Smart EV Navigation and Data Collection System for Tree Based Data Modeling Using IoT

Wirarama Wedashwara[1](✉), Heri Wijayanto[1], Andy Hidayat Jatmika[1], and I Wayan Agus Arimbawa[2]

[1] Department of Informatics Engineering, University of Mataram, Mataram, Indonesia
{wirarama,heri,andy}@unram.ac.id

[2] Department of Technology Management, Economic, and Policy, Seoul National University, Seoul, Republic of Korea
arimbawa@snu.ac.kr

Abstract. Machine learning for autonomous can be done by recording the displacement and condition of the vehicle through manual control by humans and modeling the data. The research proposes designing a data collection system for tree-based data modeling on Internet of Things (IoT) based autonomous electrical vehicles (EV). The system consists of four ESP32 cameras with servos mounted on the left, right side of the car mirror, front (dashcam), and rear. The system is also equipped with an Arduino Nano connected to GPS, a gyroscope, and four proximity sensors. Arduino nano is connected via serial software to the Wemos D1 mini, which is connected to a relay module to control lights and wipers and is equipped with an LDR sensor. Data collected via the internet (wifi) will be formed in tree-based data modeling for future genetic programming machine learning algorithms. System evaluation includes Quality of Service (QoS) data communication, statistical data collected, and electrical IoT devices built. Based on testing using an intelligent car chassis in an environment still affordable by wifi, it produces an average delay of 0.02 s and a PDR of 99.87%. The highest correlation matrix archived as 0.872 for longitude, latitude, and gyro data in detecting vehicle turns. The electricity evaluation result consists of average power consumption of 0.344 W for the ESP32 camera, 0.663 W for the Arduino nano, and 0.291 W for the Wemos d1 mini. In the future, testing will be carried out using an actual EV on a real track and in data communication outside of wifi.

Keywords: Internet of Things · Smart Electrical Vehicle · Genetic Programming

1 Introduction

Machine learning for autonomous can be done by recording the displacement and condition of the vehicle through manual control by humans and modeling the data [1, 2]. The research proposes designing a data collection system for tree-based data modeling [3] on the Internet of Things (IoT)-based automated electrical vehicles (EVs). The developed system can also simultaneously function as navigation for manual controls to obtain data when mounted on an actual EV [4].

I G. P. Suta Wijaya et al. (Eds.): MIMSE-I-C 2022, ACSR 102, pp. 130–141, 2022.
https://doi.org/10.2991/978-94-6463-084-8_13

Electric vehicles are environmentally friendly because they can be supplied with renewable energy sources such as solar power [5]. The development of electric vehicles requires considerable costs, especially to realize large-powered vehicles capable of lifting heavy loads like humans [6]. Electric vehicles are a popular research topic in the automotive industry to academia [7].

The system consists of four ESP32 cameras with servos mounted on the left, right side of the car mirror, front (dashcam), and rear. The system is also equipped with an Arduino Nano connected to GPS, a gyroscope, and four proximity sensors. Arduino nano is connected via serial software to the Wemos D1 mini, which is connected to a relay module to control lights and wipers and is equipped with an LDR sensor. Data collected via the internet (wifi) will be formed in tree-based data modeling for future genetic programming machine learning algorithms [8]. Genetic programming was chosen because it has a concise data modeling structure and is easily distributed through the MQTT protocol used in this system.

System evaluation includes Quality of Service (quality of service) data communication, statistical data collected, and electrical IoT devices built. The test was carried out using a smart car chassis in an environment that was still affordable by wifi. The data collected was tested through statistics for variables GPS (longitude, latitude), gyro(x,y,z), and four-way proximity sensor. When there is a turn, image data is also related to sensor data, especially the proximity sensor. Electrical testing was carried out with a 3.7 V 18650 battery which was stepped up to 5 V to obtain device power consumption data.

2 Related Research

Research on automatic control of IoT-based electric cars with a cyber-attack protection approach has been conducted [9]. Research related to monitoring the behavior of IoT-based electric vehicles has also been carried out [4]. Both studies focus on control monitoring, while the proposed research focuses more on the remote control for machine learning using genetic programming.

