

# The CAN FD Vehicle Network System with Machine Learning and Scheduling Algorithms

Yung-Hoh Sheu, Cheng-Yo Huang, Chen-Yu Yang, and Yi-Hong Lin

**Abstract**—The controller area network with flexible data-rate (CAN FD) inherits the primary features of a controller area network (CAN); thus, exploring the possibility of establishing a hybrid CAN and CAN FD network is essential. To develop the CAN FD network effectively, this study proposed a machine learning K-means data clustering method. The K-means method algorithms, the squared Euclidean distance was used to cluster CAN FD data. The results showed that the proposed system was compatible with current CAN vehicle networks. Experiments on processing five data quantities of CAN FD data verified that the K-means algorithms could effectively reduce the data loss rate of the CAN FD network by changing the priority of various CAN FD data according to the clustering result. Specifically, given CAN FD arbitration phase rate = 1 Mbps, for the data phase rate = 2 and 4 Mbps, the data loss rates were reduced by 7.49% and 8.34%, respectively, by using the squared Euclidean distance algorithm.

**Index Terms**—Controller area network, controller area network with flexible data-rate, machine learning, K-means.

## I. INTRODUCTION

The rapid development of the electronics, semiconductor, and communications industries has enhanced the requirements for intelligent vehicle electronics. Consumers wish to drive cars that are dynamic, comfortable, economical, and exhibit in-car entertainment facilities. To improve product competitiveness, automakers incorporate additional electronic control systems into automobiles such as electronic stability program (ESP) and passive entry passive start (PEPS). Therefore, the vehicle network is no longer limited to a low-speed interface. The increasing bandwidth requirements pose challenges for designers, and the proposed solutions require that vehicle networks feature high-speed data transmission. The concept of a conventional controller area network (CAN) [1] similarly requires high-speed data transmission. However, because the maximum transmission rate of a CAN is 1 Mbps (usually the actual maximum usage rate of the vehicle CAN system is 500 Kbps), the substantial increase in electronic control units (ECU) has resulted in a sharp increase in the CAN busload rate, leading to insufficient network bandwidth and impairing the reliability and immediacy of message transmission.

Machine learning is a multidisciplinary subject that emerged 20 years ago. It involves numerous disciplines such as probability theory, statistics, approximation theory, convex analysis, and computational complexity theory [2], [3]. The primary methods are K-means, fuzzy, linear regression, and neural networks.

Currently, machine learning is primarily used in the analysis of big data and cloud computing [4]-[9]. However, the development of unmanned vehicles requires additional sensors and vehicle electronics; therefore, integrating machine learning and artificial intelligence (AI) software to control vehicle hardware is a crucial topic for future development. Scant research has been conducted to integrate machine learning approaches into controller area network with flexible data-rate (CAN FD) scheduling methods. Conversely, introducing machine learning into an embedded system would definitely increase the R&D costs. To thoroughly improve the R&D efficiency of embedded software, numerous research institutes or business operators promote the standardization of interfaces and the circulation and reuse of machine learning software.

Wisely applying vehicle networks and machine learning technology to embedded-system simulations and tool development can improve the R&D efficiency of machine learning software. Therefore, based on the K-means clustering method, this study developed a new CAN FD vehicle communication network message scheduling system. The system was designed in line with the current vehicle network requirements and could be used in the verification and testing of On-Board Diagnostics (OBD) II simulators and analyzers for industrial vehicles. This study developed an embedded system based on NXP LPCXpresso54618 and 32-bit Cortex-M4 microcontrollers; additionally, this study explored and established a CAN FD scheduling method, which was incorporated into a new generation of machine learning. Previously, industries and academic units rarely adopted a CAN FD method. This study added the CAN FD function to vehicle network ECU nodes (engine revolutions per minute (RPM), vehicle speed, oxygen sensor voltage, and engine coolant temperature) and developed two different CAN/CAN FD gateways (embedded OBD-II CAN/CAN FD gateway and embedded diagnostic CAN/CAN FD gateway) to verify the relevant functions and practicality of the system designed in this study.

## II. SYSTEM PRINCIPLES

### A. Machine Learning

K-means clustering is an unsupervised learning method that classifies data into clusters. Specifically, it trains data without predefined labels. K-means is the most basic

Manuscript received March 25, 2019; revised November 11, 2019.

