Knowledge Engineering and Data Science (KEDS)
Vol 5, No 2, December 2022, pp. 168–178

pISSN 2597-4602
eISSN 2597-4637

Traffic Density Prediction using IoT-based Double Exponential Smoothing

Rosa Andrie Asmara [a,1,*], Noprianto [a,2], Muhammad Ainur Ilmy [a,3], Kohei Arai [b,4]

[a] *Informatics Engineering Study Program, Information Technology Department, State Polytechnic of Malang, Jl. Soekarno Hatta No.9 Malang 65141, Indonesia*
[b] *Information Science Department, Saga University, 1 Honjou 840-0027, Saga, Japan*
[1] *rosa_andrie@polinema.ac,id* *; [2] *noprianto@polinema.ac.id*; [3] *polinema1641720019@gmail.com*; [4] *arai@cc.saga-u.ac.jp*
* *corresponding author*

ARTICLE INFO

Article history:
Received 1 November 2022
Revised 7 December 2022
Accepted 11 December 2022
Published online 30 December 2022

Keywords:
Traffic Density
Haar Cascade
Double Exponential Smoothing
Mean Absolute Percentage Error
Website Application

ABSTRACT

The number of vehicles and currents that tend to increase causes traffic density. A system is proposed to calculate the number of vehicles and predict real-time traffic density. This research uses Haar Cascade to detect the number of cars and motorcycles and the Double Exponential Smoothing (DES) for forecasting the number of vehicles on the road. MAPE describes forecasting accuracy as a base for selecting the best smoothing constant (Alpha). The best test results from June 13 to 20, 2020, are cars on June 14, 2020 (alpha 0.5, MAPE 0%) and Motorcylecycles on June 18, 2020 (alpha 0.5, MAPE 0.1134%). The most significant MAPE results of the car were on June 15, 2020, with alpha 0.5 and MAPE 2.1073%. The 3 minutes haar cascade detects 72.58% of cars and 81.90% of motorcycles.

I. Introduction

The total number of automobiles on the road often keeps growing at a high rate year after year. The tremendous population increase results in a high density of traffic [1]. Many cars exceed the capacity of the traffic segment, lowering the amount of space that is free of traffic [2] and increasing the number of vehicle lines [3], which may halt or stop the mobility of vehicles. The increase in vehicle flow, which often takes place in response to increased demand for transportation during a specific period, may be used to detect the presence of traffic density when it is seen.

Congestion on the roads results in substantial losses, including the most significant increase in the amount of time spent in traffic [4], which may result in considerable societal costs [5], including operating expenses [6], wasted time [7], air pollution [8], accident rates [9], noise [10], and discomfort of pedestrians [11]. It is essential to have technology that is able to estimate the number of cars currently in traffic to keep up with the times. This kind of technology may be helpful to motorcyclists, police, the government, and other connected parties as a source of information and assessment data. In order to solve these issues, it is required to develop a system that can identify and forecast the number of cars that will be present within a particular time. Forecasting aims to make an educated guess as to what will take place in the future by using pertinent information from previous periods. The construction of a forecasting system uses various methodologies, some of which include moving averages [12], trend projection [13], and exponential smoothing [14].

Exponential smoothing is a time series forecasting approach for univariate data that may be expanded to accommodate data with a systematic trend or seasonal component [15]. The method was initially developed for use with univariate data but has now been adapted for use with multivariate data [16]. It is an effective way of forecasting that may be used as an alternative to the widely used Box-Jenkins ARIMA family of approaches. Projections made using this method for longer periods are often highly inaccurate, which is why exponential smoothing is typically reserved for making projections for shorter periods. When smoothing time series data using an exponential function, the

https://doi.org/10.17977/um018v5i22022p168-178

weights given to the most recent observations progressively lower as they move toward the oldest observations [17]. The more time has passed since the data was collected, the less importance (weight it is assigned [18]. The more recent data is given greater weight since it is considered more relevant [19]. The smoothing settings determine the weights assigned to observations [20]. Based on previous findings, When dealing with data that exhibit patterns, the single technique is often less trustworthy than the double procedure when managing the data.

