2017 IEEE 6th International Congress on Big Data

Drowsy Driving Warning System based on GS1 Standards with Machine Learning

Seongjoo Moon[1, 2], Moran Min[2], Jangwoo Nam[2,3], Jaeyou Park[2], Doheon Lee[1,2], Daeyoung Kim[2,4]
[1]*Mando, Pankyo, Kyungki, Korea*
[2]*Graduate School of Software, KAIST, Seoul, Korea*
[3]*Solutionlink, Seocho, Seoul, Korea*
[4]*Auto-ID Labs, KAIST, Daejeon, Korea*
{sj.moon, minmoran, jnam, cpuu, doheon, kimd}@kaist.ac.kr

Abstract— **Drowsy driving is the primary cause of motor vehicle accidents and is a risk factor that leads to the loss of human life, remaining as a challenge for the global automotive industry. Recently, drowsy monitoring system has been actively studied for prediction system based machine learning. However, the challenges of automotive real-time constraints and flexibility should be taken into consideration against a large amount of heterogeneous data from vehicle network and other device. To solve this problem, we propose drowsy monitoring system based machine learning using GS1 standard. First, vehicle motion data is defined and modeled using the GS1 standard language for drowsy predict. Second, we propose an optimal algorithm selection and detail architecture for automotive real-time environments through machine learning algorithms (KNN, Naïve Bayes, Logistic Regression) and deep learning algorithms (RNN-LSTM). Finally, we describe system-wide integration and implementation through the open source hardware Raspberry Pi and the machine learning SW framework. We provide optimal LSTM architecture and implementation that takes into account the real-time environmental conditions and how to improve the readability and usability of the vehicle motion data. We also share the rapid prototyping methodology case of connected car systems without other sensor devices.**

Keywords-Drowsy Monitoring System; GS1 Global Standards; Machine Learning; LSTM; Connected Car

I. INTRODUCTION

The research on the drowsy driving monitoring system to prevent drowsy driving can be roughly divided into two kinds of methods. It is a direct monitoring system that warns if a drowsy signal is detected by recognizing the driver's face, measuring brain waves, pulse, etc., and an indirect monitoring system using the dynamic motion of the vehicle[1]-[2]. These two drowsy monitoring systems have been actively studied for the development of computational power and machine learning algorithms[3].

However, such direct and indirect drowsy monitoring system based on machine learning has several problems. First, the automobiles are a real-time embedded system, and if the drowsy prediction is not performed in real time, it is related to the safety of the driver. Therefore, the time constraint condition of the machine learning prediction algorithm is very important. In addition, the prediction accuracy of drowsy driving should be high. In this context, the two optimization problems of execution time and accuracy machine learning algorithm have to be solved in automotive embedded system environment. Second, the direct monitoring system must detect driver's physical information to determine drowsy driving, so driver physical information sensor is inevitably needed. But a human physical information sensor is not mounted on an automobile, so external sensor must be attached to a vehicle or a driver must be detached every time the vehicle is operated. Third, vehicle motion information using indirect monitoring system is different and not unified for each country, automaker and supplier. In the drowsy monitoring system based machine learning, the processing technique of data is important. When loading stored data, the data processing technique should be standardized and unified so that other engineers can read and process it. That's how other engineers can use machine learning algorithms and increase their extension and accuracy.

In order to solve this problem, we focus on the problem solving and implementation of an indirect monitoring system which does not require other sensors. First, we apply GS1 Electronic Product Code Information Service (EPCIS)[4] standards for Internet of Things (IoT) products to enhance the sharing and utilization of big data in vehicle behavior information. Data on the vehicle have been limited in the use of information until now because of the enormous amount of data in the real-time environment. However, if the vehicle information can be stored effectively and other users can be reused, it will have a great impact on the automotive industry by combining it with machine learning technology. In this context, the GS1 EPCIS standard in the automotive industry has advantages such as versatility and reliability in the IoT field and infra already built. Using the GS1 EPCIS standard, vehicle motion information such as steering angle, vehicle speed, steering torque, steering angle speed, and yaw rate data is defined and modeled to be used in training data sets for machine learning algorithms. Secondly, we use the processed data to select the machine learning algorithm optimized for real-time environmental conditions and design the architecture. The machine learning execution time and accuracy of algorithm prediction are a conflicting performance in