All authors are with the Department of Computer Science and Information Engineering, National Formosa University, Huwei, Yunlin, Taiwan (e-mail: yhsheu@nfu.edu.tw; 10763104@gm.nfu.edu.tw, 10563113@gm.nfu.edu.tw, 10763105@gm.nfu.edu.tw).

method of all partition clustering methods and is applied in the vehicle communication [10].

The K-means clustering method first specifies the number of clusters and gradually reduces the error value of the objective function by repeated operations to obtain the final clustering result. Specifically, K-means minimizes the squared distance difference between each data point and the cluster center. Subsequent processing, such as data compression and classification, is deployed according to the cluster centers. In this study, the squared Euclidean distance of K-means uses the following method to determine distance.

Equation (1) is the formula of the squared Euclidean distance. The squared Euclidean distance is used to calculate the minimum distance between points in each cluster and the cluster center. The cluster center is the average value of all the points in each cluster.

$$d(x, y) = \sum_{i=1}^n (x_i - y_i)^2 \quad (1)$$

The process of K-means method used in this study is as follows:

- 1) Step 1: Set the data to contain K clusters and K initial cluster centers.
- 2) Step 2: Assign the samples in the sample set to the nearest cluster according to a minimum distance strategy.
- 3) Step 3: Use the sample mean in each cluster as the new cluster center.
- 4) Step 4: Repeat steps 2 and 3 until the cluster center does not change.
- 5) Step 5: This process is completed. K clusters are obtained.

### B. CAN FD Bus Communication Protocol

CAN FD is an extension of the original CAN protocol specified in ISO 11898-1. Bosch developed the protocol in 2011 and released it in 2012.

CAN FD inherits the primary features of CAN. CAN adopts a two-wire serial communication protocol. CAN exhibits high security because of numerous characteristics including non-destructive arbitration technology, decentralized real-time control, and reliable error handling and detection mechanisms. However, CAN's bandwidth and data length are limited. CAN FD compensates for CAN's limitations of bandwidth and data length. The two primary differences between CAN FD and CAN are the following:

#### 1) Flexible rate

CAN FD uses two bit rates. From the bit rate switch (BRS) field in the control field to the ACK field (including cyclic redundancy check (CRC)), the CAN FD uses a variable rate, and the remaining part uses the original CAN rate. Each rate exhibits a set of bit timings to define the register. In addition to using different bit time units TQ, the allocation ratio of each bit time segment can also be different.

#### 2) New data length

CAN FD substantially expands the data length. The data length code (DLC) supports up to 64 bytes. The data length is the same as the original CAN when the DLC is less than or equal to 8 bytes. However, non-linear growth occurs if

the DLC is ever larger than 8 bytes. Therefore, the maximum data length can be up to 64 bytes, as shown in Table I.

TABLE I: ENCODING SCHEME OF DLC

	Byte number	Code of data length			
		DLC3	DLC2	DLC1	DLC0
ISO 11898-1	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	0	1	1
	4	0	1	0	0
	5	0	1	0	1
	6	0	1	1	0
	7	0	1	1	1
Proposal for Additional Codes In CAN FD	8	1	0	0	0
	12	1	0	0	1
	16	1	0	1	0
	20	1	0	1	1
	24	1	1	0	0
	32	1	1	0	1
	48	1	1	1	0
	64	1	1	1	1

