Received January 11, 2018, accepted February 7, 2018, date of publication February 23, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808913

Rapid-Response Framework for Defensive Driving Based on Internet of Vehicles Using Message-Oriented Middleware

PO-YU LAI[1], **CHYI-REN DOW**[1], **AND YU-YUN CHANG**[2]

[1]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[2]Industrial Ph.D. Program of Internet of Things, Feng Chia University, Taichung 40724, Taiwan

Corresponding author: Chyi-Ren Dow (crdow@mail.fcu.edu.tw)

This work was supported by the Ministry of Science and Technology, Republic of China, under Grant 105-2221-E-035-050-MY2.

ABSTRACT With the rapid development of automatic driving and advanced driver assistance systems, vehicle safety has improved greatly. These systems mainly use sensors installed on the vehicle and help drivers deal with human operation errors. Traffic accidents are inherently unpredictable, and it is difficult to prevent mistakes made by others. Therefore, the concept of defensive driving has attracted much interest. Defensive driving aims to increase drivers' self-awareness to prevent accidents. Future self-driving vehicles should integrate defensive driving to improve driver safety. This paper proposes a framework based on the risk evaluation value of defensive driving that rapidly transmits information about high-accident-likelihood zones to drivers or vehicles by using Internet of Vehicles technology. This should enable drivers or self-driving vehicles to predict risks and operate vehicles safely. To send alert messages in a timely manner, it is essential to overcome the challenge of processing real-time data during driving. We design five kinds of services in this rapid response framework, including raw data receiver, warning area decision, accident pattern recognition, message generator, and user profile to analyze driver information using distributed system architecture. Message-oriented middleware is used for communication between services. This framework identifies high-accident-likelihood zones by using density-based spatial clustering of applications with noise, simplifying the process of association calculation. After the calculation, this framework uses the weighted severity index to weight and compare risk severities. According to our experimental results, the service-oriented middleware design increases the speed and stability of information transmission.

INDEX TERMS Apriori, DBSCAN, defensive driving, Internet of Vehicles, message-oriented middleware.

I. INTRODUCTION

Defensive driving has always been considered safe driving. Unmanned driving will become safer if self-driving technology can integrate the concept of defensive driving [1]. However, defensive driving for avoiding potential risks depends on drivers' experience [2]. Driver experience can be improved using machine learning techniques. Shimosaka *et al.* [3] proposed a constructed a risk prediction model and learning framework for drivers. The present paper proposes a method for sending messages about potential accident risks to drivers by using Internet of Vehicles (IoV) technology. In the future, alert messages can be sent to self-driving vehicles; once a vehicle receives a message, it will be able to adjust its driving action and therefore achieve the goal of avoiding accidents.

Thus far, many studies have investigated defensive driving. However, these studies have rarely focused on how to evaluate a potential risk or how to alert a vehicle about the risk. Risks are evaluated based on massive high-dimensional data computing; therefore, constructing a predicting model is very difficult. One method of increasing driving safety is to improve driving assistant systems. Another view holds that humans are the main cause of accidents. In fact, self-driving technology cannot completely avoid accidents. In a 2015 New York Times report on Google Cars' accident statistics [4], Donald Norman noted "They have to learn to be aggressive in the right amount, and the right amount depends on the culture." This shows that no matter how many sensors have been used, without a favorable responsive system to execute defensive actions, accidents will still occur. Another challenge is how to send alert messages to drivers in a timely manner. When a vehicle is moving and passing through a high-risk zone, an alert message should be sent to the driver early. Therefore, we consider that the IoV is a great solution to achieve the goal of evaluating risks and alerting vehicles about them.

With the advent of Internet of Things (IoT) technologies, vehicle networking technology has gradually evolved from vehicular ad hoc networks (VANETs) to the IoV. The two main research domains in the IoV are vehicle networking and vehicle intelligence. Yang *et al.* [5] stated that vehicle intelligence integrates drivers and vehicles. Vehicles become more intelligent through network technology; namely, processes such as deep learning, cognitive computing, swarm computing, and uncertainty artificial intelligence. The IoV is a network that can provide services over a large area or even a whole country [6].

In Taiwan, the transportation department and city governments provide smartphone warning apps; however, these apps are not capable of receiving real-time transmissions. Drivers are sometimes unable to receive alert messages in time to respond effectively. Such apps receive alert messages through the REST API, a web service interface based on the HTTP network. The app periodically contacts the server to request information (also known as server polling). The polling mechanism can cause server overload because of numerous invalid queries. This delays crucial alert messages, meaning that they are no longer be helpful to drivers when they are received [7]. Some apps use server push techniques to overcome the problems of polling mechanisms. Push notifications are sent by a third-party push notification service. However, this mechanism is not continuously stable. Another solution increases stability and reduces alert spots: the server only sends alert messages when drivers approach some high-accident-risk junctions or during specific periods. Although this approach solves the message delay problem, it may also severely limit the system capability.

To solve the aforementioned problems, we propose a rapid-response framework based on message-oriented middleware (MOM) that combines the advanced message queuing protocol (AMQP) and message queue telemetry transport (MQTT). MQTT and AMQP have proven to have better performance in IoT [8], [9]. We focus on adaptive calculation, system performance, and communication architecture. An association calculation algorithm is employed to generate alert messages after the alert period for each zone is calculated. When the system receives the calculated data, it generates a suitable alert message according to driver characteristics such as sex, age, position, date, and time.

This paper proposes a simple and efficient method to calculate accident risk. First, geographic spots are clustered using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to identify hotspots, which also distinguish some of the characteristics of the environment. Second, we use the Apriori algorithm to calculate the most likely accident-causing factors at the hotspots. We also calculate its weights using personal conditions and the combination of factors under the severity of the accident (WSI). This calculation procedure rapidly performs calculations and transmits data. Future vehicles will be able to receive risk evaluations using this system to enable appropriate defensive driving.

The remainder of this paper is organized as follows. Section II discusses studies investigating traffic accidents, IoV, and MOM. Section III presents the design of the service framework. In Section IV, we illustrate the validation and implementation of our system. In Section V, we analyze accident data for Taichung City and prove accidents are usually connected to factors in the surrounding environment. The experimental results are presented in Section VI. Finally, the conclusions of this paper are drawn in Section VII.

II. RELATE WORK

New technologies for detecting and preventing traffic accidents have been proposed in the fields of traffic engineering, computer science, and engineering. Traffic engineering focuses on road improvements, whereas computer science and engineering focus on vehicles and drivers to decrease the probability of traffic accidents.

Traffic accidents are intrinsically unpredictable. Numerous studies investigating traffic accidents have focused on conveying accident information to relevant parties in the shortest time possible and on assisting drivers and injured people as quickly as possible [10], [11]. Several studies have noted that driver behavior is the main factor causing traffic accidents, and therefore, they have used a controller area network (CAN bus) and other sensors to collect vehicle data. Kaplan *et al.* [12] identified drowsiness and driver distraction as the main reasons for traffic accidents. They proposed a method to monitor a driver's physical state by using a smartphone and a wearable device as well as to notify other drivers of potential risks through vehicle-to-vehicle (V2V) technology. Jadhav and Wagdarikar [13] used an embedded system to receive data from sensors such as eye-blink sensors, CAN buses, webcams, and GPS receivers. After data analysis, GSM modules transmit the status to a server.

Many researchers have presented analyses of the causal factors of accidents. Solaiman *et al.* [14] designed a website that enables users to input accident information. This website reports accident causes and trend data based on analyses of previous accident information. It also provides a visualization service using a geographical information system. Christian and Quintero [15] presented an intelligent driving assistant system based on artificial neural networks. Their method aimed to provide reliable driving recommendations by using accident risk maps analysis and intelligent driving diagnosis. Zhan *et al.* [2] proposed a non-conservatively defensive strategy (NCDS) for various scenarios faced in urban autonomous driving.