978-1-5386-1996-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataCongress.2017.44

automotive real-time embedded environments. In order to optimize two conflicting performances, data feature selection was performed and compared to each case. And then we compared the machine learning algorithm (KNN, Naïve Bayes, Logistic Regression) and the deep learning algorithm (RNN-LSTM). After researching optimal machine learning algorithm, we select Long Short-Term Memory (LSTM). In order to process real-time data of vehicle the optimal combination of time-window processing, LSTM layer processing, data normalization processing and LSTM input data after the comparative evaluation is proposed. Third, through the proposed technical approach, how to implement the actual product is introduced. We use CAN as vehicle communication network, open source hardware which can be easily accessed by engineers who want to implement connected cars and autonomous driving cars, and machine learning SW frameworks such as Theano and Keras. It will provide a making prototype guideline for engineers to implement connected cars and autonomous vehicles based on automotive CAN network.

The composition of this paper is as follows. Section 2 introduces the GS1, EPCIS[5] and Oliot[6] which is EPCIS open source service as a related study of this study. In Section 3, we propose a system architecture proposal and vehicle motion data modeling according to EPCIS standard and technical approach of optimal LSTM architecture for the proposed system. Section 4 describes the implementation and integration of each module of the actual prototype as an implementation step.

II. BACKGROUND

A. GS1

GS1 is a non-profit international organization that develops and disseminates standards for the identification, capture, sharing, and use of object information for the realization of a data-driven global business ecosystem.

GS1 standardizes the identification system of objects, data, and service sharing method in response to business requirements of various industries, and creates service standard by analyzing industry-specific process such as a medical and smart factory.

The GS1 standard standardizes a global identification code system to identify, track and manage objects and goods at all stages of production, distribution, and consumption (utilization) of objects so that each company or service has its own goods, location, enterprise, and service. The GS1 international standard, a data-based IoT standard for objects, can also be defined as the international standard language for the IoT, which allows objects to be created or related information in a single language for the purpose of providing global visibility of objects.

The IoT GS1 international standard consists of two main components.

1) GS1 ID key, GPC(Global Product Classification)

It corresponds to the noun of the standard. GS1 ID keys objects in various industrial domains define a strong identification system that covers everything from products to objects, services, physical places and people in the IoT service. In addition to class-level IDs that can distinguish it also provides an instance-level ID that can distinguish 'things'. For example, GTIN is an identifier that identifies the type of object, and SGTIN (Serialized GTIN) is an identifier that contains the object's serial number.

2) CBV(Common Business Vocabulary)

EPCIS is a globally distributed data repository operated by each company or application. It enables to store and share information directly or indirectly from various objects in a standardized document format. It provides capturing API and query API for data storage and query. Distributed EPCIS data stores around the world operate as one global database through standard GS1 ID Keys.

In the automotive industry, CBV (Common Business Vocabulary) of the GS1 standard can be expressed and stored in the EPCIS, such as automobile assembly, design, production, operation, maintenance and disposal.

B. EPCIS (Electronic Product Code Information Service)

The CBV provides a definition of a common standard vocabulary that can be used to describe the dynamic event data of an object in EPCIS which is the object IoT standard shared data store. The standard vocabulary is derived by analyzing the business process of an entire life cycle or industrial domain of an object. By using the standardized vocabulary provided by the CBV standard to unify the context, it is possible to share information by helping interoperability between business or industry domains and reducing the difference in the way in which different business systems express common intent make it easy. GS1 also provides a standard way to define the user vocabularies required by each country, industry or company.

C. Oliot

Oliot is an IoT open source platform for things based on the EPCIS standard and includes extensions focused on new features and capabilities for the IoT while providing a reference implementation of the latest version of the GS1 EPCIS standard and related services.

The Oliot platform consists of data filtering, collection middleware (Oliot F & C), based database systems (Oliot EPCIS), service registries (Oliot ONS, DS) and applications (Oliot Pedigree, Traceability & Recall) from the connectivity (IoT Connectivity Layer) Metadata Service (Oliot GS1 Source) is an IoT infrastructure platform. The IoT Connectivity Layer is used to filter and collect data of various objects collected through F & C, and store them in time series in EPCIS for use in high-level applications such as ONS, DS, Traceability & Recall. In particular, EPCIS, a database system that is the framework of the Oliot platform, is specialized in global object identification and data management because it is designed as a distributed system that can operate effectively in various companies and organizations.