The design of an automatic car using computer vision and IoT has been carried out [10]. Research by conducting an automatic car with road features has also been carried out [11]. Both studies use the camera as machine learning for automatic control. The proposed research uses cameras only to assist with remote control, and machine learning is carried out based on manual controls recorded and modeled by genetic programming.

3 Review of the Presented System

An overview of the system is shown in Fig. 1. The system consists of four parts: sensor nodes, camera nodes, relay module nodes, and Online Analytics Processing (OLAP) Server [12]. The sensor node functions to record GPS coordinates via the GPS module and the Gyro sensor to record the tilt orientation in x, y, and z format. The camera node records photos and videos from four sides: front, back, left, and right.

The camera node is equipped with an infrared obstacle sensor to record the distance based on reflected light. The relay module controls lights, wipers, or other devices. The

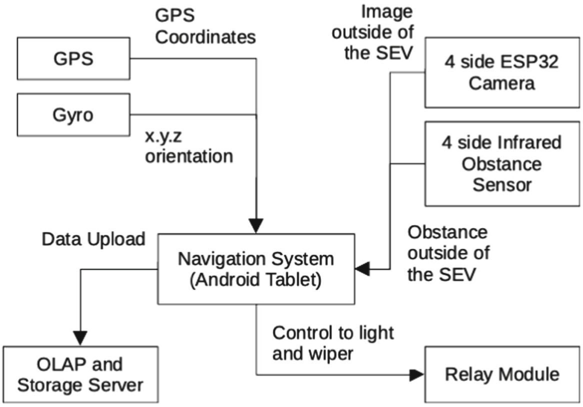

Fig. 1. General View of The System

OLAP and storage server sections are used to receive, store and process data according to the system's purpose, namely tree-based data modeling.

The flow chart of the system presented is shown in Fig. 2. The system starts by providing IoT Devices, Android tablets, and the system's interface. The first process is continuous data collection, namely GPS and Gyro coordinates. If the GPS coordinates change, the system will indicate a car's movement and mark the direction.

As the car moves, event-driven data is recorded, namely infrared obstacle sensors to mark nearby objects. If an object is found nearby, it is marked by an object around and activates the camera. Through the camera, an image of the object is obtained according to the direction it is detected.

After the data is collected, the tablet is sent to be uploaded to the internet. Sensor data is stored in CSV format via the MQTT protocol [13]. At the same time, the image data from the camera is stored directly in JPEG format. The collected data are combined as a unitary agency, and statistical analysis results are obtained in mean, median, submedian, and standard deviation. The last process is forming a data model as a tree based on the results of statistical conclusions, especially the correlation matrix, which will be discussed in the results section.

4 Detail of the Presented System

4.1 System Navigation Interface

The system interface was developed using Apache Cordova (HTML5 and Javascript) so that it is easier to communicate synchronizing with the web server [14], which in the research is used Raspberry Pi, as shown in Fig. 3. The interface aims to display a four-way camera based on gyro conditions, detecting proximity sensor objects and car movement (reverse parking); displays numerical sensor data, including GPS navigation, into a touchscreen control interface. Store and analyze data transmitted by smart cars over the internet; become an MQTT broker. Genetic programming algorithm is planned to map the condition of the sensor in the form of a tree and developed based on python, including its web server (flask based) [15].