Studies have long focused on establishing reliable network connections in mobile vehicles. Studies have investigated the IoV, especially in the fields of network connection technology, autonomous vehicles, and vehicular clouds. TrendForce [16] forecasted that by 2020, 75% of cars on the road will be able to connect to a network and that autonomous cars will number more than 1 million. Therefore, receiving or sending messages through the Internet will be common in the future. Trends in IoT research reveal that VANETs are also considered a rich mobile sensor platform. Ensuring communication connections is critical in personal vehicles because they move between highly and lowly connected regions rapidly. Many studies have used wireless access in vehicular environments and dedicated short-range communication (DSRC) to solve communication problems. Meng *et al.* [17] proposed an efficiency evaluation system that uses end-to-end delay time to detect packet loss rate. In addition to V2V communications infrastructure, long-term evolution (LTE)-based communication methods have been discussed. Imadali *et al.* [18] noted that vehicle manufacturers install LTE communication devices in on-board units (OBUs) because of the slow progress of IEEE 802.11p. In the future, IEEE 802.11p should use IPv6 to integrate with LTE networks. Grigoryev *et al.* [19] compared the costs of networks using UHF radio waves, DSRC, Wi-Fi, and LTE and found that use of LTE incurred the lowest cost.

Vehicles currently contain numerous mobile platform sensors. These sensors generate a considerable volume of data when the vehicle is operated, thereby serving as a source of big data. Instant big-data computation is another prominent research topic. Aloqaily *et al.* [6] proposed the integration of vehicular cloud computing with mobile cloud computing and vehicular communications and compared the weaknesses of some common cloud services for vehicular cloud computing. Meneguette [20] proposed a new protocol to facilitate resource sharing via a mobile cloud in vehicular networks and realized high resource availability of approximately 95%. Meng *et al.* [21] investigated the resource allocation problem in vehicular cloud computing, employed a semi-Markov decision process to calculate a maximum average rate of return, and obtained an optimal solution using an iteration algorithm. Gerla *et al.* [22] indicated that the reason for the success of Google's self-driving car project was the application of cloud computing in vehicles. Their study investigated intelligent vehicle grids for autonomous, Internet-connected vehicles and the vehicular cloud.

Scholars have researched MOM-related technology for decades. For example, Parlanti *et al.* [23] proposed a service and application integration framework using this technology. Their system obtains dynamic information through message queuing (MQ) to monitor the activity of ships at sea. Morais and Elias [24] proposed MOM architecture for mobile devices and evaluated the influence of different MOM modes on memory and data throughput. Chongnan *et al.* [25] studied the application of MOM to telemetry tracking and command systems. They analyzed the components in the architecture and proposed a virtualization strategy using multi machine. Labéjof *et al.* [26] proposed the R-MOM framework, an adaptive asynchronous middleware system for the sensor

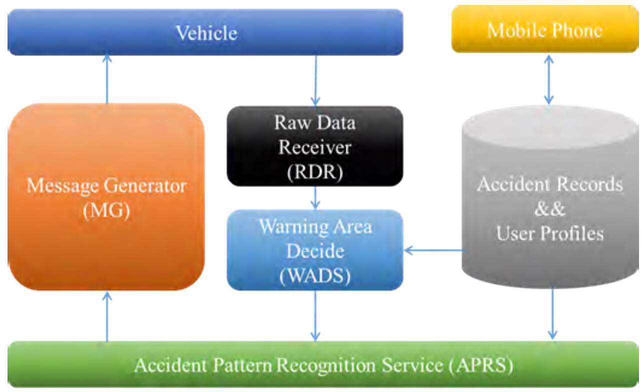

FIGURE 1. Service framework.

observation service, and implemented the interoperable framework using Java Message Service (JMS), the Advanced Message Queuing Protocol (AMQP), and the data distribution service (DDS); moreover, they experimentally evaluated the performances of JMS, the AMQP, the user datagram protocol, and the DDS.

III. FRAMEWORK DESIGN

The core services of the framework proposed herein, shown in Fig. 1, consist of the raw data receiver (RDR), warning area decision service (WADS), accident pattern recognition service (APRS), message generator (MG), and user profile service (UPS). This section discusses the design of these services. Some phenomena are observed from accident data: 1) high-accident-likelihood periods overlap with peak-traffic periods; 2) traffic accident hotspots change over time; 3) transportation use differs by age group among the victims of accidents; and 4) the distribution and time occurrence of accident hotspots differ between working days and holidays.

As mentioned previously, we propose a framework based on the spatial distribution of traffic accidents. An alert service requires a reaction when an alert has the highest priority, even when the framework is under heavy loading. Therefore, fast computing and transmission are the most critical elements of our framework design. The framework can be divided into five services.

1) The RDR is responsible for receiving information from the vehicle—including speed, latitude and longitude, and identity—and forwarding this to the AMQP. The RDR is also responsible for preliminary data filtering. For example, the receiver stops forwarding data when the vehicle's speed is zero.

2) The WADS determines whether a vehicle has entered the warning area by using the results obtained through DBSCAN.

3) The APRS infers the risk of a traffic violation and alerts drivers according to their time period and location. This service uses the association calculation algorithm to classify historical accident records by applying multiple parameters including a driver's personal information, date and time, and traffic flow as inputs.

4) The MG receives information from the APRS and sends alert messages to specific vehicles.

5) The UPS manages the personal information of users, including driver age, sex, and driving-behavior statistics.

We design the communication structure using two MOM protocols. Vehicles use MQTT to communicate with the RDR and MG, whereas the AMQP is used for communication between services. User registration with an MQTT broker is implemented through a RESTful API by using HTTP.

Our framework based on IoV, the following three problems persist. And these problems are also the goals what we mainly improve in our system.

1) **The system must react quickly:** A typical vehicle travels very fast, so the system must deliver messages to the vehicle before it enters and leaves an accidental hotspot.
2) **The system must be capable of load balance:** There are thousands of devices in the IoV environment, so the system must have the capacity of expanding to handle user needs.
3) **The message delivery must be adaptable:** Each driver has an individual risk of accidents that depends on that driver's personal conditions.

Because IoV communication is not stable when vehicles move, the greatest challenge is how traveling vehicles can communicate with the system. Using a small quantity of bandwidth for a traveling vehicle can reduce the impact of unstable communication. Aside from VANET, MQTT uses the least bandwidth of any currently available protocol of wide-area communication [27]. However, the lightweight design of MQTT also results in management and safety problems. AMQP is a full-featured communication protocol with functions such as direct exchange and load balancing [7]. However, the transmission of AMQP is not as efficient as that of MQTT [8], [9]. Therefore, we integrate MQTT and AMQP into a communications system, which simultaneously provides a rapid response function and a load balancing function. We divide the communication system into external communication and internal communication. External communication uses MQTT for communication between the vehicles and the system; MQTT is more efficient for this task. Internal communication uses AMQP with load balancing for communication between two services, as shown in Fig. 2.