Figure 1. Oliot Framework

III. ARCHITECTURE

A. Overall Architecture

The proposal comprises largely of three kinds of functions:

A machine learning-based drowsiness classifying function, a drowsy driving warning function with predicted information, and a data display function. The embedded system processes the vehicle information and then transmitted to the server. The transmitted data is studied and the learning model is ported back to the embedded system. The ported learning model predicts drowsiness and the predicted drowsiness pattern will be alerted on the smart phone.

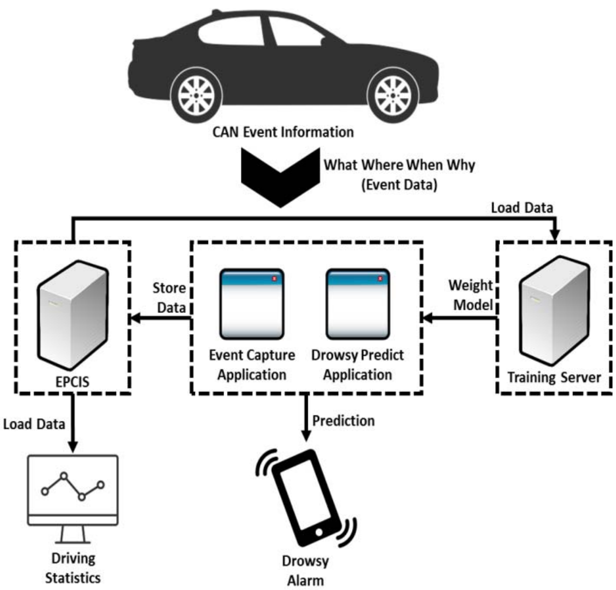


Figure 2. System Architecture

B. Data Definition using EPCIS

1) Data Definition

According to the GS1 standard, the entire life cycle activity of an automobile defines all stages based on the GS1 Core Business Vocabulary (CBV). However, using only the CBV value makes it difficult to represent real-time events in a vehicle. In particular, the driver drowsy monitoring system proposed in this paper should focus on the driver's driving condition. In addition, since the driver's drowsiness is greatly affected by the car accidents depending on the level of drowsiness, the class of drowsy driving should be classified according to the grade as shown in Table 1 [7].

So we define and propose new detail events for the vehicle. This has the advantage of integrating the data collected from other systems because event data is collected based on the same definition.

TABLE I. KAROLINSKA SLEEPINESS SCALE

Scale	Description
1	Extremely alert
2	Very alert
3	Alert
4	Rather alert
5	Neither alert nor sleepy
6	Some signs of sleepiness

7	Sleepy, but no effort to keep awake
8	Sleep, some effort to keep awake
9	Very sleepy, great effort to keep awake fighting sleep

2) Data modeling using GS1 standard format

In order to define the detailed event of drowsy driving, this paper focuses on the lateral motion of the vehicle [8]-[9]. The use of the lateral motion of the vehicle does not require the driver's direct physical information and is advantageous in that it can monitor the driver's drowsiness without decisively mounting external sensor device.

The lateral motion of the vehicle was processed by using the vehicle CAN signal. The actual lateral motion of the vehicle is shown in the Table below.

TABLE II. MODELING INFORMATION

No	Event	Input Data	Drowsy Level
1	Driving	Steering Wheel Angle	1~9
2		Vehicle Speed	
3		Steering Torque	
4		Yaw Rate	
5		Steering Wheel Angle Speed	

Using these data, we store it according to the GS1 data standard and in accordance with the EPCIS XML format. The real-time data of the drowsy driving operation corresponds to the observe action in the EPCIS object event because it is interested in observing the behavior of the product. The EPCIS object event abstraction model is constructed as shown in Table 3.

WHEN is the event in real time, WHAT is the vehicle to measure, WHY is trying to measure drowsy operation. Extension information stores the vehicle lateral motion information necessary for monitoring in drowsy operation because the WHY section has limited real-time information required for drowsy operation measurement.

For example, "A car with a specific ID (SGTIN) is driving on December 15, 2016, and the lateral motion of the vehicle (steering angle 4.5deg, vehicle speed 0kph, steering torque 1.99Nm, steering angle speed 23deg / sec) When measured, the drowsy operation level at that time is 0 (completely awake)". Event information for the example is shown in Table 4.