Therefore, this study promotes Double Exponential Smoothing (DES) as IoT based prediction system. This strategy is beneficial for predicting short-term and medium-term particularly when many outcomes are required. Data that follows a linear trend might provide support for this strategy. A system for predicting traffic density may be designed using the DES approach. In the future, this factor may decide whether the traffic density will increase or decrease. Using this forecasting technique, overcoming these challenges and roadblocks will be possible.

II. Method

The stages of research in designing a traffic density prediction system using the IoT-based double exponential smoothing method can be seen in Figure 1.

Fig. 1. The stages of research

A. Data Collection

Data collection was carried out according to the source and type required. Data collection in this study was carried out using quantitative variables/data instruments. Namely, the data obtained was regularly until the end of the study. Collecting data on the number of vehicles per 3 seconds is taken by observing the Raspberry Pi test in the field as a data parameter. This 3 second timeframe are taken to handle the computation needed for the raspberry Pi. In making the haar cascade training data, the writer recorded the traffic flow through the webcam according to the test hours, then took the object of the motorcycle, bike, and car through the video frame. The results of data training were carried out on as many as 7x cars and 3x motorcycle bikes.

In this study, an application will be made to detect and count the number of vehicles per 3 seconds and display data in the form of forecasting charts on the website. The data used as training and test data are obtained in application testing during the traffic determination process and observations at the research site. The input variables used are webcam coordinates, ROI coordinates, trigger line coordinates, and alpha values.

The training data collection aims to obtain the types and patterns of vehicle objects at the test site with a ratio of 100: 1000, namely 100 positive and 1000 negative objects. The haar cascade training data results are in the form of XML files for motorcycles, bikes, and cars. Several stages of object training in this research are:

- Take a positive image through the video frame
- Gather negative image imagery
- Minimizes positive and negative image pixels
- Positive and negative image transformations into greyscale
- Conduct training using the Cascade Trainer GUI application

Some positive and negative training results can be seen the Figure 2 to Figure 4. Figure 2 shows the positive training images for cars. While Figure 3 represents negative image of motorcycles, the Figure 4 shows negative samples (figure instead of cars and motorbikes).