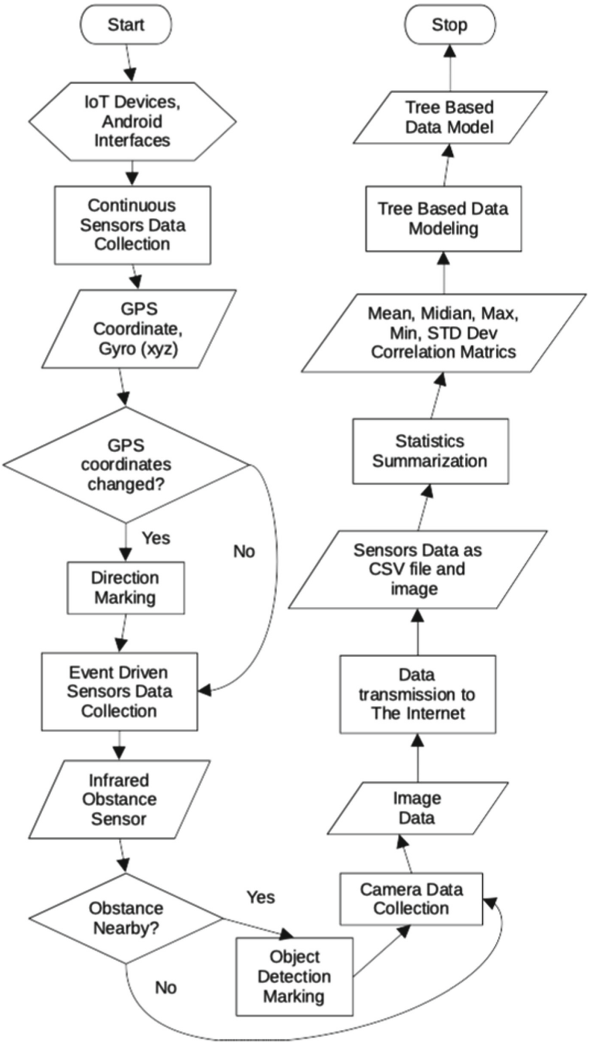
Start
IoT Devices, Android Interfaces
Continuous Sensors Data Collection
GPS Coordinate, Gyro (xyz)
GPS coordinates changed?
Yes
No
Direction Marking
Event Driven Sensors Data Collection
Infrared Obstance Sensor
Obstance Nearby?
Yes
No
Object Detection Marking
Camera Data Collection
Image Data
Data transmission to The Internet
Sensors Data as CSV file and image
Statistics Summarization
Mean, Midian, Max, Min, STD Dev Correlation Matrics
Tree Based Data Modeling
Tree Based Data Model
Stop

Fig. 2. Flowchart of The System

4.2 IoT Devices

The circuit for the sensor node is shown in Fig. 4. The sensor node consists of two micro-controllers, Arduino Nano and Wemos D1 Mini, connected via serial software. Both micro-controllers are supplied with 5 V power. GPS, Gyro, infrared obstacle sensor connected with Arduino nano. The relay module and LDR sensor are connected to the Wemos d1.

The realization of the PCB is shown in Fig. 5. In the realization, two PCBs measuring 7 × 5 cm are used with serial software that can relate to a cable connector. Each PCB also has a power cable to connect to a 5 V source. Wemos functions to connect to the