TABLE III. EPCIS ABSTRACT DATA MODELING

Object Event (Observable)	
WHEN	Real-Time Event Time
WHAT	Car
WHY	Drowsy Monitoring when driver driving
Where	NA
Extension	Vehicle lateral motion data for monitoring

TABLE IV. EVENT MODEL EXAMPLE

EPCIS Event	Event Type	Object Event
	Action	OBSERVE
WHEN	Event Time	2016-12-15T16:19:48.162Z
WHAT	epcList	Urn:epc:id:sgtin:0614141.107346.2017
WHY	bizStep	Driving
Extension	Steering Wheel Angle	4.5
	Vehicle Speed	0
	Steering Torque	-1.99
	Yaw Rate	-0.02
	Steering Wheel Angle Speed	23
	Drowsy Level	0

C. Machine Learning

1) Data set definition

It is important to select the proper input features and determine the suitability of the output labeling for performing machine learning. The description of training set of machine learning is shown in the Table below.

TABLE V. TRAINING SET SUMMARY AND EXAMPLE

Training Set Summary	
Sample of Driver	5 people
Data Size	70,0000
Label	Drowsy Level (awake/sleepy)

Training Set Example							
ref_time (Sec)	WheelAng (Deg)	WheelSpd (Deg/Sec)	DegOfDev (Deg/Sec)	Torque (Nm)	SideAccl (g)	SpeedCom (km/h)	DrowsyLevel
0	-532.35	-134.925	28.1	2.74	-0.11	7.89	0
0.02	-535.5	-134.925	28.64	2.96	-0.12	8	0

First, in the case of input feature selection, it is necessary to judge whether the selected feature is well selected and whether it is an optimized feature selection to satisfy constraints on drowsy prediction time. Among the vehicle lateral motion data stored in the EPCIS, features that greatly affect drowsiness are the steering wheel angle and steering wheel speed[9]. And machine learning algorithms (KNN, Logistic Regression, Naïve Bayes, LSTM) was performed for optimized feature selection. As shown in Figure 3, it is confirmed that the results of using the two selected features are better than those of all feature.

Second, in the case of output labeling, open data [10] is used because currently drowsy driving data is not collected enough. The open data did not fit the purpose of the system of this paper to measure the drowsy state of the current point of view. Thus, the steering angle and the steering wheel speed were used to determine and label the specific drowsy driving condition [9].

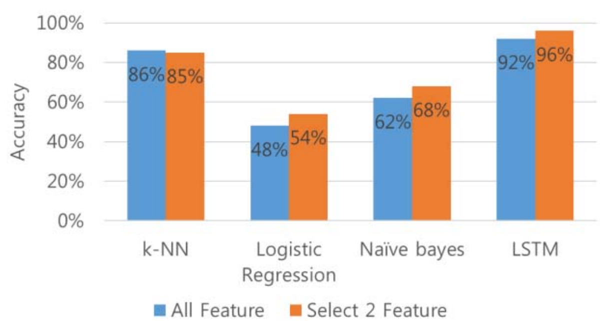


Figure 3. Comparison of feature selection performance

2) LSTM Architecture

The currently selected features, the steering wheel angle and the steering wheel speed are time-dependent signals whose predictions can change in relation to time before and after. Since the time-dependent signal of the vehicle must analyze the flow of the waveform within a certain time interval to predict the drowsy driving, the LSTM machine learning algorithm is suitable for storing the operation information on the previous signal and using it for the next calculation [3].

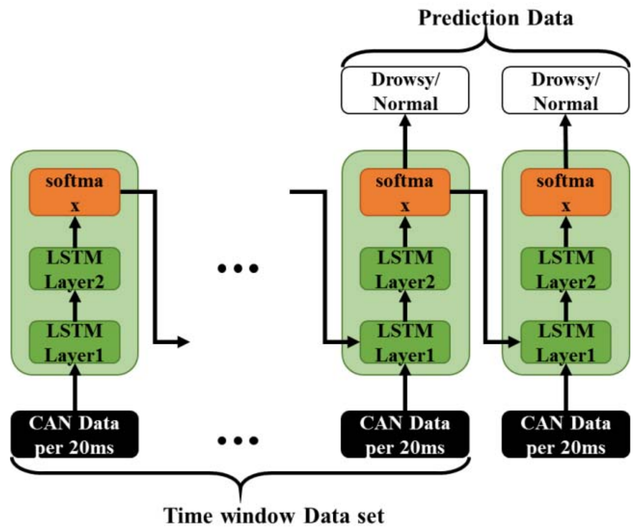


Figure 4. LSTM Network

The LSTM architecture of our system is shown in the Figure 4. It receives CAN data generated in 20ms unit and performs one prediction using time window data set. The LSTM configuration utilized two LSTM layers and one softmax activation.