Fig. 2. Car samples

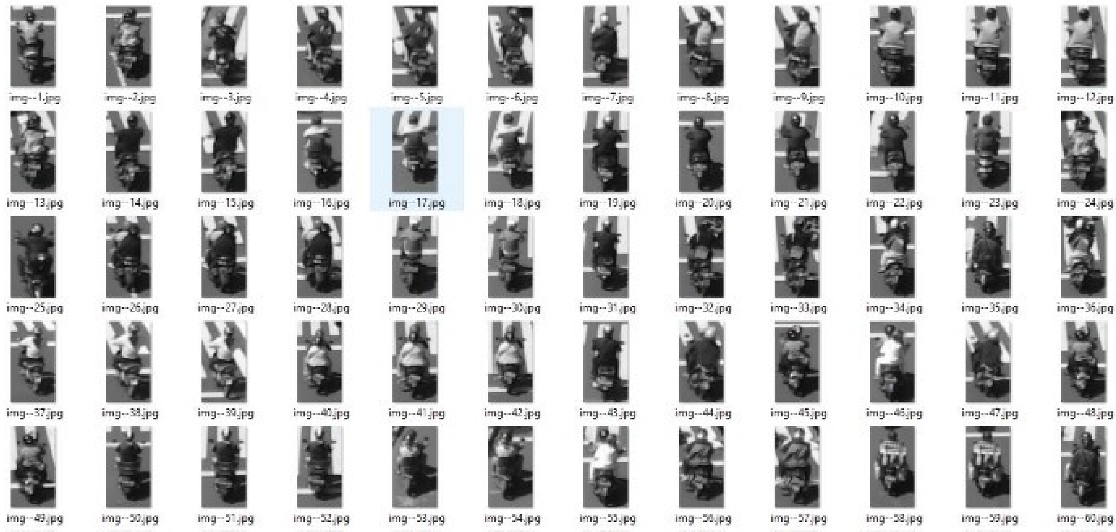

Fig. 3. Motorcylecycle samples

Fig. 4. Negative samples

B. Haar Cascade Classifier

Haar-like feature, also known as Haar Cascade Classifier, is a rectangular (square) feature that gives a specific indication of an image [21]. The Haar cascade classifier comes from the idea of Paul Viola and Michael Jhon, hence the name Viola & Jhon method [22]. The idea of the Haar-like feature is to recognize objects based on the simple value of the feature but not the pixel value of the object's image [23]. This method has the advantage of being high-speed computation because it only depends on the number of pixels in a square, not every pixel value of an image [24]. This method is a method that uses a statistical model (classifier). The approach to detecting objects in images combines four primary keys, namely Haar-like feature, Integral Image, Adaboost learning, and Cascade Classification can be seen in Figure 5.

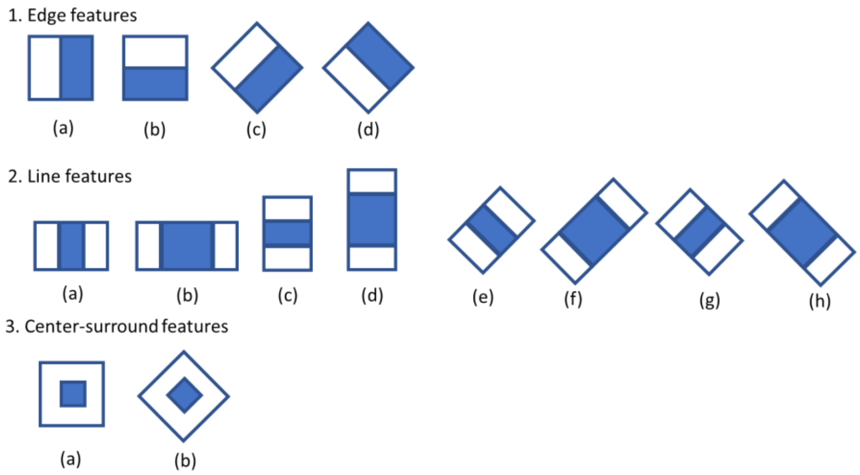

Fig. 5. Haar Cascade Classifier

The haar cascade process can be seen in Figure 6. The haar method requires two types of object images in the training process to detect an object. Positive samples contain the image of the object you want to detect. Positive samples contain a knife image if you want to detect a knife. Negative samples contain images other than the object that you want to recognize. Negative samples are generally in the form of background images such as walls and scenery. The resolution for the negative sample is recommended to have a camera resolution. The Haar method training uses those two types of samples [25]. The information from the training is then converted into a statistical model parameter. A Cascade classifier is a chain of stage classifiers, where each stage classifier is used to detect an object of interest in the image sub-window.

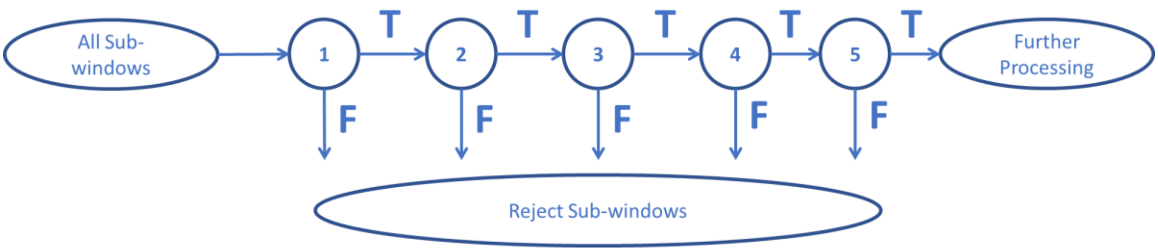

Fig. 6. Haar Cascade Process

C. Double Exponential Smoothing

Double exponential smoothing (DES) is the improvement of the exponential smoothing (ES) method. The ES (Makridakis, 1999) is a continuous procedure focused on the exponential descent of priority to older objects of observation [26]. The DES method is a linear model invented by Brown. This method performs a twice-smoothing process. The DES method is usually used to predict data with trends. Moreover, data patterns are likely to rise.