## III. SYSTEM DESIGN

### A. System Architecture

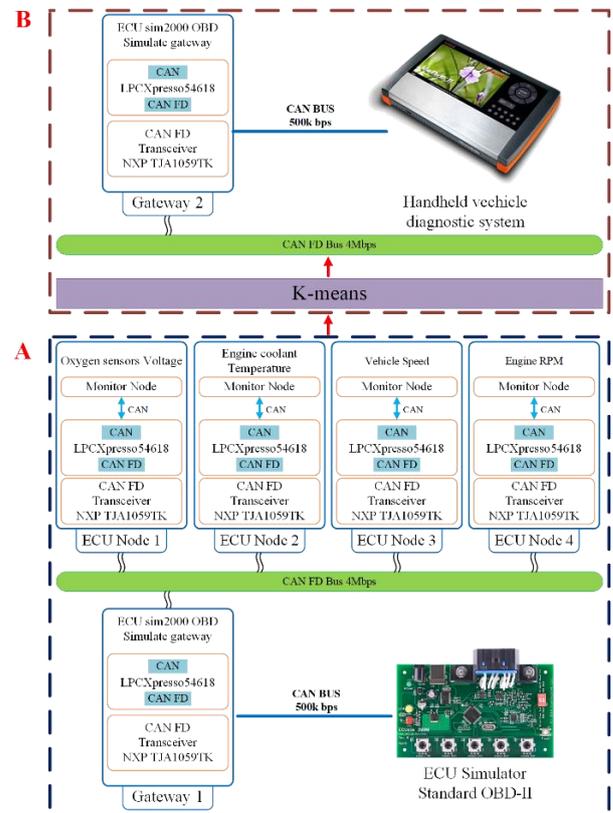


Fig. 1. System architecture.

Fig. 1 shows the system architecture of this study. In this study, the vehicle network can be divided into Part A—the CAN FD/CAN gateway design and the OBD-II CAN FD network design and Part B—the design of K-means clustering and the verification of OBD-II CAN FD network. The ECU sim2000 OBD simulator in this study used four parameters, namely engine RPM, vehicle speed, oxygen sensor voltage, and engine coolant temperature, to simulate the data of vehicles' CAN FD ECUs.

In Part A, data of the engine RPM, vehicle speed, oxygen sensor voltage, and engine coolant temperature from a sim2000 OBD simulator were converted to CAN FD signals by using CAN/CAN FD Gateway 1. Four ECU nodes were associated with the four parameters. The four ECU nodes exhibited individual CAN digital instruments. Each of the four ECUs was also a gateway, which could convert CAN FD signals into CAN signals and transmit them to CAN digital instruments for subsequent verification.

In Part B, K-means machine learning was applied to the design of data clustering. According to the data volume of CAN FD, the proposed K-means clustering method classified and changed the CAN FD identifier. The K-means method also reduced the data loss rate of the CAN FD network. CAN FD/CAN gateway 2 converted the ECU data in the CAN FD network into CAN signals and transmitted them to a commercial VeDiS II handheld vehicle diagnostic system to verify the OBD-II data in the CAN FD network.

### B. Design of K-means Clustering

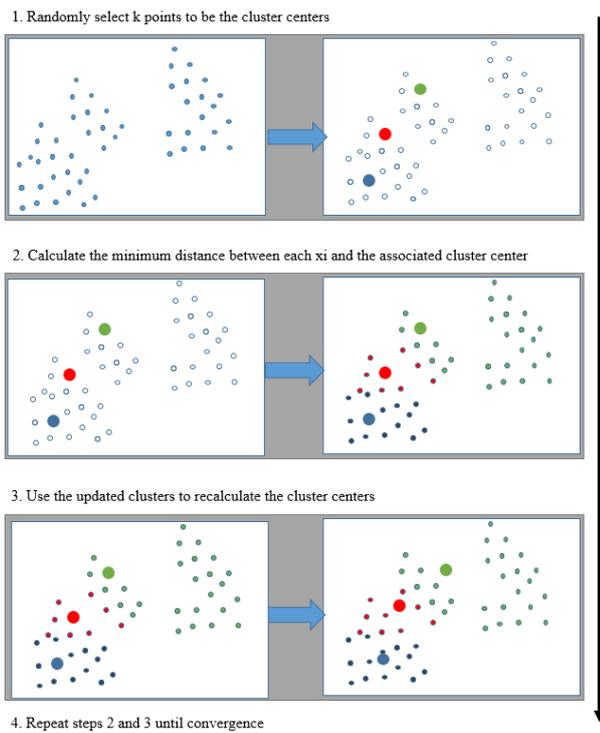


Fig. 2. Process of K-means data clustering.

The data clusters were first generated by a K-means method, and then the data were introduced into the CAN FD network. MATLAB2016a was used to cluster CAN FD data with the K-means method, and Visual Studio C# (C Sharp) was used to generate random CAN FD data and calculate the data loss rate of CAN FD. Specifically, through the C# graphical user interface, the data length, data cycle, and data quantity of CAN FD were generated by the program field. Accordingly, uncluttered random CAN FD data were generated. In addition to outputting data, the system can calculate the data loss rate before and after clustering. Fig. 2 displays the K-means data clustering process.

For clustering CAN FD data by K-means, the initial data were generated using a self-designed CAN FD data generator that applied the squared Euclidean distance and the sum of absolute differences. The CAN FD data in

various data quantities were clustered by the two methods; the results were compared. The data quantities for this study were 100, 200, 300, 400, and 500 pieces of data.