The entire service is based on high cohesion and low coupling. RDR is responsible for receiving; WADS is responsible for determining whether the vehicle is located in the hotspot; APRS is responsible for risk assessment; and MG is responsible for generating messages. All these services operate independently. With an IoV/IoT design, we assume that a single computer would be paralyzed by a large number of users, so our service is designed to run on multiple computers. For example, when a single APRS server is paralyzed, we can add APRS functionality arbitrarily by distributing messages equally to other APRS services through the direct exchange function of AMQP, so the system load can be balanced

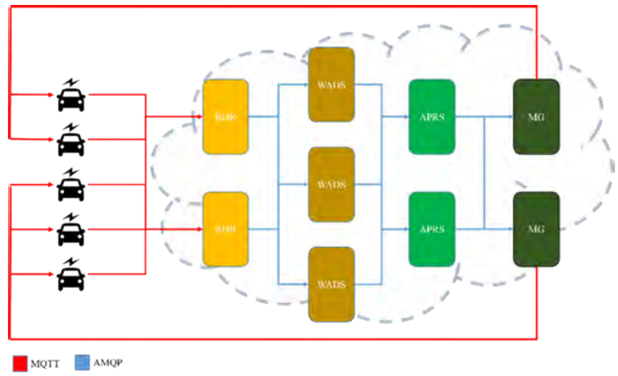

FIGURE 2. Communication architecture.

simply. This is also the main reason why we use a mixed design of MQTT and AMQP.

A. RDR

The RDR is a service between clients (vehicles) and the server; it also converts between MQTT and the AMQP. As an MQTT subscriber, the RDR should be able to process large data streams without difficulty. The vehicle control unit transmits the position and speed of the vehicle to the RDR. After receiving the data, the RDR only assesses the speed. When the speed data is zero, the vehicle is stationary. The RDR then omits this record rather than forwarding it. When the vehicle speed is greater than zero, the RDR becomes a message producer and transmits messages to the AMQP service. Thus, the WADS becomes a message consumer.

B. WADS

The WADS is used to calculate the speed of traffic flow within a warning range. Traffic collisions involve constant and variable factors. Variable factors include weather conditions, time, lighting, and driver behavior. Constant factors include fixed facilities such as lanes, markings, and traffic signs. These factors are related to the occurrence of accidents. The WADS analyzes a driver's current location and time and determines requirements for further data processing.

We find that traffic collision factors differ by zone and period. The date and time of accidents can be used as a filter. The date is categorized into weekdays and holidays, and the time is divided into 12 periods of 2 h each. These datasets can be evaluated using DBSCAN to determine the intensities of the accident points. DBSCAN evaluates the number of points adjacent to an accident point to determine whether there are sufficient adjacent points to form a cluster. If there are sufficient adjacent points, the accident point becomes a core point. After DBSCAN has found a core point, the remaining adjacent points are evaluated incrementally to identify the next core point. If another adjacent point becomes a core point, the cluster boundary is extended. DBSCAN continues this process until no more core points can be found, as shown in Fig. 3.

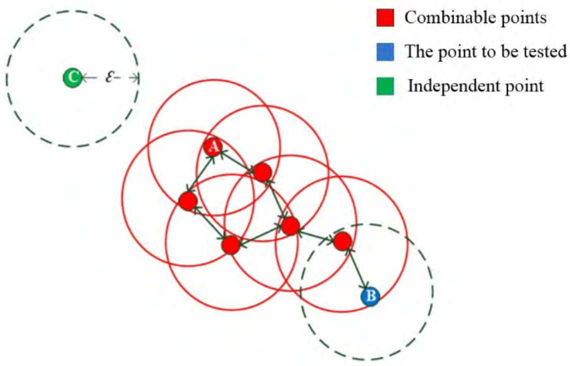

FIGURE 3. DBSCAN schematic (Neps = 3).

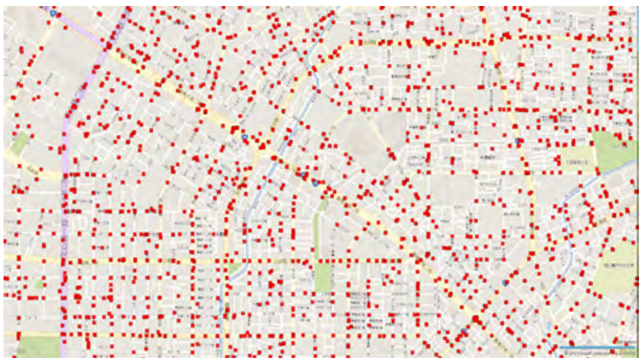

FIGURE 4. Original distribution of traffic accidents.

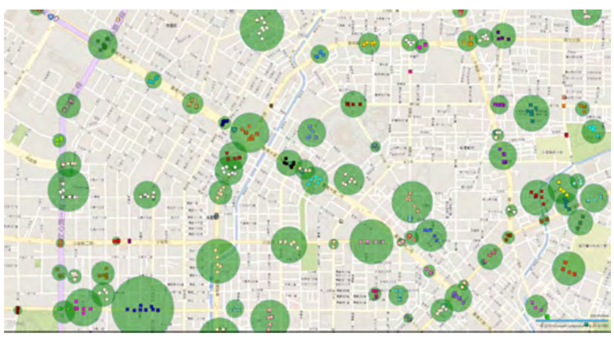

FIGURE 5. Warning range distribution map after DBSCAN implementation.

DBSCAN has two important parameters:

- ***Eps*** - epsilon neighborhood: Each point uses *Eps* as radius to search the near point within this range.
- ***Neps*** - minimum number of points required within *Eps*: If a point can search adjacent points in the range of *Eps*, and the amount of adjacent points is bigger than *Neps*. This point and adjacent points will become a cluster.

Figure. 4 shows the DBSCAN operation for the locations of traffic accidents that occur between 8:00 A.M. and 10:00 A.M. in Taichung City. We implement DBSCAN with *Eps*=25 m and *Neps*=10, and the results are shown in Fig. 5.

The main road of Taichung City has four lanes and a width of approximately 25 m; thus, *Eps*=25 m and *Neps*=10. This means that a cluster is formed if there were ten accidents, and this is considered an accident hotspot.

TABLE 1. Analysis categories.

Category	Description
Accident types and patterns	The type of collision object: vehicles/ motorcycles to pedestrians, vehicles to vehicles/ motorcycles, vehicle/ motorcycle self-crush and railway level-crossing etc..
Transportation type	Transportation in accident, such as buses, large trucks, minibuses, minivans, locomotives, people etc..
Age	The age of driver.
Transportation action	The action of transportation in accident: starting, backing a car, turning.
Judgment of accident cause	A preliminary judgment report of accident cause. The police investigate the cause of the accident within 30 days. There are some categories: driver, lights, loading, mechanical, pedestrians (or passenger) and traffic facilities.
Accident position	The road type of accident position, such as intersection, road section, interchange.

When a vehicle enters the warning range, the service assesses its data and adds the vehicle to the alert message publishing table.

C. APRS

The APRS is the control center for warning messages. This service handles the majority of computing tasks. Thus, we prefilter the data in the RDR and WADS to reduce the data processing load in the APRS.

Vains [28] studied accident data analysis using Bayesian networks and conditional probability functions and found that accidents can be considered a multidimensional data set. This characteristic of data is difficult to train using supervised learning algorithms such as a neural network. We have previously used Random Forest (RF) and Back Propagation Network (BPN) to cluster data. However, data is insufficient for numerous accident hotspots for making accurate predictions, resulting in overfitting.

The accident record is a table containing information about location conditions. The record comprises accident data from Taichung City during 2013–2015 and is divided into a main table and a subtable. The main table includes the date, time, weather conditions, road conditions, traffic signs, and location of an accident on the road. The subtable includes information about the driver and transportation, such as vehicle type, age, and driving qualifications. Because these categories have several subcategories, some personal information should be deleted, such as the driver's occupation and travel purpose. Moreover, we must filter some static row data, such as road type, markings, traffic signs, and other fixed facilities. Finally, we select categories for analysis (see Table 1).

These six categories include dozens of subcategories; judgment of accident cause has 67 subcategories. These categories are too complicated to analyze. We use the hierarchy method to filter data and construct associations between categories. To construct a directed graph of categories, we filter these data by using three conditions: administrative region, date, and time. The filter sequence of conditions is: driver age → transportation → transportation action → accident type and pattern → accident position → judgment of accident cause. The age category is divided into nine segments, each with an interval of 10 years.

