000 & 12.0000 \end{bmatrix}$$

Thus it can be concluded that the matrix $H^{\dagger}$ is the inverse of the H matrix because it fulfills one of the 4 conditions of inverse moore penrose matrix.

3.4. Forecasting

Stadtler (2005) [28], in an effort to know or see future developments, forecasting is needed to determine when an event will occur or a need will arise so that policies or actions that need to be prepared can be prepared. In general, forecasting techniques are divided into two, namely qualitative and quantitative techniques. Qualitative forecasting techniques are forecasting the data in the form of a manager's opinions, expert advice, input from consumers. Such data is in the form of sentences or summary statements of the method used, namely exploratory and normative. While quantitative techniques are forecasting, the data in the form of numerical methods used are time series [28]. The purpose of forecasting is for planning activities that must be carried out before the request comes [29].

3.5. Measuring Forecasting Errors

Mean Absolute Percentage Error (MAPE) is one method used to find the absolute value of the difference between predicted and observed values divided by the observed data (actual data), then displayed using percentages. The MAPE formula is seen in equation [30,31,32].

$$MAPE = \frac{1}{N}\sum_{j=1}^{N} \left|\frac{o_j - y_j}{y_j}\right| \times 100\% \tag{8}$$

where

N: Many forecast periods per road (1 period),
o_j: j-th output data (prediction of queue length),
y_j: j-th actual queue length data (data testing).

The size of forecasting ability in this study is determined by the small percentage of error obtained from the testing process, the lower the error achieved, the higher the accuracy value and the better the forecasting ability.

3.6. *Correlation Analysis*

Correlation analysis attempts to measure such a close relationship between two variables by using a number called the correlation coefficient [5]. The correlation coefficient can range from an extreme value -1 (perfect negative correlation) through zero to an extreme value +1 (perfect positive correlation). The number of pairs (1,1), (2,2), (3,3), (4,4) and (5,5) are perfectly correlated, i.e. r = + 1, and the number of pairs (1.5), (2,4), (3,3), (4,2) and (5,1) have perfect negative correlation, r = -1. Intuitively, the correlation coefficient can be interpreted in two events: (i) as the direction of the relationship between two meaningful measures: they tend to increase or decrease together (positively related), one increases and the other decreases (negatively related), or their movements are separate (uncorrelated); and (ii) as a strength of association which means that if the absolute value of the correlation moves away from zero then the two measures of association become stronger [33].

To compute correlation coefficient can be used formula on equation (9).

$$r_{XY} = \frac{n\sum XY - (\sum X)(\sum Y)}{\sqrt{n\sum X^2 - (\sum X)^2}\sqrt{n\sum Y^2 - (\sum Y)^2}} \tag{9}$$

where

n = The number of X and Y data pairs,
$\sum X_i$ = Total number of variable X,
$\sum Y_i$ = Total number of variable Y,
$\sum {X_i}^2$ = Total sum of the squares of variable X,
$\sum {Y_i}^2$ = Total sum of the squares of variable Y,
$\sum X_i Y_i$ = Total sum of the multiplication results between variable X and Y.

3.7. *Traffic Congestion*

Congestion is a condition in which the flow of traffic that passes on the road segment that is reviewed exceeds the capacity of the road plan which results in the free speed of the road segment approaching or exceeding 0 km / h, causing a queue. At the time of congestion, the value of the degree of saturation in the road section will be reviewed where the congestion will occur if the value of the degree of saturation reaches more than 0.5 [34].

Traffic depends on the capacity of the road, the amount of traffic that wants to move, but if the capacity of the road cannot accommodate, then the existing traffic will be hampered and will flow according to the maximum road network capacity [35]. If the flow of traffic approaches capacity, congestion starts. Congestion increases when the current is so large that vehicles are very close to each other. Total congestion occurs when a vehicle must stop or move very slowly [36].

Traffic congestion on a highway occurs when the flow of traffic vehicles increases with the increase in travel demand at a certain period and the number of road users exceeds the existing capacity [37].

3.8. *Data (Pre-processing)*

The data used in this study are data taken from DISHUB DI Yogyakarta, data in the form of CCTV video recordings on several roads then viewed and calculated the length of the queue on the traffic lights manually when the status of the red light before the green light, then the numerical results of the length of the queue The calculated results were used in this study.