The software architecture of K-means clustering can be divided into three parts—pre-processing, solution, and post-processing.

#### 1) Pre-processing

A self-designed CAN FD data generator generated random data in the CAN FD data format. Only the data length of new CAN FD data was randomly assigned.

#### 2) Solution

Five different CAN FD data quantities were divided into 10 clusters by the squared Euclidean distance and the sum of absolute differences. The two methods were also used to find the suitable number of iterations for each data quantity. Subsequently, the results of the 10 clusters were associated with different CAN FD ID priorities. The CAN FD ID allocation was that data of short cycle and short data length were assigned high priority, and data of long cycle and long data length were assigned low priority.

#### 3) Post-processing

MATLAB was used to export text and graphical results and to compare simulation results. The results were converted and exported in XMT format.

##### a) CAN FD Random Data Generator

The CAN FD random data generator exhibited functions including the generation of random data of CAN FD, the calculation of CAN FD data loss rate, and the calculation of CAN FD network bandwidth load. The configuration of these three functions is introduced in the following sections.

The random parameter allocation for the CAN FD random data generator was assigned as follows:

- 1) Data cycle: 50–100 ms.
- 2) Data length: 12, 16, 20, 24, 32, 48, and 64 Bytes.
- 3) Data ID: 1–2015.

The data loss rate is calculated as (2).

$$\text{Data lost} = \frac{\text{Data Total Receive Quantity}}{\text{Data Total Transmission Quantity}} \quad (2)$$

The total number of data packets transmitted was derived from the sum of the total transmission time divided by each data cycle of CAN FD. Table II lists the number of messages in the CAN FD network generated by the CAN FD random data generator for data quantities of 100, 300, and 500 pieces of data.

TABLE II: NUMBER OF MESSAGES FOR FIVE CAN FD DATA QUANTITIES IN THE CAN FD NETWORK

CAN FD data quantity	Number of messages/s
100	1419
300	4145
500	6965

In Part 3, the bandwidth load rate of CAN FD network is calculated as (3). In (3),  $C_m$  is the worst-case transmission time (WCTT) of a CAN FD message.  $T_m$  is the cycle of each CAN FD message [11].

$$\text{Busload} = \sum_m \frac{C_m}{T_m} \quad (3)$$

The calculation of  $C_m$  is as shown in (4).  $T_s$  is the transmission time of the arbitration phase in the CAN FD frame, and the maximum transmission bandwidth is 1 Mbps.  $T_f$  is the transmission time of the data phase in the CAN FD frame, and the maximum transmission bandwidth is 8 Mbps.

$$C_m = T_s + T_f \tag{4}$$

Equation (6) shows the calculation of  $T_s$ .  $t_x$  is the bit rate of the arbitration phase. The maximum bit rate is 1 Mbps.

$$T_s = \left[ \frac{(SOF + ID + r1 + IDE + EDL + r0 + \frac{BRS}{2} + \frac{CRCdel}{2}) * 1.2}{t_x} + \frac{ACK + DEL + EOF + IFS}{t_x} \right] \tag{5}$$

In the CAN FD frame, two options exist for the number of bits in the CRC field depending on data length and type. If the data length is less than or equal to 16 bytes,  $T_f$  is calculated as (6).  $t_y$  is the bit rate of the data phase. The maximum bit rate is 8 Mbps.

$$T_f = \frac{[(D_f + \frac{BRS}{2} + ESI + DLC + \frac{CRCdel}{2}) * 1.2] + CRC_{17} + 5}{t_y} \tag{6}$$

If the data length is higher than 16 bytes,  $T_f$  is calculated as (7).

$$T_f = \frac{[(D_f + \frac{BRS}{2} + ESI + DLC + \frac{CRCdel}{2}) * 1.2] + CRC_{21} + 6}{t_y} \tag{7}$$

Table III lists the names of bit fields in the CAN FD frame and the lengths of bit fields.

According to the algorithm steps and (1) in Section 2.1, the data were classified into 10 clusters, and the optimal numbers of iterations for five different data quantities of CAN FD (100, 300, and 500 pieces of data) were identified in advance.