The adaptive messages are required for us, and it must be evaluated in terms of time, place, individual information and the severity of the incident. For this purpose, we perform the algorithm with these three steps:

1) Data association: The data-associated degrees of these six categories are computed using the Apriori algorithm. For any combination of categories, the higher the data-association degree, the higher is the probability that an accident will occur. After the data-associated degrees are known, the critical messages of the accident hazard can be generated.

2) Personal characteristic: In the weighted calculation of personal characteristics, the candidate item generated by the Apriori computation is compared with individual data, such as age, gender, and occupation. Each itemset in the historical data is compared with personal data. If they match, the value of the support is added by a priority value P and the summation thus obtained is called a weight.

3) Accident severity: We calculate the WSIs of each data combination, and then multiply the supporting degrees with WSIs. The WSI represents the index of degree of accident severity, which is generated by the numbers of deaths, injuries, and property losses caused by the occurring accident. A higher WSI implies a higher probability of casualty.

After these three calculations, the particular data combination with the highest weight value has been chosen as the model of the sending message.

We must find the association between six factors. In specific accident hotspots, the frequency of accident factors is different. To use the highest-frequency factor as the main factor is not correct because accidents are caused by not only one factor. For example, one intersection has the highest accident frequency with driver age of 10 years and transportation type as car. However, this set occurs rarely or never; thus, we calculate the association between the six categories together. If a situation like our example actually exists, the confidence degree will be low owing to its low frequency. We use the Apriori algorithm to evaluate the associations among the six factors. The Apriori algorithm has been used in numerous data mining applications. It also yields good results for time-series data. Some researchers have analyzed alert correlations using the Apriori algorithm [29], [30]. A novel aspect of our procedure is that we set the minimum support to 2. When the dataset has two identical records, this procedure continues. In general, setting the minimum support inappropriately may cause excessive calculation of candidate itemsets and may increase the system load. As per the DBSCAN procedure, the total number of hotspots is approximately 10–1000, and therefore, the system load is relatively light.

Because the Apriori algorithm is an iterative process, Fig. 6 shows an iterative procedure. L indicates the itemsets; $L(x, y)$ is the itemset of the union of x and y categories, and S represents the support value. When S is larger than the minimum support, this itemset is a Large-Itemset. For example, if $L(a, b)$ is larger than the minimum support, we can continue to test $L(ab, c)$ and $L(ab, d)$. The Apriori algorithm is crucial for association data mining, but has poor efficiency. The dataset must be searched once for each calculation. Our accident database has more than 90 fields, and it would be time consuming to execute extensive calculations for all fields. Thus, we only calculate six categories in the Apriori algorithm to reduce the computation time. We search the database for the accident records from the hotspot area where the vehicle is currently located. The record contains only the ID number (Table 1) for generation of the message template.

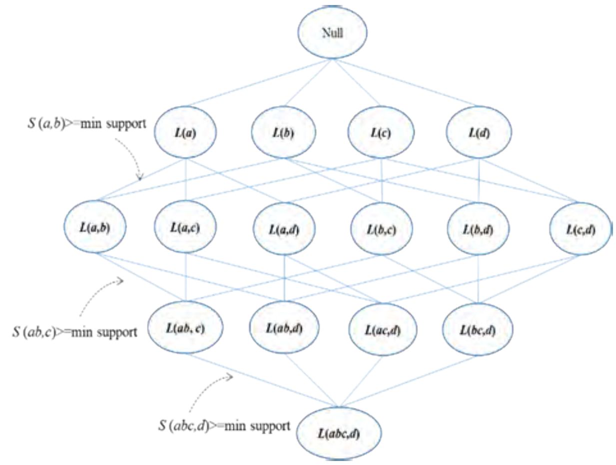

FIGURE 6. **The iterative process of Apriori algorithm.**

The search results are denoted as Dataset D, as illustrated in Fig. 7.

- Dataset: A record set that is two-dimensional, defined as D.
- Items: A set of all items, defined as I.
- Transaction: A record in the dataset, defined as T, $T \in D$.
- Itemset: A set of items that appears at the same time, defined as the *K-itemset*, where K represents the number of items.
- Support: It is defined as$supp(X) = occur(X)/count(D) = P(X)$. $P(A \cap B)$ represents the probability of A and B appearing simultaneously.
- Confidence: It is defined as $conf(X \rightarrow Y) = supp(X \cup Y)/(supp(X) = P(Y|X))$. For example, $P(B|A)$ represents the probability of event B occurring when the precondition that event A occurring is existing $P(AB)/P(A)$.
- Candidate Itemset: An itemset obtained by merging down, defined as $C_{[k]}$.
- A large *K-itemset*: If event A contains k elements, it is called a *K-itemset* event. If event A satisfies the minimum supporting threshold, it is called a large *K-itemset*. That is, the *K-itemset* is represented as $L_{[k]}$ if its support is greater than or equal to a particular Minimum Support.

The Apriori algorithm uses an iterative method with layer-by-layer search; that is, the K-1 itemset is used to search for