The calculation process is manually done by counting the number of vehicles that are queued at the time of the red traffic lights, and only vehicles recorded by CCTV camera are counted. Queues are counted only the longest vehicle queue without considering the width of the road. So if there is more than one queue in one lane, only the longest queue will be counted before the light turns green.

The unit of queue length data is the number of vehicles queuing in a traffic light. In this case, the unit is the car.

where

1 car = 2 motorcycles which waiting in line,
0.5 car = 1 motorcycle
2 cars = 1 bus

The car unit has been chosen because the car is a vehicle that is easier to see on CCTV camera footage. Many of the Roads to be taken in the length of the queue data in this study are Three (3) Roads, namely Affandi Street, National 15 Street and Ringroad Utara Street. The connected road is between Affandi Street and Ringroad Utara Street, between National 15 Street and Ringroad Utara Street.

4. Results and Discussion

In this study, an experiment will be conducted to test by changing the size of the Window to predict every hour in the testing data, as well as measuring the running time from the training process to testing for each of these experiments. Then you will see which experiments have good performance based on the MAPE obtained and the Running Time of each of these experiments.

The experiment result obtained by changing the size of the Window which is the number of input data to the network. Afterwards, measuring the accuracy of the predicted results to the real data for all testing data by calculating the MAPE of all data at the predicting hour, Running Time of each Experiment and the level of correlation of each connected road for each selected hour. There are 10 experiments in which one experiment had a different window size with 15 repetitions for each experiment then average MAPE and average Running Time could be obtained. The repetitions are needed because the parameters on the hidden node (weights and biases) being determined at the beginning randomly. Therefore, by doing the repetitions, the results will be optimum to measure the error (Average MAPE).

4.1. MAPE and Running Time of All Experiments

At this stage, testing of all experiments was done by calculating the average of MAPE and Running Time for each experiment, wherein 1 experiment had 15 repetitions that were then calculated and the average MAPE and average Running Time for each experiment (experiment).

Table 1. The Average MAPE of Each Experiment

Experiment to-	Window Size	Average MAPE (%)
1	12 minutes	16.41349
2	18 minutes	15.87821
3	24 minutes	15.5882
4	30 minutes	15.62788
5	36 minutes	15.64965
6	42 minutes	15.55169
7	48 minutes	15.45339
8	54 minutes	15.55314
9	60 minutes	15.41094
10	192 minutes	15.46429

The average MAPE for each experiment is obtained as shown in Table 1. Furthermore, the average Running Time of each experiment can be seen in Table 2. The window size was devided into 10 for each experiment where the difference of each window size was 6 minutes as shown in Table 1 and Table 2.

Table 2. The Average Running Time of Each Experiment

Experiment to-	Window Size	Average MAPE (%)
1	12 minutes	3.432895
2	18 minutes	4.249102
3	24 minutes	5.222633
4	30 minutes	5.737357
5	36 minutes	6.541303
6	42 minutes	7.846938
7	48 minutes	8.466337
8	54 minutes	9.129009
9	60 minutes	10.14822
10	192 minutes	10.95131