DES method can more efficiently model trends and levels from a time series than other methods. DES requires fewer data and uses one parameter to simplify it. However, DES requires parameter optimization, finding the most optimal α (Alpha) that takes time [27]. The steps in calculating using the DES method are as follows. Determine the First Smoothing Value (S_t^I) as in (1).

$$S_t^I = \alpha X_t + (1 - \alpha) S_{t-1}^I \quad (1)$$

Determine the Second Smoothing Value (S_t^{II}) as in (2).

$$S_t^{II} = \alpha S_t^I + (1 - \alpha) S_{t-1}^{II} \quad (2)$$

where S_t^I is forecast value for period t, α is exponential weighting constant, $X_t$is actual value of period t, S_{t-1}^I is forecast value for period $t-1$, S_t^{II} is value of DES period t, S_{t-1}^{II} is value of DES t-1.

Specifying a constant value (a_t) as in (3).

$$a_t = 2S_t^I - S_t^{II} \quad (3)$$

Specifying slope value (b_t) as in (4).

$$b_t = \frac{a}{1-a}(S_t^I - S_t^{II}) \quad (4)$$

Determining forecasting value as in (5).

$$F_{t+1} = a_t + b_t m \quad (5)$$

where F_{t+1} is forecasting value, and m is predicted future period.

D. System Design of the Internet of Things (IoT)

Internet of Things (IoT) is a computing concept that describes the idea of everyday physical objects connected to the internet and can identify themselves to other devices [28]. IoT is significant because an object can be presented digitally as more extensive than the object itself [29]. The object is no longer only related to the user; now, it is connected to the object and its surrounding database data. In this research, the system has three stages: monitoring, process automation, and controlling.

In the monitoring process, the system will recognize the vehicle object using the haar cascade detection via a webcam installed on the Raspberry PI to determine the number of vehicles per 3 seconds. Then the data will be transferred to the database on the MySQL server via the network. Then at the Automation stage, the data in the MySQL server is taken to carry out the forecasting process using the Double Exponential Smoothing method to produce forecasting or output values on the website. The final stage is Controlling, a control system that the user can manage to make arrangements as expected of the system, anticipating the automation system if there is no traffic as desired by the user. For the software and hardware used in this system can be seen in Table 1 and Table 2 respectively.

Table 1. Software specifications

No.	Device	Information
1	Text Editor	To make it easier to write programs and develop applications on Windows and Raspbian.
2	Local Server	as a server consisting of MySQL database and as a PHP language translator
3	Remote Desktop Protocol	Windows application to perform remote desktop on different device operating systems.
4	Cascade Trainer GUI	As an image data trainer application for vehicle detection using the Haar Cascade
5	Raspbian OS	Operating system to traffic all computer activities on the Raspberry PI
6	Python IDE	An application for writing unique program code for python programming language.
7	Library OpenCV	A software library aimed at real-time dynamic image processing.

Table 2. Hardware specifications

No	Device	Picture	Information
1	Raspberry PI 3 Model B+		To perform the computational process.
2	USB Cable		To connect power to the Raspberry
3	Heatsink Raspberry Pi		To expand the heat transfer of the raspberry processor so that the temperature is not too hot
4	Raspberry Pi cooling fan type C		To speed up the air circulation process on the raspberry processor
5	Power Bank 10000 mah		As a power source for raspberries
6	Webcam Logitech c290		To carry out the vehicle detection process in real time
7	Gorilla Tripod		Support for the webcam makes finding an image capture angle easier.
8	LAN Cable		Used as a local network when data retrieval for eight days

E. Evaluation Metric

The evaluation metric in this research is using MAPE. MAPE measures the accuracy of a prediction. MAPE is used to evaluate the accuracy of forecasting using errors in percentage terms. The interpretation of the MAPE value is as follows [30].