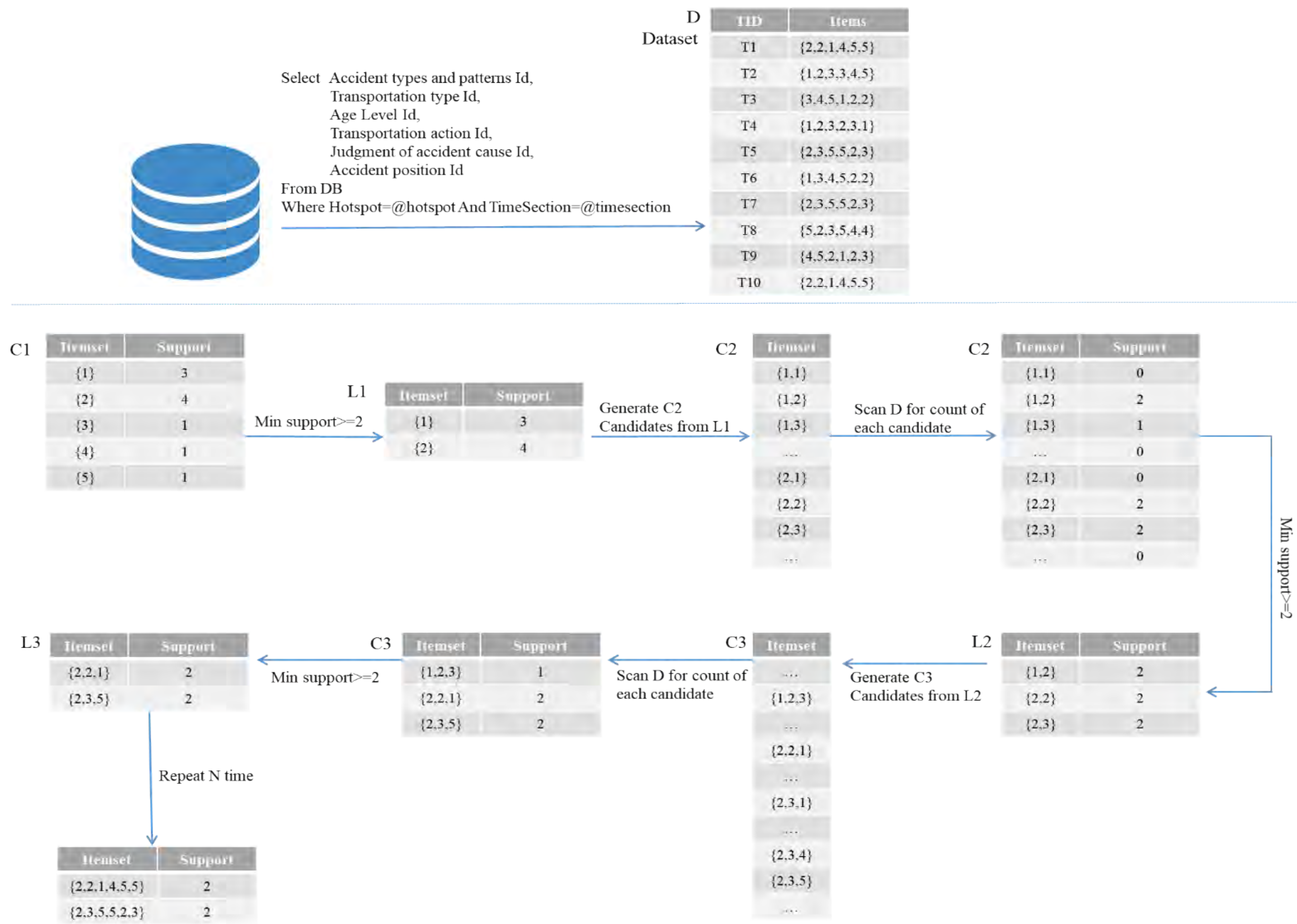

FIGURE 7. Apriori procedure diagram.

the K itemset. After searching the first large itemset, we mark it as L1. L1 is used to search the second large itemset, which is marked as L2. After that, L2 is used to search L3, and so on to Lk. The procedure is repeated until it cannot find any available itemset. The procedure searches a large itemset by iterating a loop of two steps: first, it generates candidate itemset C1, C2, . . ., Ck (which can become a portfolio of a large itemset). Then, it calculates the support of candidate itemset Ck to determine whether Lk is a large itemset. The searching strategy of the Apriori algorithm starts with a few items and gradually searches itemsets of multiple items.

The Apriori calculation chooses the maximum support itemset. There are two reasons why our calculation does not use the maximum support itemset. First, the accident severity should be considered. Thus, the candidate itemset amount is multiplied by the weighted severity index (WSI). There are a few serious accidents, and these accidents usually cause death or serious injury. We thought that these serious accidents should attract more attention than others. Second, when any factor of user profile has the same item in the candidate set, this candidate set is added by a priority value P which called a weight.

Accidents include many uncertainties, and the purpose of our service is not to predict the probability of a driver suffering an accident. Instead, it is to identify the factors causing an accident. Therefore, we provide an alert service that considers the risk of accidents.

At present, the most widely used weighted severity calculation is Equivalent Property Damage Only (EPDO) [31]; however, because the dataset we had is lack of some parameter, we are unable to use this method. Therefore, we choose the WSI for our weighting calculations. The WSI is modified from EPDO, and the area feature should be considered in EPDO. Therefore, we choose the WSI to calculate different accident types using the following equation [32]:

$$WSI = (12F + 3I + P)/(F + I + P) \tag{1}$$

F: number of accidents resulting in deaths
I: number of accidents resulting in injury
P: amount of property lost owing to accident

After the WSI is obtained, using amount of itemsets × WSI, the item with the highest value is used as the template

Algorithm 1 Apriori

```
Procedure: Apriori
Input: K;
Output: R; /*itemset association collection*/
L = Layer of itemsets;
S_min = min support;
For each L_K in L from D
        if K+1 < count of L then
                C_{K∪K+1} = L_K ∪ L_{K+1};
                if (C_{K∪K+1} ≥ S_min) then
                        R add N_{K∪K+1};
                        Apriori (K++);
                end
        end
end for
Return R;
```

TABLE 2. Message format.

VehicleId	AccidentTypeId	CausesId	PositionId	VehicleTypeId	ActiveId
Vid09813	2	5	4	1	3

for the alert message.

$$\forall x = \{A_1, A_2, A_3 \ldots A_n\}$$
$$T = \max(\forall x \times WSI) \tag{2}$$

A: amount of itemsets

T: message template

We evaluate the accident types and patterns in each administration zone. After the distance and weight have been calculated, the candidate records are multiplied by the risk evaluation of the accident type and pattern. The candidate with the highest weight becomes the template for generating a warning message. The warning message is transmitted to the MG in the format shown in Table 2.

D. MG

Usually, more than one person is involved in an accident. In an accident record, the first record is of the person who bears the greatest responsibility. Therefore, we use the first record as the sample to create a message. The template record includes the following columns, from which the message is formed:

1) Accident type: Transportation types of the persons involved. There are four type combinations, namely vehicle/motorcycle to pedestrian, vehicle to vehicle/motorcycle, vehicle/motorcycle self-crash, and railway level crossing. Each type has an independent subtype. For example, the subtypes of vehicle to vehicle/motorcycle include rear-end collision, side collision, and rubbing collision. The subtypes of vehicle/motorcycle to pedestrian include walking on the road and playing on the road.

2) Accident cause: The main cause of collision, such as driver behavior, vehicle mechanical failure, or signal failure.

TABLE 3. Raw vehicle data.

VehicleId	Latitude	Longitude	AvgSpeed
Vid09813	24.427658	120.980732	35

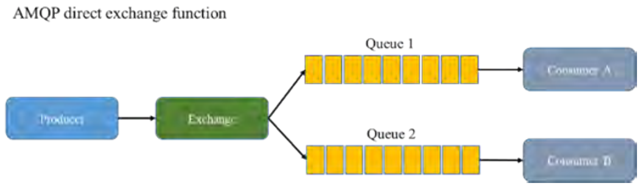

FIGURE 8. Work queue and routing modes.

3) Accident position: The road type of the accident location, such as crossroad, section, or dedicated lane.

4) Transportation type: The means of transportation, such as motorcycle, private small passenger vehicle, bus, or truck.

5) Transportation operation: The vehicle action, such as driving straight, making a right turn, and making a U-turn.

We use these five columns to create messages such as "At the crossing (accident position) ahead, motorcycle (transportation type) side impacts (accident type and subtype) often occur. Please pay attention to left-turn (transportation operation) red-light-running (accident cause) violations."

After the warning message is created, the MG publishes it to a vehicle according to its ID number.

E. MIDDLEWARE DESIGN

After a vehicle is activated, it must subscribe to a topic provided by the MG. The topic is defined as Alarm/{VehicleId}, where VehicleId is the unique identification number of each vehicle. Thus, each vehicle receives a unique topic depending on its VehicleId. The RDR subscribes to a topic–service/receiver. Each vehicle publishes messages to the RDR every 5s. The data format is shown in Table 3.

We use the AMQP for communication between the RDR, WADS, and APRS. The AMQP chooses different modes, such as work queue, routing, publish/subscribe, and topic. When a service bottleneck occurs, we use the work queue method to share traffic by adding servers. Conversely, if the framework needs to process specific information, the AMQP can achieve this through the routing key. A message with a specific routing key is passed to a specific service and process. Considering the scalability and load balance, we choose the AMQP as the server protocol [26]. Fig. 8 displays the work queue and routing modes.

F. UPS

We develop a user profile service using a Web API method. We consider that a vehicle may have multiple users, and each user has different personal conditions. We design a simple method that can link the current users and vehicles rapidly. Whenever a different driver takes control of the vehicle, the driver must first log into the APP, and then scan a unique

FIGURE 9. The mapping of driver and vehicle.

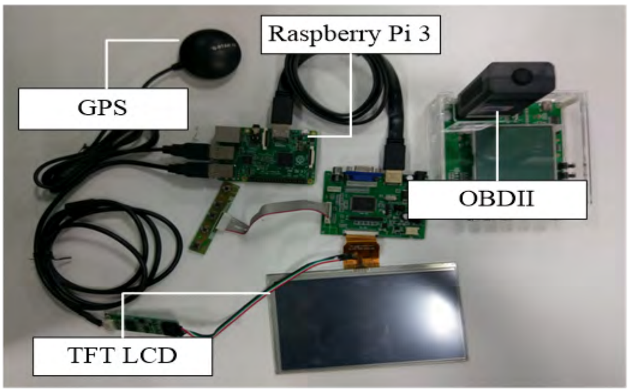

FIGURE 10. Implementation equipment.