IV. IMPLEMENTATION

A. Embedded HW implementation

Embedded system is divided into CAN communication signal processing part and machine learning model execution part.

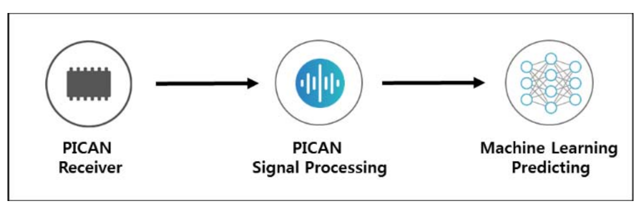


Figure 5. CAN communication signal processor

Embedded System is composed of Open Source Hardware using Rasberry3 and PICAN[11] which is a commercial circuit. The Machine Learning Prediction Unit uses the Theano-based Keras Framework because the learned weight parameters must be transplanted in the machine learning server. Python is used for both signal processing and machine learning predicators.

B. EPCIS implementation

Through the driving event modeling of the vehicle, the XML to be sent to the EPCIS server conforming to the GS1 standard is created. At this time, since CBV suitable for drowsy operation is not applied to GS1 as described above, it is also necessary to create an extended data schema defined by the user.

```
<ObjectEvent>
  <eventTime>2016-12-15T16:19:48.162Z</eventTime>
    <epcList>
      <epc>urn:epc:id:sgtin:0614141.107346.2017</epc>
    </epcList>
    <action>OBSERVE</action>
    <car:Steering_Angle>4.5</car:Steering_Angle>
    <car:SAS_Speed>0.0</car:SAS_Speed>
    <car:Drive_Torque>1.0</car:Drive_Torque>
    <car:Vechlie_Speed>0.0</car:Vechlie_Speed>
    <car:Steering_Torque>-1.19999999999</car:Steering_Torque>
    <car:Yaw_Rate>-0.0200000000000</car:Yaw_Rate>
    <car:SAS_Angle>100.0</car:SAS_Angle>
    <car:drowsy>0.0</car:drowsy>
</ObjectEvent>
```

Figure 6. Driving Event Model Implementation

XML based on the final GS1 standard based on data modeling is shown in Figure 6.

C. LSTM implementation

Based on the theoretical reasoning, LSTM is applied to the training server. Gradient descent variants, normalization, batch size and each layer hidden units were selected through experiments. The following Figure shows the creation code of LSTM model in Keras framework.

```
from keras.models import Sequential
from keras.layers.core import Dense, Activation, Dropout
from keras.layers import Embedding
from keras.layers.recurrent import LSTM, GRU
from keras.optimizers import SGD,RMSprop,Adadelta

in_out_neurons = 2
num_of_vectors =500

model = Sequential()
model.add(LSTM(100, input_shape=(num_of_vectors, in_out_neurons), return_sequences=True))
model.add(LSTM(2, input_shape=(num_of_vectors, 100)))
model.add(Dense(2))
model.add(Activation("softmax"))
sgd = SGD(lr=0.00001, decay=1e-6, momentum=0.9, nesterov=True)
RMSprop = RMSprop(lr=0.003, rho=0.9, epsilon=1e-08, decay=0.0)
model.compile(loss='binary_crossentropy',optimizer='RMSprop',metrics=['accuracy'])
model.fit(X_train, y_train, batch_size=50, nb_epoch=50, validation_split=0.0)
```

Figure 7. LSTM Application

In order to investigate the performance of the implemented LSTM, we compared with other machine learning algorithms.

First, we conducted a comparative evaluation of prediction accuracy. Among the machine learning algorithms, LSTM shows the best accuracy when performing learning and testing using a feature that has three algorithms: Logistic Regression, Naïve Bayes, and LSTM.