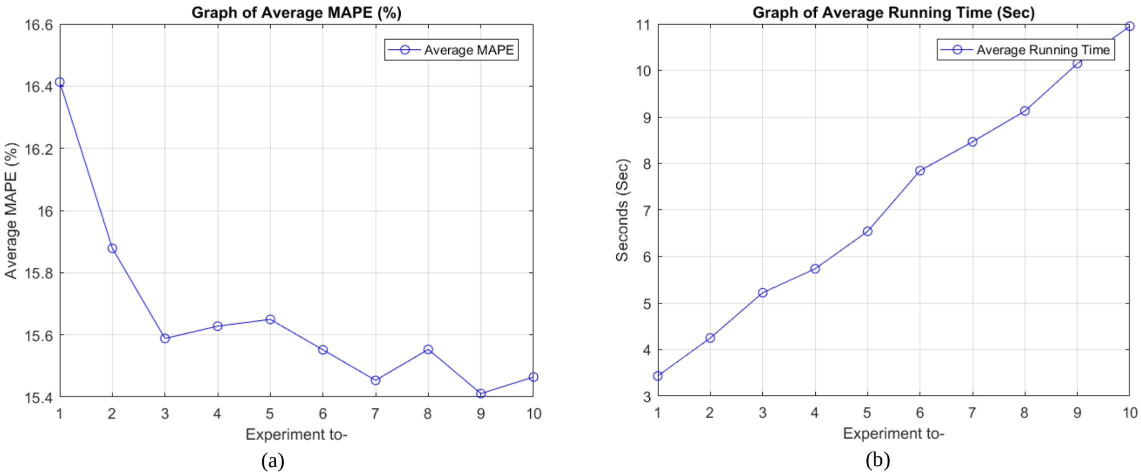


Fig.2. Chart of Average MAPE and Chart of Average Running Time

Then plot graphs for the average MAPE of all experiments after repetition of 15 times and graph plot for the average Running Time of all experiments after repetition of 15 times as shown in Fig. 2. The MAPE of each experiment (Fig. 2(a)) has decreased as the window size of the training data gets bigger. However, for the Running Time (Fig. 2(b)) of each trial, it took longer to run the ELM for the larger size. It means that when the window size was bigger, then it will end up with a lower error (Average MAPE as shown in Fig. 2(a)), but it needs more time to finish the computation as shown in Fig. 2(b).

Furthermore, to find out the optimum window size based on the average MAPE and average Running Time, a graph plot is performed for both simultaneously as shown in Fig. 3 by Normalizing for each average MAPE and average Running Time. The Normalized formula is seen in equation (10).

$$Normalized(x) = \frac{x - minValue}{maxValue - minValue} \tag{10}$$

where

x : The value to be normalized,
$minValue$: The minimum value of the data set,
$maxValue$: The maximum value of the data set.

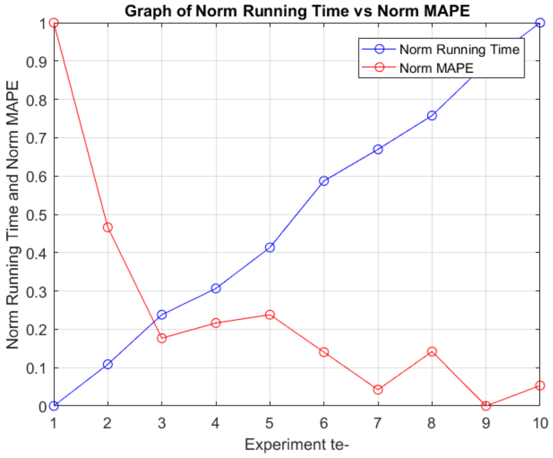


Fig.3. Chart of Norm Average Running Time vs Norm Average MAPE

From Fig. 3, it can be seen that the 3rd Experiment with a Window size of 24 minutes is the optimum window in this study because it has better accuracy and running time compared to other experiments with an average MAPE value of 15.5882% and average Running Time of 5.2226 seconds. Even though the 9th trial has the smallest average MAPE value, it takes much longer Running Time.