FIGURE 11. High-accident-likelihood location.

FIGURE 12. No-left-turn sign.

QR code, which contains the Vehicle ID on the vehicle OBU, before driving. As shown in Fig. 9, once the driver has scanned the QR code, the APP sends the User ID and Vehicle ID to the server; together, the server and the APP provide a personal alert service.

IV. FRAMEWORK IMPLEMENTATION AND PROTOTYPE

We developed a driver alert service that uses Windows Server 2013 as the application server and Microsoft SQL 2014 Express as the database. For the middleware, Mosquitto is employed as the MQTT broker and RabbitMQ is used as the AMQP server. This framework is implemented using the C# language. We also use Raspberry Pi 3 as our OBU with Windows 10 IoT. The user client was developed using the Universal Windows Platform. The GPS system employs G-STAR IV (GlobalSat WorldCom Corp.). Vehicle speed is determined using OBDII with a Bluetooth interface device to retrieve vehicle CAN bus data (Fig. 10).

The intersection of Wuquan West Road and Huanzhong Road has the highest accident rate in Taichung City [33]. Wuquan West Road is the main conduit between urban and industrial areas and has plane and elevated sections. Historical records show that from 7:00 A.M. to 9:00 A.M. and 5:00 P.M. to 7:00 P.M., the majority of accidents are side impacts between right-turning vehicles and motorcycles caused by vehicle drivers not obeying the traffic rules. Figs. 11 and 12 illustrate the right turn. At other times of the day, the majority of collisions are side impacts caused by left-turning vehicles running a red light, giving straight-driving vehicles under the elevated road insufficient time to slow down to avoid a collision. Fig. 11 shows that a no-left-turn sign is installed above the road, and it is applicable from 7:00 A.M. to 9:00 A.M. and 5:00 P.M. to 7:00 P.M. This indicates that the traffic flow at this intersection reaches a peak during these two periods. When the traffic flow is high, side collisions usually occur because drivers disobey traffic signs. However, historical records show that side collisions are also frequent during nonpeak hours. At rush hour, the most common type of collision is vehicles leaving the interchange side and colliding with a motorcycle approaching from the right. However, traffic signs do not alert drivers at times outside these two periods.

We consider this intersection as an example. When a vehicle is moving, the screen shows its real-time location. The OBU connects to the server through the 4G LTE network. When the OBU executes a background MQTT service, it transmits the coordinates and speed to the MQTT broker. When the service determines the location of a driver who requires notification, the OBU receives a warning message on the subscribed topic.

In Fig. 13, the driver is leaving Provincial Highway 74 and entering Wuquan West Road. When the vehicle arrives at the

FIGURE 13. Alert message.

intersection of Wuquan West Road and Huanzhong Road, the driver receives an alert message regarding the intersection ahead; because accidents at this intersection are usually side collisions, the driver must pay attention to vehicles on the left-hand side.

Furthermore, we also use a text-to-speech (TTS) service to allow the driver to focus on driving and not remove their eyes from the road. If the user enables the voice function, the text message is sent to the service, transformed to speech by the TTS service, and then played to the user.

V. DATA ANALYSIS

We analyzed accident data for Taichung City, the third-largest city in Taiwan, collected from September 2013 to August 2015. The difference between urban and rural areas was substantial. Taichung City was formerly divided into two administrative regions, Taichung City and Taichung County, but they were merged into Taichung City in 2010. The terrain of Taichung City is extremely diverse, including coast, plains, and mountainous areas more than 3000 m in altitude. The urban area of Taichung City accounts for 7.3% of the total area of Taichung, 41.2% of the population, and 53.4% of traffic accidents.

Accidents are unforeseen events or circumstances that are random and lack patterns. Therefore, we cannot predict when, where, or who will experience a traffic accident. However, accidents are usually connected to factors in the surrounding environment, such as road design and traffic flow. By classifying the times and severities of specific conditions, we can obtain a reference and valuable information.

Over the course of a day, the accident rate increases starting at 8:00 A.M., and the period with the highest accident rate is 4:00 to 8:00 P.M. We use Xitun District, which has the highest proportion of accidents in Taichung City, as our example. We analyzed the accident records for small passenger vehicles and motorcycles on weekdays and holidays (Fig. 14).

During weekdays, collisions between small passenger vehicles and motorcycles were more frequent during rush hour and the likelihood of a collision decreased between 8:00 A.M. and 2:00 P.M. At weekends, small passenger vehicle collisions increased slowly from 8:00 A.M., whereas collisions involving motorcycles showed no notable hourly

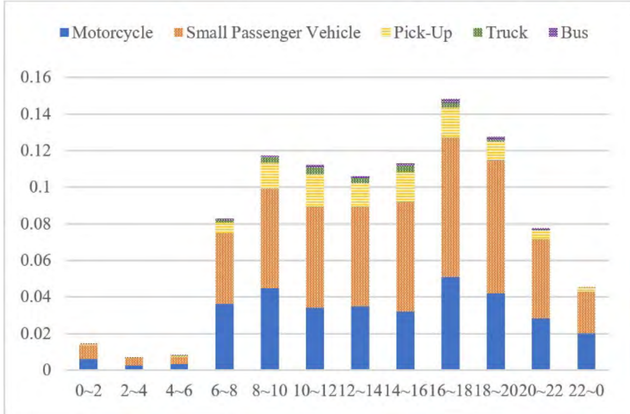

FIGURE 14. Ratio of accidents by transportation type and time of day in Xitun District.

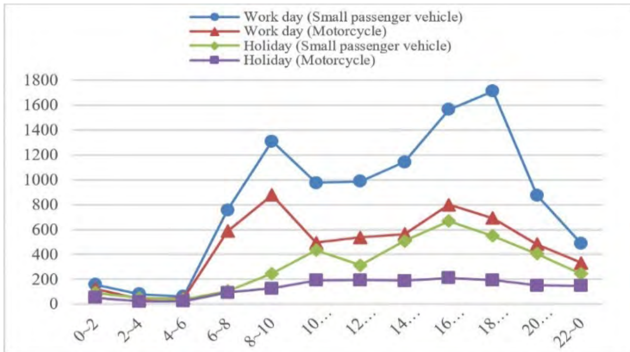

FIGURE 15. Analysis of accident periods for weekdays and weekends.

change. The distribution of collisions differed substantially between weekdays and holidays, especially among motorcycles. This is largely because many people use motorcycles as their main method of transportation in Taiwan. Notably, although Taiwan has the highest motorcycle density in the world, the rate of accidents caused by motorcycles is lower than that caused by cars (Fig. 15).

We analyzed the number of events involving small passenger vehicles and motorcycles. As shown in Fig. 16, for drivers aged 20–30 years, motorcycle events were slightly more frequent than those involving small passenger vehicles; motorcycles events decreased sharply with age after peaking at 20$\sim$30 years.

VI. EXPERIMENTAL RESULTS

We use the REST API, which is the most widely used interface in service-oriented architecture, as a control and compare it with the proposed system using the MOM architecture. Under the same service, the communication interfaces of each service are redesigned using the REST API. The RESP API is developed using the ASP.NET MVC API, and the client specifications are Intel i5-4460 3.2 GHz, 16 GB of RAM, and Windows 10. The server specifications are Intel i7-7700, 16GB of RAM, and Windows Server 2012. The network environment is a local network.