a. < 10 % = very accurate forecast
b. 10 % - 20% = accurate forecast
c. 20 % - 50% = forecast is quite accurate
d. > 50 % = forecast is not accurate

Calculating MAPE shows the forecast's accuracy in percentage form by determining the PE (Percentage Error). PE determines the percentage error of the forecast. Here is the formula for calculating PE and MAPE as in (6) and (7).

$$\text{PE} = \left(\frac{Xt-Ft}{Xt}\right) X\ 100 \tag{6}$$

$$\text{MAPE} = \sum_{t=1}^{n} \frac{PEt}{n} \tag{7}$$

where n is period value, Xt is true value in period 1, and Ft is the forecast value in the 1st period.

III. Results and Discussion

This system testing is carried out to provide an accurate system in automation and quick and easy settings for users of these systems and applications. In implementing the system, it has been tested at the place it is implemented, namely at the crossing bridge Jl. Basuki Rahmat Klojen, Malang City starting from 10:00 to 10:30 am. The system can detect cars and motorcycle bikes and inform data per 3 seconds on the website in real time. Monitoring the number of vehicles can also be done on the website application in real time, as seen in Figure 7.

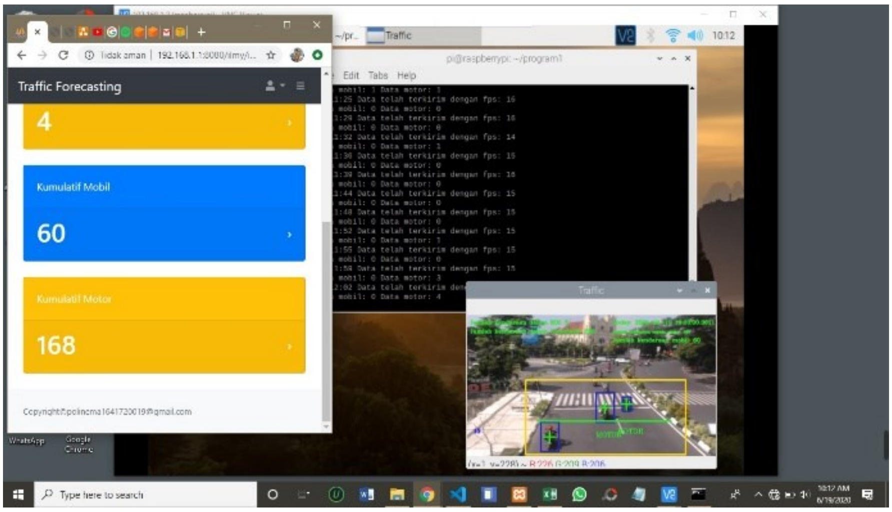

Fig. 7. Test System

The detection test on the haar cascade is carried out by comparing the application calculation and manual calculations for eight days of experiment with a period of 3 minutes to obtain a comparison and an estimate of the accuracy of the calculation. Table 3 is a detection test analysis.

Table 3. Haar cascade detection accuracy percentage

Date	Condition	Haar Cascade		Manual Calculation	
		Car	Motorbike	Car	Motorbike
Sat, June 13 2020	Sunny	26	72	30	61
Sun, June 14 2020	Cloudy	10	43	26	39
Mon, June 15 2020	Cloudy	11	12	14	22
Tue, June 16 2020	Cloudy	13	26	24	72
Wed, June 17 2020	Sunny	32	60	30	71
Thu, June 18 2020	Sunny	19	59	23	45
Fri, June 19 2020	Cloudy	11	39	17	60
Sat, June 20 2020	Cloudy	13	60	22	83
Total		135	371	186	453

Table 3 shows that the manual calculation and haar cascade on June 13 to 20, 2020, takes eight days and has similar patterns. It can be concluded that car vehicle detection is outnumbered by motorcycle detection. The total of the detected car is 135, while the total of manually calculated cars is 186. Similarly, the number of motorcycles is 371 for the detected and 453 for manual calculations.