TABLE III: BIT FIELDS IN THE CAN FD FRAME AND THEIR LENGTHS

Field name	Length (bits)
SOF (Start of Frame)	1
ID (Identifier)	11
r1(Reserved Bit 1)	1
IDE (Identifier Extension Bit)	1
EDL (Extended Data Length)	1
r0(Reserved Bit 0)	1
BRS (Bit Rate Switch)	1
CRCdel (CRC Delimiter)	1
ACK (Acknowledge)	1
DEL (Delimiter)	1
EOF (End of Frame)	7
IFS (Interframe Spacing)	3
DLC (Data Length Code)	4
ESI (Error State Indicator)	1

*b) K-means data clustering—squared Euclidean distance algorithm*

Fig. 3 shows the sum of minimum distances between clusters associated with different numbers of iterations for 100 pieces of CAN FD data clustered using the squared Euclidean distance method. After 750 iterations, the distance between the clusters was short and stable; thus, the number of iterations was set as 750 in the experiment.

Fig. 4 shows 300 pieces of CAN FD data clustered using the squared Euclidean distance method. The distance

between clusters was short and stable after 1200 iterations; thus, the number of iteration was set to be 1200 in the experiment.

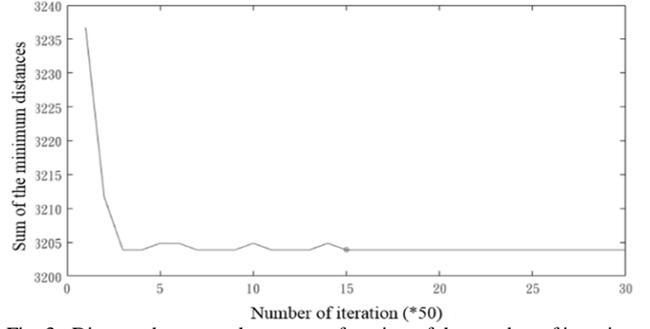


Fig. 3. Distance between clusters as a function of the number of iteration (100 pieces of CAN FD data).

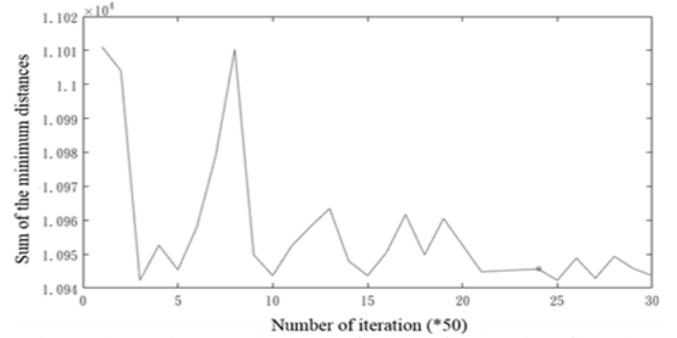


Fig. 4. Distance between clusters as a function of the number of iteration (300 pieces of CAN FD data).

Fig. 5 shows the 500 pieces of CAN FD data clustered using the squared Euclidean distance method. The distance between clusters was short and stable after 750 iterations; thus, the number of iteration was set to be 750 in the experiment.

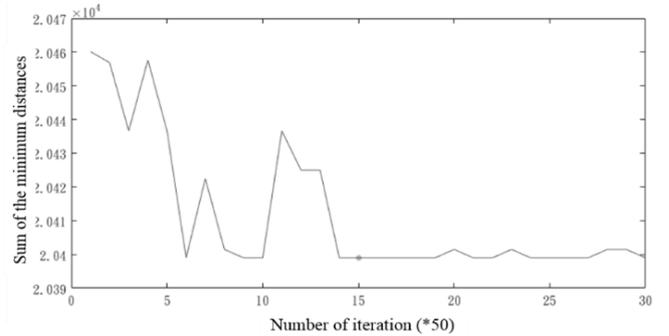


Fig. 5. Distance between clusters as a function of the number of iteration (500 pieces of CAN FD data).

Table IV lists the iteration numbers for associated CAN FD data quantities.

TABLE IV: ITERATION NUMBERS OF CAN FD

CAN FD data quantity	Optimal iteration number
100	750
300	1200
500	750

IV. SYSTEM TEST

In addition to the verification of the effect of K-means clustering on data loss, this section also presents the verification of CAN FD nodes, CAN nodes, and gateways.

Fig. 6 shows the system designed in this study.