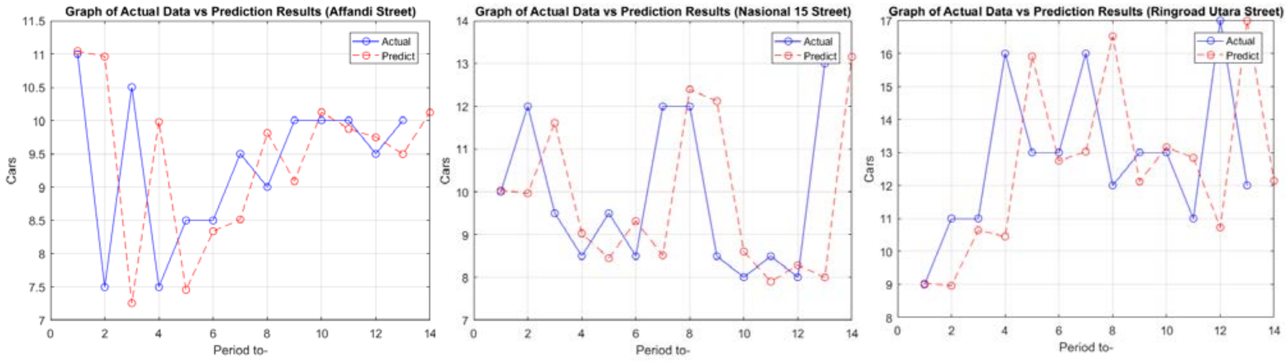
4.2. Prediction Model

From the results of experiments conducted on each different window size, the model produced by ELM with a 24-minute window size is optimum. ELM with a window size of 24 minutes is able to produce predictive values with an average MAPE value of 15.5882% and an average value of Running Time of 5.2226 seconds. The best results from ELM are obtained with a window size of 24 minutes, which means to predict the length of the vehicle queue at the next traffic light (3 minutes ahead) required data queue length of the vehicle 24 minutes earlier as the input.

Table 3. Actual Data and Prediction Result of Queue Length (Cars)

Period to-	Clock	Affandi Street		Nasional 15 Street		Ringroad Utara Street	
		Actual Data	Prediction	Actual Data	Prediction	Actual Data	Prediction
1	09:12	11	11.04046	10	10,03678	9	9,033101
2	09:15	7.5	10.95993	12	9,963573	11	8,967216
3	09:18	10.5	7.257287	9.5	11,61166	11	10,64402
4	09:21	7.5	9.980215	8.5	9,029718	16	10,45546
5	09:24	8.5	7.45649	9.5	8,450688	13	15,90718
6	09:27	8.5	8.334242	8.5	9,314741	13	12,74649
7	09:30	9.5	8.515588	12	8,515588	16	13,02384
8	09:33	9	9.810302	12	12,39196	12	16,52261
9	09:36	10	9.089621	8.5	12,1195	13	12,1195
10	09:39	10	10.12424	8	8,605604	13	13,16151
11	09:42	10	9.875997	8.5	7,900798	11	12,8388
12	09:45	9.5	9.751183	8	8,288505	17	10,7263
13	09:48	10	9.495575	13	7,996274	12	16,99208
14	**09:51**		**10.11937**		**13.15519**		**12.14325**

By using the model that has been generated, prediction ware made for all data testing. So, the prediction result ware obtained as shown in Table 3. Based on the optimum window size that has been used to predict, it can be said that to predict the vehicle queue at 09:36 needs 24 minutes earlier of the vehicle queue which is at 09:12 to 09:33 as the input, the same thing also applied to predict the other time which are at 09:12 to 09:48 as shown in Table 3. Therefore, the prediction at 09:51 can be obtained where using the vehicle queue data at 09:27 to 09:48 as input. So, the predicted queue length for Affandi Street was 10.11937, for National 15 Street was 13.155 and Ringroad Utara Street was 12.14325.



(a) (b) (c)

Fig.4. Chart of Actual Data vs Prediction Results on Every Roads

A graph plot is performed for each road to see the graphic patterns between the actual data and the predicted results. Where 1 period has 3 minutes difference of each time start at 09:12 to 09:51. As can be seen that the predictive chart pattern follows the chart pattern from the actual data as shown in Fig. 4. Although The Affandi street has significant differences between actual data and prediction value on some period (Fig.4 (a)) such as on period 2 (at 09:15), period 3 (at 09:18) and period 4 (at 09:21), but it also has prediction on period 1, period 6, period 10, period 11 and period 12 which have small error value. Hence, the prediction result compared to actual data has small error in average. Also, the same thing occure as shown in Fig. 4 (b) and Fig. 4(c).

4.3. *Correlation Between Connected Roads*

Correlation that will be calculated only between roads that are connected, is meant to be connected which is directly connected as shown in Fig. 5 where the dependent variable (Y) is the road whose queue length is affected by the road another (X) is connected where the direction of vehicle flow on the road (X) leads to the road being the dependent variable (Y). In addition, the vehicle queue as shown in Fig. 5 also shown the direction of the vehicles on the traffic light.

In this case, the connected road is between Affandi Street ($X1$) and Ringroad Utara Street (Y), and between National 15 Street ($X2$) and Ringroad Utara Street (Y) as shown in Fig. 5. It is because the direction of the vehicles queue on $X1$ and $X2$ could affect the vechicle queue on Y. Obtained correlation coefficients between the roads can be seen as shown in Table 4.