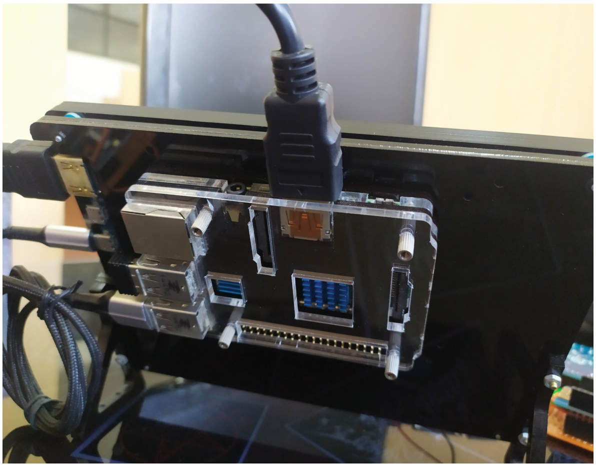

Fig. 3. Raspberry Pi for The Web Interface and Uploader

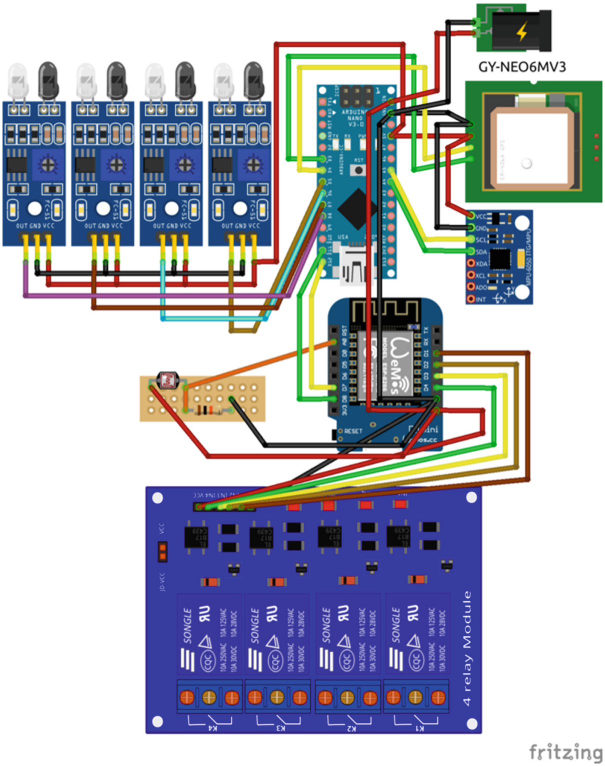

Fig. 4. Circuit of Sensor Node

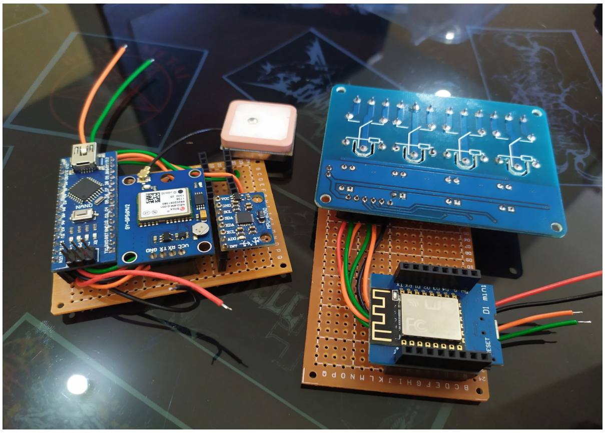

Fig. 5. PCB of Sensor Node

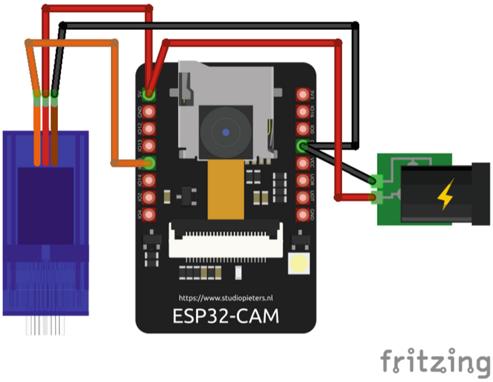

Fig. 6. Circuit of Camera Node

internet and control relays based on Android tablet input and light sensors (for lights). Arduino nano is connected to a GPS module, gyro, and 4 proximity sensors that require higher computing.

The camera node circuit is shown in Fig. 6. The circuit is simple because the camera is already installed on the ESP32. The serial pins on the ESP32 are used for servos and 5 V sources only.

The PCB realization for the camera node is shown in Fig. 7. The PCB realization is a half size, 7 × 5 cm. The camera node can function as a video streaming service accessed via the web by the android interface. The servo can be controlled manually or automatically detects nearby objects. Detection of objects using OpenCV (Haar Cascade) is on the python server but is not discussed in this paper.

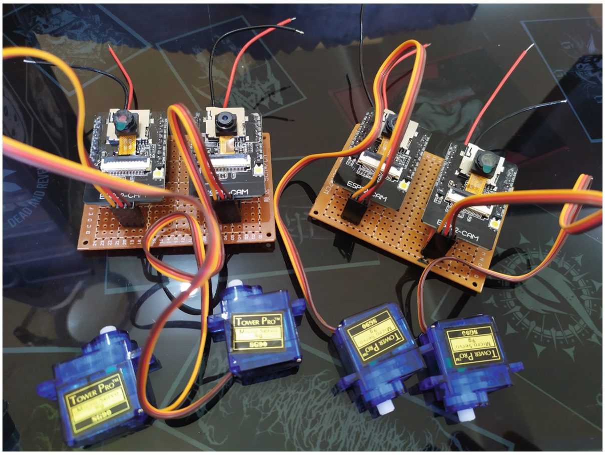

Fig. 7. PCB of Camera Node

5 Result and Evaluation

The test was carried out not using a full-size car but using a small smart car chassis [16]. Control of the smart car chassis is carried out via the internet remotely. The route involves straight movement, turning right and left at 45 and 90 degrees, respectively.