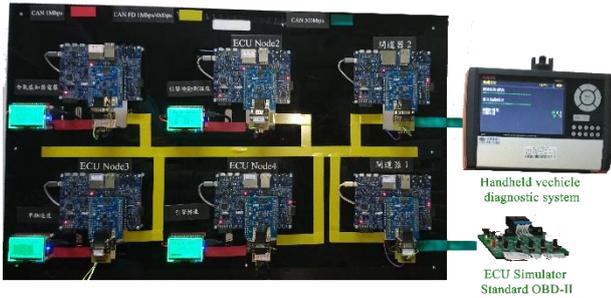


Fig. 6. Network system designed in this study.

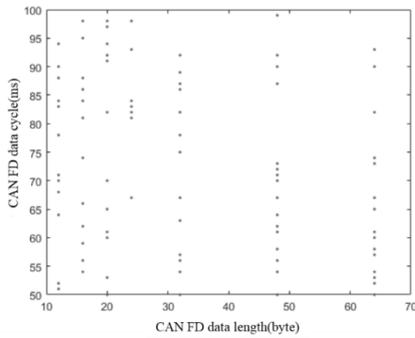


Fig. 7. Distribution of 100 pieces of CAN FD data.

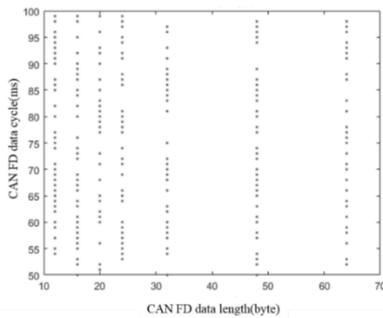


Fig. 8. Distribution of 300 pieces of CAN FD data.

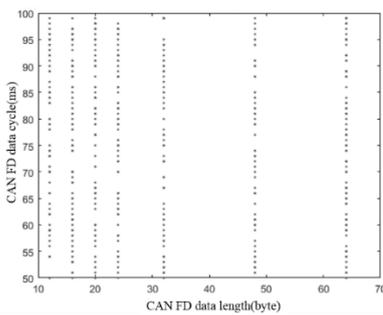


Fig. 9. Distribution of 500 pieces of CAN FD data

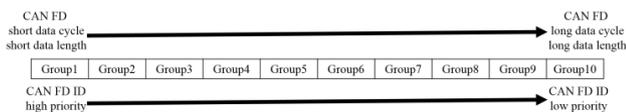


Fig. 10. Priority of CAN FD data.

To improve the stability of the CAN FD network, this study used K-means clustering to reduce the data loss rate of the CAN FD network, thereby improving the security of the communication network. The machine learning method, K-means clustering, was incorporated into the CAN FD network of the LPC54618 microcontroller to improve the transmission performance of the overall communication network. K-means was adopted to test five different quantities of CAN FD messages (100, 300, and 500 pieces

of data) by using two algorithms and two CAN FD network rates. The effect of data clustering on data loss rate in CAN FD network was observed by using a LPC54618 microcontroller. Fig. 7-9 show the distributions of the three quantities of CAN FD data before data clustering.

From the distributions of the five sets of CAN FD data, the lengths of the CAN FD data were 12, 16, 20, 24, 32, 48, and 64 bytes, all of which were evenly allocated to the cycles of 50 to 100 ms.

The K-means method was then used to perform data clustering by using two algorithms, namely the squared Euclidean distance and the sum of absolute differences. The five quantities of CAN FD data were clustered by the two algorithms, and each data quantity was divided into 10 clusters. Fig. 10 shows the priority of CAN FD data after clustering. The cluster exhibiting the shortest CAN FD data length and the shortest data cycle was designated as the first cluster and was assigned the highest priority ID, and the rule was applied to the remaining clusters to reduce the priority of each CAN FD data ID.

The following sections introduce the verification and testing of CAN FD data clustering by using two algorithms of K-means—the squared Euclidean distance and the sum of absolute differences.

1) *K-means data clustering—squared Euclidean distance method*

Fig. 11-Fig. 13 show the test results of different quantities of CAN FD data (100, 300, and 500 pieces of data) clustered by the K-means Squared Euclidean distance method.