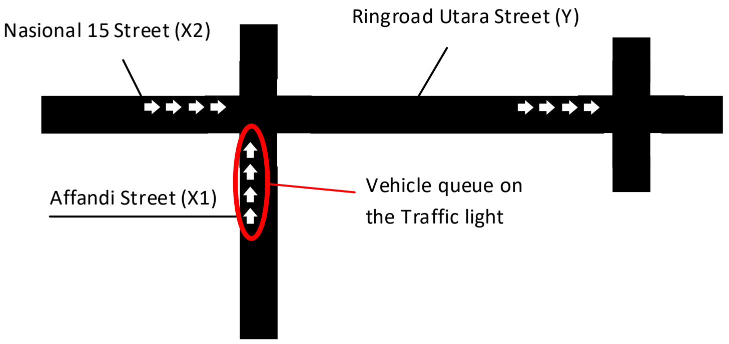


Fig.5. The Connected Roads

From Table 4, it can be seen that the relationship between National 15 Street and Ringroad Utara Street (from 09:15 to 09:48) had a weak relationship where the largest correlation coefficient value was 0.2685 (at 09:12). Therefore, if there is congestion occure on Nasional 15 street, then it cannot be determined that it is also happening on Ringroad Utara Street.

Table 4. Correlation Coefficient for Every Connected Roads

Clock	Affandi Street and Ringroad Utara Street	Nasional 15 Street and Ringroad Utara Street
09:12	0.507022	0.268467
09:15	0.468307	0.176482
09:18	0.379734	0.205509
09:21	0.320192	0.207115
09:24	0.24692	0.156944
09:27	0.215309	0.138359
09:30	0.109082	0.074999
09:33	0.090264	0.065324
09:36	0.07429	0.069607
09:39	0.033142	0.067505
09:42	0.00994	0.064571
09:45	-0.04281	-0.01516
09:48	-0.09017	-0.00746

Besides, the relationship between Affandi Street and Ringroad Utara Street had a strong relationship with a correlation coefficient of 0.507 (at 09:12), but their level of relationship decreased (at 09:15 to 09:48). However, if there is congestion occurs on Affandi Street at 09:12, then it can be predicted that it also occurs on Ringroad Utara Street. On the other hand, at 09:15 to 09:48 cannot be determined because they had a weak relationship based on the correlation coefficient obtained.

5. Conclusions

The optimum prediction of queue length using ELM on particular roads can be obtained by giving input to the network with 24 minutes observation data (window size). This optimum condition is based on the accuracy of prediction and the running time which have an average MAPE value of 15.5882% and an average running time of 5.2226 seconds. Even though The Window with a Size of 60 minutes has the smallest average value of MAPE, but requires a much longer Running Time. In addition, the window size gives an impact on the accuracy and running time in making predictions. It is showed by the value of MAPE on every different window size in the experiment.

The Extreme Learning Machine method can be used to predict the length of the queue at the intersection of traffic lights by using the optimum Window size (in this study 24 minutes). By using the optimum Window size, the result of the predicted length of the queue for Affandi street was 10.119, for National 15 street was 13.155 and for Ringroad Utara street was 12.143. Where the graphic patterns between Actual Data and Prediction Results had a similar pattern.

Based on the overall findings, further research is required. Firstly, using more data set from more roads at different time, day and week which more interval of each time. Secondly, improving data input for ELM before doing prediction by creating a system that can calculate the length of the queue on the traffic lights automatically.

References

[1] I. G. A. Wibawa, L. Arida, and A. Rahning, “Deteksi Kemacetan Lalu Lintas Jalan Raya Menggunakan Metode Moving Object Detection,” no. November, pp. 2–4, 2016.

[2] I. dkk Santoso, *Manajemen Lalu-Lintas Perkotaan*. Bandung: Badan Penerbit ITB, 1997.

[3] G. Huang, Q. Zhu, and C. Siew, “Extreme Learning Machine : A New Learning Scheme of Feedforward Neural Networks,” *IEEE Int. Jt. Conf. Neural Networks*, vol. 2, pp. 985–990, 2004, doi: 10.1109/IJCNN.2004.1380068.