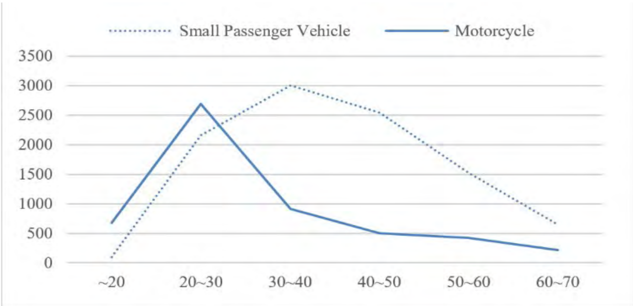

FIGURE 16. Distribution of accidents involving small passenger vehicles and motorcycles by age.

FIGURE 17. Delay time in experiments.

TABLE 4. REST API and our system delay time.

	Average time (ms)	Standard deviation	Total time (ms)
REST API	59.242	17.051	59242
Our Framework	13.934	12.796	13934

A. DELAY TIME COMPARISON

In the experiment design, MQ and HTTP are very different protocols. MQTT transmits/subscribes messages using multiple threads, and therefore, its performance is better than that of REST, which uses a single thread for request and response. To ensure a fair result, the design of the control group creates a thread when a client provides a service, and we also use the POST method. Each service should obtain a calculation result without polling; this design improves the performance of the REST API by sending and receiving results through one connection. The weakness of this method is that it also reduces the service turnover ratio of the REST API. However, this experiment focuses on the performance of HTTP and MQ in different architectures; thus, the turnover ratio is not considered. During the experiment, we only send the coordinate from the same accident hotspot to ensure that the message length is the same. Fig. 17 shows the test delay time obtained in the experiment. We used 1000 latitude and longitude data points for our experimental simulation; the results are presented in Table 4.

The results of the comparison of the REST API system and our framework reveal a more than fourfold performance gap

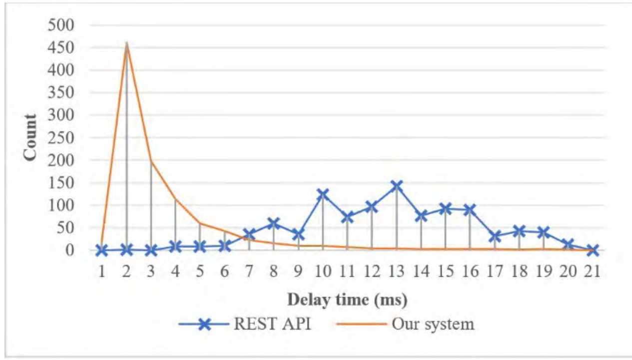

FIGURE 18. Delay time distribution.

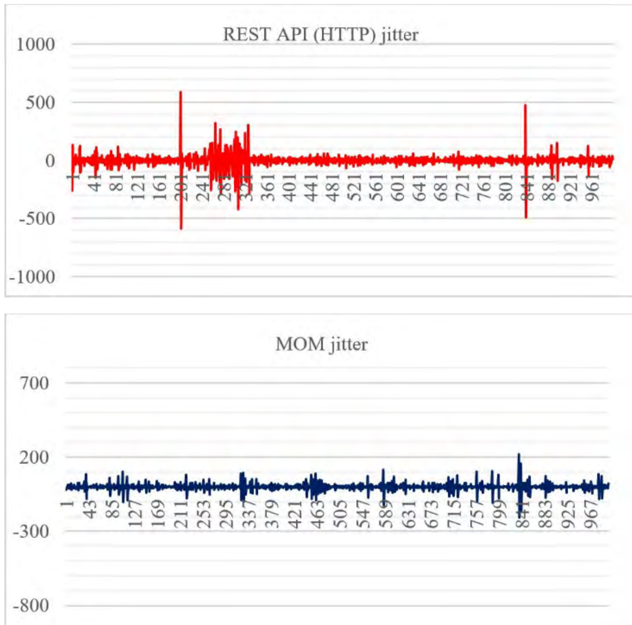

FIGURE 19. Jitter rate comparison.

in the whole system processing time. In terms of stability, we calculated the distribution of data (Fig. 18).

In experiments to determine the delay time counts, the delay time distribution obtained using the MOM framework was between 2 and 10 ms; however, that obtained using the REST API oscillates between 6 and 20 ms, as shown in Fig. 18. The results thus indicate that the MOM architecture is relatively stable.

B. JITTER RATE COMPARISON

We analyzed the jitter rate of the REST API and MOM (Fig. 19). The result reveals that the average jitter rate of the REST API is 0.229 and that of MOM is 0.064. According to this experiment, the transmission performance of MOM is better than that of the HTTP API, and the jitter rate also demonstrates that MOM has superior reliability.

C. SUMMARY

HTTP has long headers. The server can determine user requests and give responses according to these headers. In an

IoT or IoV environment, complex headers waste transmission time. For example, the default limit is 8 KB in Apache and 16 KB in IIS. The general Message Queue protocol header is shorter, e.g., a MQTT header is only 2 bytes [7].

A typical HTTP connection is not sustainable. It begins when the user requests it and must endure until the server has output. However, if the user is not willing to wait for the previous request to be answered, and requests the data again, the connection must be re-established and the server data must be re-read. This causes frequent polling and can lead to servers being paralyzed with Distributed Denial of Service (DDoS) attacks and similar problems. Therefore, we must avoid this type of paralysis if the system has several users.

Many previous studies have demonstrated that the transmission efficiency of Message Queue is higher than that of HTTP. The four protocol comparisons show that AMQP and MQTT are better than HTTP in terms of both bandwidth and latency [8], [9]. Our system is based on MOM with MQTT and AMQP. The experimental results are also consistent with the literature.

VII. CONCLUSIONS

In the design of our framework, we considered its possible application carefully, and the resulting MOM framework was found to obtain superior results to the HTTP-based architecture. With great developments in self-driving and advanced driver assistance system technologies, vehicles will undeniably become safer than they are currently. However, even the safest vehicles cannot avoid accidents caused by external factors, such as surroundings, other vehicles, or pedestrians. Thus, defensive driving based on big data analysis is necessary for improving driving safety. This paper proposes a rapid-response framework that enables defensive driving by not only providing proper alert messages to drivers but also improving drivers' awareness of accident risk. After proper encoding, it will be able to provide risk assessments to self-driving vehicles as a basis for appropriate defensive driving operations. Our framework proposes a favorable concept for integrating the IoV with self-driving applications.

We attempted to use a random forest and BPN to construct the analysis model; however, these models were infeasible because they resulted in overfitting. This situation can occur because of the data distribution, with large amounts of data in a hotspot, or because the amount of data is insufficient to construct an accurate model. We use an Apriori algorithm to calculate factor associations. This algorithm can obtain the optimal result even when data are insufficient. To evaluate the prediction precision, more data is still required. In the future, we will continue to collect data and a deep learning method will be applied to this model to improve its accuracy.

REFERENCES

[1] P. Stahl, B. Donmez, and G. A. Jamieson, "Anticipation in driving: The role of experience in the efficacy of pre-event conflict cues," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 5, pp. 603–613, Oct. 2014.

[2] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, Nov. 2016, pp. 459–464.

[3] M. Shimosaka, T. Kaneko, and K. Nishi, "Modeling Risk Anticipation and Defensive Driving on Residential Roads with Inverse Reinforcement Learning," in *Proc. IEEE 17th Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 1694–1700.

[4] M. Richtel and C. Dougherty. (Sep. 1, 2015). *Google's Driverless Cars Run Into Problem: Cars With Drivers*. [Online]. Available: http://www.nytimes.com/2015/09/02/technology/personaltech/google-says-its-not-the-driverless-cars-fault-its-other-drivers.html?_r=1

[5] Y. Fangchun, W. Shangguang, L. Jinglin, L. Zhihan, and S. Qibo, "An overview of Internet of vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[6] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Vehicular clouds: State of the art, challenges and future directions," in *Proc. IEEE Jordan Conf. Appl. Elect. Eng. Comput. Technol. (AEECT)*, Nov. 2015, pp. 1–6.