Data communication testing is done by comparing the delivery time on the Raspberry Pi with the website application to determine the accuracy of data transmission times per 3 seconds. Can be seen in Table 4.

Table 4. Data communication

Date	Sending Time	Receiving Time
2020-06-14	10:31:04	10:31:04
2020-06-14	10:31:01	10:31:01
2020-06-14	10:30:57	10:30:57
2020-06-14	10:30:54	10:30:54

Table 5 shows the results of the FPS test on the Raspberry Pi. The result determines the computing capabilities of the Raspberry Pi and the webcam with a resolution of 427 x 2 40. The FPS test in Table 5 is obtained every 3 seconds. The highest FPS for sending data is 17, while the lowest is 13.

Table 5. FPS test

Time	Car Data	Motorcycle Data	FPS
10:10:02	0	0	14
10:10:05	2	1	16
10:10:09	0	0	15
10:10:12	1	1	18
10:10:15	0	1	14
10:10:19	0	0	12
10:10:24	0	0	15
10:10:27	0	2	17
10:10:31	0	3	13

This experiment is carried out by comparing the classification results on each vehicle. Can be seen in Table 6.

Table 6. Results forecasting applications

Alpha	Class category	Description
0,1	137	Car
0,2	137	Car
0,3	427	Motorcycle
0,4	427	Motorcycle
0,5	137	Car
0,6	137	Car
0,7	427	Motorcycle
0,8	427	Motorcycle
0,9	137	Car

Table 7 determines the suitability and accuracy of forecasting cars and motorbikes based on MAPE.

Table 7. Results of application MAPE

Alpha	Car		Motorcycle	
	MAPE	PE	MAPE	PE
	0		0	
0,3	2.5	2.4444	6.0377358490566	3.6655
	4.833333333		4.9586776859505	
	0		0	
0,6	1.25	1.75	3.0188679245283	3.403
	4.0000000000001		7.1900826446281	
	0		0	
0,9	5	4.9444	12.075471698113	7.4962
	9.833333333		10.413223140496	

In discussing and testing the best Alpha recommendations, we will discuss the field test result data with the best Alpha recommendations in the application to obtain a small percentage of errors in real time on cars and motorbikes. Can be seen in Table 8.

Table 8. Best alpha recommendations

Date	Car		Motorcycle	
	Alpha Recommendations	PE Values	Alpha Recommendations	PE Values
Saturday-June 13-2020	0.4	1.0302	0.5	1.3158
Sunday-14 June-2020	0.5	0	0.4	8.5205
Monday-15 June-2020	0.5	2.1073	0.5	3.2258
Tuesday-16 June-2020	0.5	0.3846	0.6	3.1089
Wednesday-June 17-2020	0.5	0.463	0.4	1.0338
Thursday-18 June-2020	0.4	1.7875	0.5	0.1788
Friday-19 June-2020	0.5	1.2712	0.5	0.1134
Saturday-20 June-2020	0.5	1.5306	0.5	1.0791

In Table 8, the application recommends the best Alpha to get the smallest PE real-time value. The smallest PE car was on Sunday, June 14, 2020, at alpha 0.5 with a PE value of 0; on Friday, June 18, 2020, at alpha 0.5 with a 0.1134.

After conducting tests and trials to compare the calculation results of the system with manual calculations, the DES method has been successfully applied to the real-time traffic density detection system. It can be used as information for service planning in overcoming traffic congestion by other related parties. Prediction results of the number of vehicles detected per 3 seconds and the number of times per 10 minutes with the category of cars and motorbikes on June 13 to 20, 2020, in the direction of alun-alun malang, get the best prediction results, including cars on June 14, 2020, with alpha 0.5, MAPE 0% (perfect prediction), and motorcycles on June 18, 2020, with alpha 0.5, MAPE 0.1134% (Perfect prediction). Then the most significant MAPE results for cars were on June 15, 2020, with alpha 0.5, MAPE 2.1073% (perfect prediction), and motorbikes on June 14, 2020, with alpha 0.4, MAPE 8.5205% (predictions are perfect). In the test results, the detection of haar cascade by comparing the application and manual calculations during the test is car detection at 72.58% and motorbike detection at 81.90%. This research results give good performance and can be used for traffic prediction systems in low-specification computers such as Raspberry Pi [31] or other single-board computers.