5.1 QoS Evaluation

The results of the quality-of-service evaluation are shown in Table 1. The test is based on packet delivery ratio (PDR) and delay [17]. The test is divided into 7 parts and 3 data transmission methods: serial software [18], WLAN, and the Internet.

Software serialization occurs via cable between Arduino nano to Wemos D1 and produces the best results. WLAN occurs from Wemos d1 and ESP32 devices, which

Table 1. QoS Evaluation

Direction	**Method**	**PDR (%)**	**Delay (s)**
Arduino Nano to Wemos D1	Software Serial	99.992	0.001
Wemos D1 to RPi	WLAN	99.796	0.026
ESP32 (left) to Rpi	WLAN	99.872	0.021
ESP32 (right) to Rpi	WLAN	99.883	0.022
ESP32 (front) to Rpi	WLAN	99.781	0.024
ESP32 (back) to Rpi	WLAN	99.871	0.024
Rpi to Server	Internet	98.783	0.218
		99.711	0.048

Table 2. Electricity Evaluation

Devices	Current (A)	Voltage (V)	Power (W)
Arduino Nano	0.433	3.029	2.165
Wemos D1	0.223	3.223	1.115
Arduino Nano + Wemos D1	0.633	2.893	3.165
ESP32	0.257	3.176	1.285
Rpi + Monitor	0.898	2.789	4.490
	0.489	3.022	2.444

have a built-in wifi module. Data transmission over the Internet was done via raspberry pi and had the worst results but still on a normal scale. The overall results show an average of 99,711 for PDR and 0.048 for the delay.

5.2 Electricity Evaluation

The electrical evaluation is shown in Table 2. The evaluation is divided into 5 parts. Tests were carried out with a 3.7 V 18650 battery with a 1 A limit using the diode on the TP4056 module. The data collected using IoT devices from previous research [19].

Power is calculated using the assumed 5 V voltage step-up from 3.7 V. Rpi with its monitor shows the highest results, followed by a combination of Arduino nano and Wemos d1, which is supplied by a single 5 V source. The lowest results were obtained on Wemos d1. The overall average produces 0.489 A for current, 3.022 V for voltage, and 2,444 W for power.

5.3 Data Evaluation

The evaluation of the correlation matrix [20] is shown in Table 3. The correlation matrix is carried out only with continuous variables from three sensors: GPS, Gyro, and infrared obstacle. GPS consists of latitude and longitude coordinates that indicate the car's displacement. The GyroGyro consists of a tilt indicated by X, Y, and Z. The infrared obstacle sensor consists of four directions: left, right, front and rear.

The correlation matrix results show that GPS highly correlates with the Gyro, especially the X and Y angles. The Z of Gyro has a low correlation because the car only moves on the X and Y axes. Z tilt only occurs when a shock makes the car lift or fall. Correlation with distance occurs because the object is near when making a turn or a long distance when in front of an empty car far away.

GPS data modeling is shown in Table 4. The data recorded in the system is the latitude and longitude location according to the GPS module and displacement from the initial position. The data is standardized for processes like the correlation matrix discussed earlier. The test is carried out in four movements, namely turning left and right at 90 and 45, respectively. Through this modeling data, we get variations in coordinate shifts in general and can be processed as continuous data.

Table 3. Correlation Matrix

	GPS Lat	GPS Long	Gyro X	Gyro Y	Gyro Z	dist L	dist R	dist F	dist B
GPS Lat	1	0.678	0.457	0.446	0.189	0.245	0.238	0.129	0.162
GPS Long	0.678	1	0.781	0.453	0.156	0.215	0.217	0.134	0.153
Gyro X	0.457	0.781	1	0.435	0.241	0.231	0.242	0.152	0.121
Gyro Y	0.446	0.453	0.435	1	0.218	0.102	0.092	0.034	0.035
Gyro Z	0.189	0.156	0.241	0.218	1	0.001	0.001	0.001	0.001
dist L	0.245	0.215	0.231	0.102	0.001	1	0.219	0.032	0.043
dist R	0.238	0.217	0.242	0.092	0.001	0.219	1	0.026	0.021
dist F	0.129	0.134	0.152	0.034	0.001	0.032	0.026	1	0.022
dist B	0.162	0.153	0.121	0.035	0.001	0.043	0.021	0.022	1