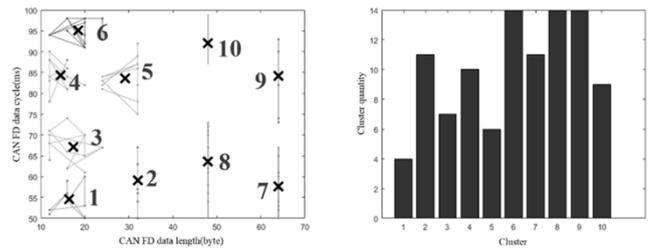


Fig. 11. Results of clustering 100 pieces of CAN FD data by using the squared Euclidean distance method.

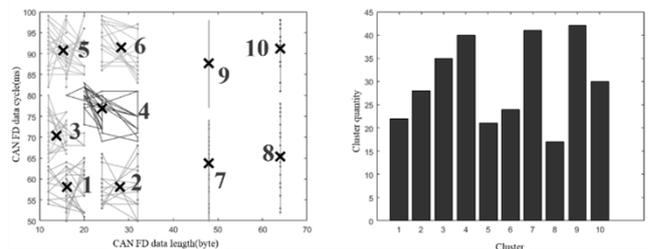


Fig. 12. Results of clustering 300 pieces of CAN FD data by using the squared Euclidean distance method.

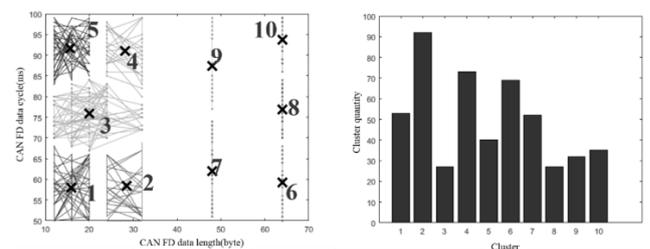


Fig. 13. Results of clustering 500 pieces of CAN FD data by using the squared Euclidean distance method.

After data clustering, these CAN FD data were imported

into the CAN FD network of the LPC54618 microcontroller. The data loss rate before and after data clustering was tested using two CAN FD data rates. The results indicated that due to the clock rate limit of LPC54618, different random data sets exhibited different improvement effects for data loss rate. Therefore, the data loss rates derived from the LPC54618 microcontroller were the average values of repeatedly testing on each datum for five times. Equation (2) was used for calculation, and the busload rate was calculated by equations (3)–(7) given a CAN FD arbitration phase rate = 1 Mbps, Table V and Table VI list the data loss rates of the data phase rates of 2 and 4 Mbps, respectively.

TABLE V: DATA LOSS RATES AFTER USING THE SQUARED EUCLIDEAN DISTANCE ALGORITHM FOR DATA CLUSTERING

Number of pieces of CAN FD data	100	300	500
busload rate	29.6%	79.79%	142.63%
Successful reception rate (before clustering)	62.19%	58.72%	40.52%
Successful reception rate (after clustering)	69.68%	61.35%	42.56%
Reduction in data loss rate	7.49%	2.63%	2.04%

TABLE VI: DATA LOSS RATES AFTER USING THE SQUARED EUCLIDEAN DISTANCE ALGORITHM FOR DATA CLUSTERING

Number of pieces of CAN FD data	100	300	500
busload rate	17.1%	46.62%	82.6%
Successful reception rate (before clustering)	64.37%	60.42%	59.49%
Successful reception rate (after clustering)	72.71%	66.29%	61.80%
Reduction in data loss rate	8.34%	5.87%	2.31%

In the case of CAN FD data phase rate = 2 Mbps, due to bandwidth overload, data loss rates associated with 500 pieces of data were relatively high. However, the data loss rates were still declined after data clustering. For 100 and 300 pieces of data, the reduction in data loss rates was evident. In the case of the CAN FD data phase rate = 4 Mbps, due to the elevation in bandwidth, the data loss rate was lower than that of the CAN FD data phase rate = 2 Mbps. After data clustering, the data loss rates for the five data quantities were substantially reduced.

## V. CONCLUSION

To meet the requirements of complex vehicle network in the future market, this study adopted the machine learning K-means method to establish the network ID configuration of CAN FD, and the configuration was imported into the CAN FD network of LPC54618 microcontroller. Additionally, different data priority was assigned to CAN FD data. According to the experiment results in the fourth chapter, in the case of 100–500 pieces of CAN FD data, given the CAN FD arbitration phase rate = 1 Mbps, for the data phase rates = 2 and 4 Mbps, the squared Euclidean distance algorithm reduced the data loss rates by 7.49% and 8.34%, respectively. The results indicated that the overall performance of CAN FD network could be improved by using the K-means squared Euclidean distance algorithm to assign high data priority of CAN FD IDs to the short length and short cycle of CAN FD data.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

Yung-Hoh Sheu conducted the research; Cheng-Yo Huang, Chen-Yu Yang and Yi-Hong Lin analyzed the data; Yung-Hoh Sheu and Cheng-Yo Huang wrote the paper. All authors had approved the final version.