IV. Conclusion

The findings indicate that real-time traffic density detection based on an IoT system can benefit from the application of the Haar Cascade Classifier with the DES technique. Implementation and testing of a system for detecting and monitoring vehicles at a specific location in real-time using a webcam and Raspberry Pi. The results of the detection test showed that the haar cascade algorithm was effective in detecting vehicles, but motorcycle detection was better than car detection. The data communication testing showed that the system can transmit data accurately every 3 seconds. The FPS test on Raspberry Pi demonstrated that the system's computing capabilities were sufficient for processing data in real time. The forecasting application results showed that the system's accuracy in predicting vehicles was acceptable, with some small percentage errors. The study recommends using alpha values of 0.9 for cars and 0.6 for motorcycles to reduce the percentage error in real-time monitoring of vehicles. For future research can implement another exponential smoothing like holt-winters or triple exponential smoothing method.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

References

[1] M. Mohsin, Q. Abbas, J. Zhang, M. Ikram, and N. Iqbal, "Integrated effect of energy consumption, economic development, and population growth on CO2 based environmental degradation: a case of transport sector," Environ. Sci. Pollut. Res., vol. 26, no. 32, pp. 32824–32835, Nov. 2019.

[2] M. Sala and F. Soriguera, "Capacity of a freeway lane with platoons of autonomous vehicles mixed with regular traffic," Transp. Res. Part B Methodol., vol. 147, pp. 116–131, May 2021.

[3] D. Göhlich et al., "Integrated Approach for the Assessment of Strategies for the Decarbonization of Urban Traffic," Sustainability, vol. 13, no. 2, p. 839, Jan. 2021.

[4] D. Albalate and X. Fageda, "Congestion, Road Safety, and the Effectiveness of Public Policies in Urban Areas," Sustainability, vol. 11, no. 18, p. 5092, Sep. 2019.

[5] K. Pyatkova, A. S. Chen, D. Butler, Z. Vojinović, and S. Djordjević, "Assessing the knock-on effects of flooding on road transportation," J. Environ. Manage., vol. 244, pp. 48–60, Aug. 2019.

[6] C. K. Tang, "The Cost of Traffic: Evidence from the London Congestion Charge," J. Urban Econ., vol. 121, p. 103302, Jan. 2021.

[7] T. Afrin and N. Yodo, "A Survey of Road Traffic Congestion Measures towards a Sustainable and Resilient Transportation System," Sustainability, vol. 12, no. 11, p. 4660, Jun. 2020.

[8] C. Sun, S. Xu, M. Yang, and X. Gong, "Urban traffic regulation and air pollution: A case study of urban motor vehicle restriction policy," Energy Policy, vol. 163, p. 112819, Apr. 2022.

[9] C. Onyeneke, "Modeling the Effects of Traffic Congestion on Economic Activities - Accidents, Fatalities and Casualties," Biomed. Stat. Informatics, vol. 3, no. 2, p. 7, 2018.

[10] I. Kaddoura and K. Nagel, "Simultaneous internalization of traffic congestion and noise exposure costs," Transportation (Amst)., vol. 45, no. 5, pp. 1579–1600, Sep. 2018.

[11] V. Shepelev, S. Aliukov, K. Nikolskaya, and S. Shabiev, "The Capacity of the Road Network: Data Collection and Statistical Analysis of Traffic Characteristics," Energies, vol. 13, no. 7, p. 1765, Apr. 2020.

[12] D. J. Pedregal and J. R. Trapero, "Adjusted combination of moving averages: A forecasting system for medium-term solar irradiance," Appl. Energy, vol. 298, p. 117155, Sep. 2021.

[13] A. C. Adamuthe and G. T. Thampi, "Technology forecasting: A case study of computational technologies," Technol. Forecast. Soc. Change, vol. 143, pp. 181–189, Jun. 2019.