[4] U. Mahdiyah, M. I. Irawan, and E. M. Imah, “Study Comparison Backpropogation, Support Vector Machine, and Extreme Learning Machine for Bioinformatics Data,” *J. Ilmu Komput. dan Inf. (Journal Comput. Sci. Information)*, vol. 1, pp. 53–59, 2015, [Online]. Available: http://dx.doi.org/10.21609/jiki.v8i1.284.

[5] R. E. Walpole and R. H. Myers, *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuan (Terjemahan RK Sembiring)*, Ed. 4. Bandung: Penerbit ITB, 1995.

[6] R. E. Walpole, *Pengantar Statistika (Alihbahasa: Ir. Bambang Sumantri)*, Ed. 3. Jakarta: PT Gramedia Pustaka Utama, 1995.

[7] Surjandy, F. Anindra, H. Soeparno, and T. A. Napitupulu, “CCTV traffic congestion analysis at Pejompongan using case based reasoning,” *2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018*, vol. 2018-Janua, pp. 861–865, 2018, doi: 10.1109/ICOIACT.2018.8350807.

[8] R. Chairan and A. Martiningtyas, “Sistem Lampu Lalu Lintas Dengan Video Processing Sebagai Pendeteksi Kepadatan Lalu Lintas,” Universitas Gadjah Mada, 2017.

[9] S. Mahatmaputra, E. Permata, and William, “Deteksi Kemacetan Lalu Lintas Melalui Kamera Menggunakan Pin Hole Algorithm,” *Comtech*, vol. 2, no. 2, pp. 821–834, 2011.

[10] I. G. A. Wibawa, L. Arida, and A. Rahning, “Deteksi Kemacetan Lalu Lintas Jalan Raya Menggunakan Metode Moving Object Detection,” *Semin. Nas. SAINSTEK 2016*, no. November, pp. 2–4, 2016.

[11] I. H. Setiadi and Y. H. P, “Perancangan Sistem Pendeteksi Kepadatan Lalu Lintas Menggunakan Image Processing Dengan Metode Background Subtraction Pada Sikomolintas,” *Perpust. UNIKOM*, pp. 1–8, 2017, [Online]. Available: https://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-indrahadis-37151.

[12] A. R. Wibowo and I. Soesanti, “Analisis data time series dan VCR kepadatan lalu lintas (studi kasus : Jalan Adisucipto depan Ambarukmo Plaza),” *Elinvo*, vol. 2, Nomor 2, pp. 131–137, 2015.

[13] T. B. Atmojo, R. Pulungan, and H. Syahputra, “Pengembangan Model Peramalan Permintaan Kebutuhan Reseller Menggunakan Extreme Learning Machine dalam Konteks Intelligent Warehouse Management System(IWMS),” *Semin. Nas. Inform. 2013*, vol. 2013, no. semnasIF, pp. 258–263, 2013.

[14] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi Offset, 2005.

[15] S. Haykin, *Neural Networks : A Comprehensive Foundation*, Second. Delhi: Pearson Education, 2005.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 17, no. 1, pp. 489–501, 2006, doi: 10.1007/s10462-013-9405-z.

[17] G. Bin Huang, D. H. Wang, and Y. Lan, “Extreme learning machines: A survey,” *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, 2011, doi: 10.1007/s13042-011-0019-y.

[18] G. Huang and L. Chen, “Convex incremental extreme learning machine,” vol. 70, pp. 3056–3062, 2007, doi: 10.1016/j.neucom.2007.02.009.

[19] G. Huang and L. Chen, “Author ’ s personal copy Enhanced random search based incremental extreme learning machine Author ’ s personal copy,” vol. 71, pp. 3460–3468, 2008, doi: 10.1016/j.neucom.2007.10.008.

[20] G. Huang, S. Member, H. Zhou, X. Ding, and R. Zhang, “Extreme Learning Machine for Regression and Multiclass Classification,” vol. 42, no. 2, pp. 513–529, 2012.

[21] G. Bin Huang, X. Ding, and H. Zhou, “Optimization method based extreme learning machine for classification,” *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010, doi: 10.1016/j.neucom.2010.02.019.

[22] M. A. A. Albadr and S. Tiun, “Extreme learning machine: A review,” *Int. J. Appl. Eng. Res.*, vol. 12, no. 14, pp. 4610–4623, 2017.