Table 4. GPS Data Modeling

Lat	Long	Movement		Potition
		Lat	Long	
8.574056	116.102543	0	0	Initial
8.573866	116.102318	−0.00019	−0.00023	Turn Left 90
8.573505	116.102446	−0.00036	0.00012	Turn Left 45
8.57395	116.103004	0.0004	0.00056	Turn Right 90
8.573547	116.102945	−0.00040	−5.89999E−05	Turn Right 45

Gyro data modeling is shown in Table 5. The data recorded on the system is the same as GPS data modeling, namely the position according to the gyro sensor and its changes. Changes in X and Y are permanent depending on the direction the car moves. So that data modeling is carried out to normalize changes in the direction of car movement. The Z axis only changes if the car vibrates.

5.4 Tree Data Modeling

The tree-based data model is shown in Fig. 8. Tree Data Modeling is based on the correlation matrix, GPS and Gyro data modelling. The label consists of the car's displacement direction, turning right and left, 45 and 90 degrees, respectively. GPS data consists of latitude and longitude. Gyro only consists of X and Y based on the results of

Table 5. Gyro Data Modeling

X	Y	Z	Movement			Potition
			X	Y	Z	
0.002	0.001	0.001	0	0	0	Initial
−0.231	−0.132	0.002	−0.233	−0.133	0.001	Turn Left 90
−0.128	−0.102	0.001	0.103	0.030	−0.001	Turn Left 45
0.253	0.162	0.002	0.381	0.264	0.001	Turn Right 90
0.131	0.156	0.001	−0.122	−0.006	−0.001	Turn Right 45

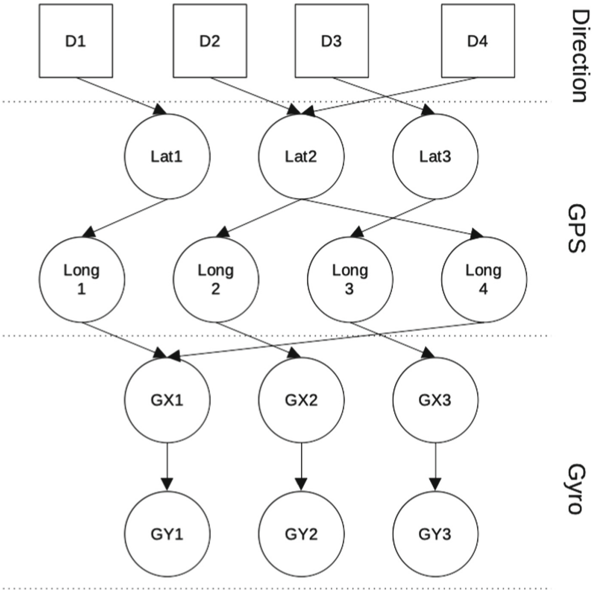

Fig. 8. Tree Data Modeling

data correlation. Results with proximity are merged into a single node, such as Lat2, a combination of −0.00036 and −0.00040 values. Another combination is GX1 which is a combination of −0.233 and −0.122 values.

6 Conclusion

Based on testing using a smart car chassis in an environment still affordable by wifi, it produces an average delay of 0.02 s and a PDR of 99.87%. Collected 15268k data with the highest correlation matrix between 0.872 for longitude, latitude, and gyro data in detecting vehicle turns. When there is a turn, image data is also related to sensor data, especially the proximity sensor. Electrical testing was carried out with a 3.7 V 18650 battery, which was up to 5 V. The test resulted in average power consumption of 0.344 W for the ESP32 camera, 0.663 W for the Arduino nano, and 0.291 W for the Wemos d1 mini. Tree Data Modeling is based on the correlation matrix, GPS and Gyro data modelling. The label consists of the car's displacement direction, turning right and left, 45 and 90 degrees, respectively. GPS data consists of latitude and longitude. Gyro only consists of X and Y based on the results of data correlation. In the future, testing will be carried out using an actual EV on a real track and in data communication outside of wifi.