[14] J. F. Rendon-Sanchez and L. M. de Menezes, "Structural combination of seasonal exponential smoothing forecasts applied to load forecasting," Eur. J. Oper. Res., vol. 275, no. 3, pp. 916–924, Jun. 2019.

[15] M. B. A. Rabbani et al., "A Comparison Between Seasonal Autoregressive Integrated Moving Average (SARIMA) and Exponential Smoothing (ES) Based on Time Series Model for Forecasting Road Accidents," Arab. J. Sci. Eng., vol. 46, no. 11, pp. 11113–11138, Nov. 2021.

[16] M. A. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, and G. Asencio-Cortés, "A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting," Inf. Sci. (Ny)., vol. 586, pp. 611–627, Mar. 2022.

[17] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," Inf. Sci. (Ny)., vol. 484, pp. 302–337, May 2019.

[18] H. Tyralis, G. Papacharalampous, and A. Langousis, "Super ensemble learning for daily streamflow forecasting: large-scale demonstration and comparison with multiple machine learning algorithms," Neural Comput. Appl., vol. 33, no. 8, pp. 3053–3068, Apr. 2021.

[19] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," Int. J. Forecast., vol. 36, no. 1, pp. 75–85, Jan. 2020.

[20] S. Dhamodharavadhani and R. Rathipriya, "Region-Wise Rainfall Prediction Using MapReduce-Based Exponential Smoothing Techniques," in Advances in Big Data and Cloud Computing. Advances in Intelligent Systems and Computing, J. Peter, A. Alavi, and B. Javadi, Eds. Springer, 2019, pp. 229–239.

[21] Purwono, A. Ma'arif, and A. Wulandari, "Face Shape-Based Physiognomy in LinkedIn Profiles with Cascade Classifier and K-Means Clustering," in 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Oct. 2021, pp. 347–353.

[22] B. Fatima, A. R. Shahid, S. Ziauddin, A. A. Safi, and H. Ramzan, "Driver Fatigue Detection Using Viola Jones and Principal Component Analysis," Appl. Artif. Intell., vol. 34, no. 6, pp. 456–483, May 2020.

[23] A. Arunmozhi and J. Park, "Comparison of HOG, LBP and Haar-Like Features for On-Road Vehicle Detection," in 2018 IEEE International Conference on Electro/Information Technology (EIT), May 2018, pp. 0362–0367.

[24] Y. Morimoto, A. Masaya, and M. Ueki, "High-speed 3D shape measurement by one pitch phase analysis method using brightness values in small square area of single-shot image," Opt. Lasers Eng., vol. 113, pp. 38–46, Feb. 2019.

[25] H. Filali, J. Riffi, A. M. Mahraz, and H. Tairi, "Multiple face detection based on machine learning," in 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Apr. 2018, pp. 1–8.

[26] D. Kucharavy, D. Damand, and M. Barth, "Technological forecasting using mixed methods approach," Int. J. Prod. Res., pp. 1–25, Jul. 2022.

[27] G. Airlangga, A. Rachmat, and D. Lapihu, "Comparison of exponential smoothing and neural network method to forecast rice production in Indonesia," TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 17, no. 3, p. 1367, 2019.

[28] R. Mehta, J. Sahni, and K. Khanna, "Internet of Things: Vision, Applications and Challenges," Procedia Comput. Sci., vol. 132, pp. 1263–1269, 2018.

[29] R. Minerva, G. M. Lee, and N. Crespi, "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models," Proc. IEEE, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.

[30] A. P. Wibawa, Z. N. Izdihar, A. B. P. Utama, L. Hernandez, and Haviluddin, “Min-Max Backpropagation Neural Network to Forecast e-Journal Visitors,” in 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Apr. 2021, pp. 052–058.
[31] R. A. Asmara, B. Syahputro, D. Supriyanto, and A. N. Handayani, “Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2,” in 2020 4th International Conference on Vocational Education and Training (ICOVET), Sep. 2020, pp. 391–395.