[23] Anoop Kumar Patel, Sanjay Kumar Jain, " Arterial Parameters and Elasticity Estimation in Common Carotid Artery Using Deep Learning Approach", International Journal of Image, Graphics and Signal Processing, Vol.11, No.11, pp. 18-28, 2019.

[24] R. Singh and S. Balasundaram, “Application of Extreme Learning Machine Method for Time Series Analysis,” *Proc. World Acad. Sci.*, vol. 26, no. Part 1, pp. 361–367, 2007, doi: 10.1148/radiology.138.2.7455105.

[25] A. Anbarasa Pandian, R. Balasubramanian,"Analysis on Shape Image Retrieval Using DNN and ELM Classifiers for MRI Brain Tumor Images", International Journal of Information Engineering and Electronic Business, Vol.8, No.4, pp.63-72, 2016.

[26] R. Macausland, “University of Puget Sound MATH 420: Advanced Topics in Linear Algebra The Moore-Penrose Inverse and Least Squares,” 2014.

[27] P. A. G. T. Ag, “The Moore-Penrose Pseudoinverse,” vol. 1, no. 1, pp. 1–5, 1972.

[28] H. Stadtler, “Supply chain management and advanced planning - Basics, overview and challenges,” *Eur. J. Oper. Res.*, vol. 163, no. 3, pp. 575–588, 2005, doi: 10.1016/j.ejor.2004.03.001.

[29] I. N. Pujawan, *Supply Chain Management*, Edisi Kedu. Surabaya: Guna Widya, 2010.

[30] P. M. Swamidass, Ed., “MAPE (mean absolute percentage error)MEAN ABSOLUTE PERCENTAGE ERROR (MAPE),” in *Encyclopedia of Production and Manufacturing Management*, Boston, MA: Springer US, 2000, p. 462.

[31] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, “Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error,” 2017.

[32] Fahirah, Lily Wulandari, " The Implementation of Least Square Method on the Palm Shells Sales Forecasting Application", International Journal of Information Engineering and Electronic Business, Vol.12, No.5, pp. 1-13, 2020.

[33] S. Makridakis, S. C. Wheelwright, and V. E. McGEE, *Metode dan Aplikasi Peramalan (Terjemahan: Ir. Untung Sus Andriyanto, M.Sc.)*, Ed. 2. Jakarta: Erlangga, 1995.

[34] D. G. of H. D. of U. R. Directorate General Bina Marga, “Highway Capacity Manual Project (Hcm),” vol. 1, no. I, p. 564, 1997, doi: 10.1021/acsami.7b07816.

[35] B. D. Sinulingga, *Pembangunan Kota Tinjauan Regional dan Lokal*. Bandung: Penerbit ITB, 1999.

[36] O. Z. Tamin, *Perencanaan dan Pemodelan Transportasi*, Ed. Ke 2. Bandung: Penerbit ITB, 2000.

[37] M. D. Meyer and E. J. Miller, *Urban Transportation Planning: A Decision-Oriented Approach*. New York: McGraw-Hill, 1984.

Authors’ Profiles

Kasliono, S.Mat., M.Cs. is a Master Student in Computer Science, Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science, Universitas Gadjah Mada. The research interest is artificial intelligence.

Suprapto, Drs., M.I.Kom., Dr. received his Doctor in computer science from Universitas Gadjah Mada, Indonesia. His Master in computer science from University of Indonesia, Indonesia. His Undergraduate in Mathematics from Universitas Gadjah Mada, Indonesia. The research interest is graph theory with research cluster is computer science.

Faizal Makhrus, S.Kom., M.Sc., Ph.D. received his Master from Osmania University, India. His Doctor was received from Kanazawa University, Japan. The research interest are Simulation, Numerical Method, Distributed Database, and Pattern Recognition with research cluster is computer science.

How to cite this paper: Kasliono, Suprapto, Faizal Makhrus, "Point Based Forecasting Model of Vehicle Queue with Extreme Learning Machine Method and Correlation Analysis", International Journal of Intelligent Systems and Applications(IJISA), Vol.13, No.3, pp.11-22, 2021. DOI: 10.5815/ijisa.2021.03.02