References

1. H. Hejazi and L. Bokor.: A survey on the use-cases and deployment efforts toward converged internet of things (IoT) and vehicle-to-everything (V2X) environments. Acta Technica Jaurinensis 15(2), 58–73 (2022).
2. B. Vaidya and H. T. Mouftah.: IoT applications and services for connected and autonomous electric vehicles. Arab J Sci Eng 45(4), 2559–2569 (2020).
3. N. Zhang and J. S. Simonoff.: Joint latent class trees: A tree-based approach to modeling time-to-event and longitudinal data. Stat Methods Med Res 31(4), (2022).
4. S. Echavarr, R. Meia-Gutiérrez, and A. Montoya.: Development of an IoT platform for monitoring electric vehicle behaviour. In: Workshop on Engineering Applications, pp. 363–374 (2020).
5. G. Z. de Rubens.: Who will buy electric vehicles after early adopters? Using machine learning to identify the electric vehicle mainstream market. Energy 172, 243–254 (2019).
6. H. Zhang and X. Lu.: Vehicle communication network in intelligent transportation system based on Internet of Things. Comput Commun 160, 799–806 (2020).
7. N. Chowdhury, C. A. Hossain, M. Longo, and W. Yaici.: Optimization of solar energy system for the electric vehicle at university campus in Dhaka, Bangladesh. Energies (Basel) 11 (9), 2433 (2018).
8. L. W. Santoso, B. Singh, S. S. Rajest, R. Regin, and K. H. Kadhim.: A Genetic Programming Approach to Binary Classification Problem. EAI Endorsed Transactions on Energy Web 8(31) (2021).
9. M. M. Rana.: IoT-based electric vehicle state estimation and control algorithms under cyber-attacks. IEEE Internet Things J 7(2), 874–881 (2019).
10. I. Ahmad and K. Pothuganti.: Design & implementation of real time autonomous car by using image processing & IoT. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 107–113 (2020).
11. B. Padmaja, P. V. N. Rao, M. M. Bala, and E. K. R. Patro.: A novel design of autonomous cars using IoT and visual features. In: 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference, pp. 18–21 (2018).
12. O. Moscoso-Zea, J. Castro, J. Paredes-Gualtor, and S. Lujan-Mora.: A Hybrid Infrastructure of Enterprise Architecture and Business Intelligence Analytics for Knowledge Management in Education. IEEE Access 7 (2019).
13. Y. Chen, R. CUI, X. ZHU, Y. ZHOU, Z. LIN, and M. LIU.: Transmission earthquake waveform using IOT MQTT protocol. Progress in Geophysics 35(4), 1232–1237 (2020).
14. F. Sukmana and F. Rozi.: Digitalization In: Scanning and Remote Shutdown Host of Computer Using Apache Cordova Framework. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) 4(2) (2019).

15. N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja.: Implementation of database using python flask framework. International Journal of Engineering and Computer Science 8(12) (2019).
16. A. Bukola.: Development of an Anti-Theft Vehicle Security System using GPS and GSM Technology with Biometric Authentication. Int J Innov Sci Res Technol 5(2) (2020).
17. N. Varyani, Z.-L. Zhang, and D. Dai.: QROUTE: an efficient quality of service (QoS) routing scheme for software-defined overlay networks. IEEE Access 8, 104109–104126 (2020).
18. T. W. Schubert, A. D'Ausilio, and R. Canto.: Using Arduino microcontroller boards to measure response latencies. Behav Res Methods 45(4) (2013).
19. W. Wedashwara, I. W. A. Arimbawa, A. H. Jatmika, A. Zubaidi, and T. Mulyana.: IoT based Smart Small Scale Solar Energy Planning using Evolutionary Fuzzy Association Rule Mining. In: 2020 International Conference on Advancement in Data Science, pp. 1–6. E-learning and Information Systems (ICADEIS) (2020).
20. R. P. Ghosh, B. Mallick, and M. Pourahmadi.: Bayesian Estimation of Correlation Matrices of Longitudinal Data. Bayesian Anal 16(3) (2021).