## ACKNOWLEDGMENTS

We would like to express appreciation to the Ministry of Science and Technology for sponsoring this project (MOST 107-2221-E-150-008 - ).

## REFERENCES

- [1] M. N. Ramadan, M. A. Al-Khedher, and S. A. Al-Kheder, "Al-kheder intelligent anti-theft and tracking system for automobiles," *International Journal of Machine Learning and Computing*, vol. 2, no. 1, pp. 83-88, February 2012.
- [2] Y. Yoshida, H. Ohwada, F. Mizoguchi, and H. Iwasaki, "Classifying cognitive load and driving situation with machine learning," *International Journal of Machine Learning and Computing*, vol. 4, no. 3, pp. 210-215, 2014.
- [3] H. Hormozi, E. Hormozi, and H. R. Nohooji, "The classification of the applicable machine learning methods in robot manipulators," *International Journal of Machine Learning and Computing*, vol. 2, no. 5, pp. 560-563, 2012.
- [4] M. Q. Mao, Y. Y. Wang, and Y. Yue, "Multi-time scale forecast for schedulable capacity of EVs based on big data and machine learning," *IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 1425-1431, Cincinnati, OH, USA, Oct. 1-5, 2017.
- [5] L. L. Vy, L. P. Tung, and B. S. P. Lin, "Big data and machine learning driven handover management and forecasting," in *Proc. IEEE Conference on Standards for Communications and Networking (CSCN)*, 2017, pp. 214-219.
- [6] F. C. Sun, G. B. Huang, and Q. M. J. Wu, "Efficient and rapid machine learning algorithms for big data and dynamic varying systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2625-2626, Sept. 1, 2017.
- [7] C. Chen, K. L. Li, A. J. Ouyang *et al.*, "GPU-accelerated parallel hierarchical extreme learning machine on flink for big data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2740-2753, April 24, 2017.
- [8] J. Chin, V. Callaghan, and I. Lam, "Understanding and personalising smart city services using machine learning. The Internet-of-Things and Big Data," presented at International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, June 19-21, 2017.
- [9] M. Chen, Y. X. Hao, and K. Hwang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869-8879, April 2017.
- [10] Y.-H. Sheu, Z.-W. Wang, and Y.-J. Liou, "The K-means clustering methods applied in the message scheduling methods of FlexRay communication network based on the embedded FlexRay node," presented at International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, August 27-August 29, 2014.
- [11] R. D. Andrade, K. N. Hodel, J. F. Justo, A. M. Lagan á M. M. Santos, and Z. H. Gu, "Analytical and experimental performance evaluations of CAN-FD bus," *IEEE Access*, vol. 6, pp. 21287-21295, April 16, 2018.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



**Yung-Hoh Sheu** received the M.S. degree in biomedical engineering and the Ph.D. degree in the Department of Electrical Engineering from the National Cheng Kung University of Taiwan, in 1991 and 1996, respectively. He is currently a professor at the Institute for Computer Science and Information Engineering at the National Formosa University in Taiwan. Dr. Sheu operates an Embedded Software System Lab and received several awards of international invention new product expedition. His research interests include modeling, design, test, and verification of embedded systems with a focus on vehicle communication network, USB device design and interactive game design.



**Cheng-Yo Huang** received the B.S. degree in computer science and information engineering from National Formosa University, Yunlin, Taiwan, in 2016 and the M.S. degree in computer science and information engineering from National Formosa University, Yunlin, Taiwan, in 2018. His research interests include design, test, and verification of embedded systems with a focus on vehicle communication network design.



**Yi-Hong Lin** received the B.S. degree in computer science and information engineering from National Formosa University, Yunlin, Taiwan, in 2018. His research interests include design, test, and verification of embedded systems with a focus on vehicle communication network design.



**Chen-Yu Yang** received the B.S. degree in computer science and information engineering from National Formosa University, Yunlin, Taiwan, in 2018. His research interests include design, test, and verification of embedded systems with a focus on vehicle communication network design.