[7] P. Thota and Y. Kim, "Implementation and Comparison of M2M protocols for Internet of Things," in *Proc. 4th Int. Conf. Appl. Comput. Inf. Technol.*, Las Vegas, NV, USA, Dec. 2016, pp. 43–48.

[8] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, Vienna, Austria, Oct. 2017, pp. 1–7.

[9] A. Saxena and R. Prakash, "Universal BLDC controller—With IIoT set of features," in *Proc. Int. Conf. Nextgen Electron. Technol., Silicon Softw. (ICNETS)*, Chennai, India, Mar. 2017, pp. 419–425.

[10] F. Aloul, I. Zualkernan, R. Abu-Salma, and H. M. Al-Aliand Al-Merri, "iBump: Smartphone application to detect car accidents," in *Proc. Int. Conf. Ind. Autom., Inf. Commun. Technol. (IAICT)*, Bali, Indonesia, Aug. 2014, pp. 52–56.

[11] H. Jiang, L. Zhong, C. Li, and H. Feng, "Research on identification method for road accident black spots with ordinal clustering method," in *Proc. Remote Sens., Environ. Transp. Eng. (RSETE)*, Nanjing, China, Jun. 2011, pp. 2401–2404.

[12] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver behavior analysis for safe driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3017–3032, Dec. 2015.

[13] A. U. Jadhav and N. M. Wagdarikar, "A review: Control area network (CAN) based intelligent vehicle system for driver assistance using advanced RISC machines (ARM)," in *Proc. Int. Conf. Pervasive Comput. (ICPC)*, Pune, India, Jan. 2015, pp. 1–3.

[14] K. M. A. Solaiman, M. M. Rahman, and N. Shahriar, "AVRA BANGLADESH collection, analysis & visualization of road accident data in Bangladesh," in *Proc. Int. Conf. Informat., Electron. Vis. (ICIEV)*, Dhaka, Bangladesh, May 2013, pp. 1–6.

[15] M. C. G. Quintero and P. A. C. Cuervo, "Intelligent driving assistant based on accident risk maps analysis and intelligent driving diagnosis," in *Proc. Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, Jun. 2017, pp. 914–919.

[16] E. Chen. (Apr. 18, 2015). *75% of the World's Cars Will be Connected by 2020, Reports*. [Online]. Available: http://press.trendforce.com/node/view/1880.html

[17] Z. Meng *et al.*, "Guaranteed V2V QoS services implementation and field measurements in hybrid WAVE LTE environments," in *Proc. IEEE Region Conf. (TENCON)*, Macao, China, Nov. 2015, pp. 1–6.

[18] S. Imadali, A. Kaiser, S. Decremps, A. Petrescu, and V. Veque, "V2V2I: Extended inter-vehicles to infrastructure communication paradigm," in *Proc. Global Inf. Infrastruct. Symp.*, Trento, Italy, Oct. 2013, pp. 1–3.

[19] V. Grigoryev, I. Khvorov, Y. Raspaev, and E. Grigoreva, "Intelligent transportation systems: Techno-economic comparison of dedicated UHF, DSRC, Wi-Fi and LTE access networks: Case study of St. Petersburg, Russia," in *Proc. Telecommun., Media Internet Techno-Econ. (CTTE)*, Munich, Germany, Nov. 2015, pp. 1–8.

[20] R. I. Meneguette, "Peer-to-peer protocol for allocated resources in vehicular cloud based on V2V communication," in *Proc. Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, May 2017, pp. 1–6.

[21] H. Meng, K. Zheng, P. Chatzimisios, H. Zhao, and L. Ma, "A utility-based resource allocation scheme in cloud-assisted vehicular network architecture," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1833–1838.

[22] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Seoul, South Korea, Mar. 2014, pp. 241–246.

[23] D. Parlanti, F. Paganelli, and D. Giuli, "A service-oriented approach for network-centric data integration and its application to maritime surveillance," *IEEE Syst. J.*, vol. 5, no. 2, pp. 164–175, Jun. 2011.
[24] Y. Morais and G. Elias, "Experimental evaluation of a multi-paradigm, message-oriented mobile middleware," in *Proc. Int. Conf. Inf. Technol., New Generat.*, Las Vegas, NV, USA, Apr. 2014, pp. 214–219.
[25] W. Chongnan, W. Zongtao, and X. Hongwei, "Design of message-oriented middleware with publish/subscribe model on telemetry and command computer," in *Proc. Int. Conf. Syst. Informat. (ICSAI)*, Shanghai, China, Nov. 2014, pp. 454–458.
[26] J. Labéjof, A. Léger, and P. Merle, "R-MOM: A component-based framework for interoperable and adaptive asynchronous middleware systems," in *Proc. Int. Enterprise Distrib. Object Comput. Conf. Workshops*, Beijing, China, Sep. 2012, pp. 204–213.
[27] F. Z. Benchara, M. Youssfi, and O. Bouattane, "A new efficient distributed computing middleware based on cloud micro-services for HPC," in *Proc. Multimedia Comput. Syst. (ICMCS)*, Marrakech, Morocco, Oct. 2016, pp. 354–359.
[28] M. Vains and K. Urbaniec "Employing Bayesian networks and conditional probability functions for determining dependences in road traffic accidents data," in *Proc. Smart City Symp. Prague (SCSP)*, Prague, Czech Republic, May 2017, pp. 1–5.
[29] H. K. D. Sarma and S. Mishra, "Mining time series data with Apriori tid algorithm," in *Proc. Inf. Technol. (ICIT)*, Bhubaneswar, India, Dec. 2016, pp. 160–164.
[30] M. O. Sarkan, A. Akcakoca, and C. Kucukakdag, "Alarm correlation using Apriori algorithm," in *Proc. Signal Process. Commun. Appl. Conf. (SIU)*, Turkey, Malatya, May 2015, pp. 1602–1605.
[31] J. Olusina and W. A. Ajanaku, "Spatial analysis of accident spots using weighted severity index (WSI) and density-based clustering algorithm," *J. Appl. Sci. Environ. Manage.*, vol. 21, no. 2, pp. 397–403, Apr. 2017.
[32] J. S. Chen, "A study of traffic accident-prone intersection and cause analysis in Taichung City," M.S. thesis, Dept. Traffic Eng., Feng Chia Univ., Taichung, Taiwan, 2010.
[33] *Asian Economy News Agency*. Accessed: Jun. 15, 2017. [Online]. Available: http://www.myaena.net/newspaper.php?nn_id=54&news_id=11018

PO-YU LAI received the M.S. degree from the Institute of Applied Information Technology, Ling Tung University, Taiwan, in 2011. He is currently pursuing the Ph.D. degree with the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include IoT, metaheuristics, and wireless sensor networks.

CHYI-REN DOW was born in 1962. He received the B.S. and M.S. degrees in information engineering from National Chiao Tung University, Taiwan, in 1984 and 1988, respectively, and the M.S. and Ph.D. degrees in computer science from the University of Pittsburgh, Pittsburgh, PA, USA, in 1992 and 1994, respectively. He is currently a Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include mobile computing, ad hoc wireless networks, telematics, and IoT.

YU-YUN CHANG received the B.S. degree from the Department of Computer Science and Information Engineering, Providence University, Taiwan, in 2009. She is currently studying in the Industrial Ph.D. Program of the Internet of Things (IoT), Feng Chia University, Taiwan. Her research interests include IoT, big data, and data mining.