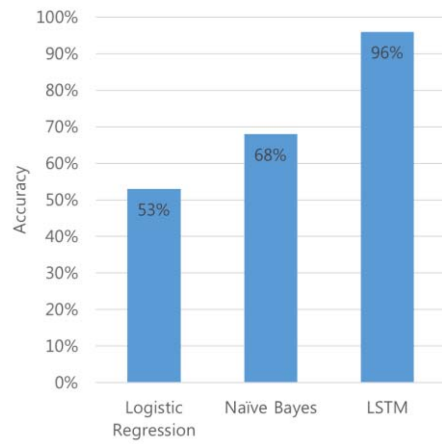


Figure 8. Comparison of Accuracy by Machin Learning algorithm

Second, we compared the execution time of prediction. LSTM has the highest accuracy, but it takes 0.003 seconds, which is the most time to perform per one predication. However, since the system recognizes the danger of drowsiness, the total warning time is usually set to less than one second, and there is no abnormality in the timing of the prototype test. Therefore, the system uses LSTM for drowsy prediction, it is judged that there is no problem.

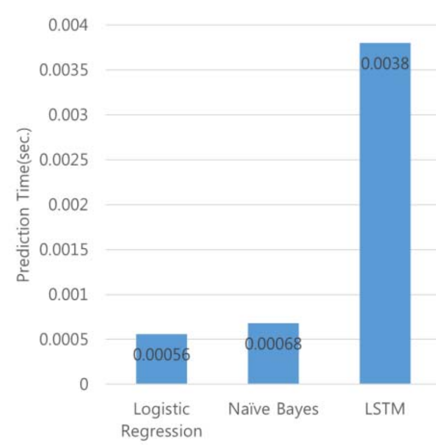


Figure 9. Comparison of Prediction Time by Machin Learning algorithm

D. Smartphone Warning implementation

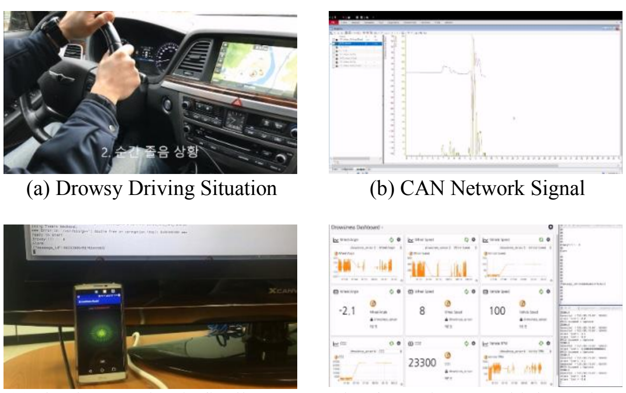
When the driver is drowsy, he uses Fire Cloud Message (FCM), which is a service that can send push messages between the server and client application provided by Google, as a way to alert the driver to the Android terminal. First, when an event occurs, a client (an Android terminal) to receive the event is registered in the FCM server. When the driver is drowsy in the machine learning predicting process and sends a signal to the FCM server in the Raspberry Pi environment, the alert service of the client registered in the server is finally executed. The client receives the beeping signal through the Android terminal so that the driver can break the sleepiness.

E. Integration

Fig 10 is a service scenario of the integrated system.

The actual vehicle test was conducted assuming simulated drowsy driving. The drowsy driving operation assumes a zig-zag state of the vehicle motion, which is a typical drowsy driving pattern.

When a zig-zag pattern occurs in the vehicle, abnormal information of the vehicle motions is transmitted on the CAN. And then information is sensed by the machine learning prediction algorithm, and finally, the drowsy alarm signal is transmitted to the smartphone. The transmitted information sounds the alarm sound of the smartphone, and the vehicle information and the drowsy detection information at that time are stored and displayed on the server. Figure 10 shows the scenario situation of the actual test of the vehicle.



(a) Drowsy Driving Situation (b) CAN Network Signal

(c)The drowsy alert is finally sent to the smartphone. (d)Information on vehicle motions on the CAN network is detected

Figure 10. Vehicle Test Demo Scenario

In Figure 10.(a) we simulated drowsy driving situation using a real vehicle. Figure 10.(b) is the CAN signal generated during the simulation was collected through the developed embedded system. Figure 10.(c) shown the system is installed with an application that detects drowsy driving and sends an alarm to the driver when it detects drowsy driving. Figure 10. (d) shown the data collected from the embedded system is sent to EPCIS, and these data are visualized so that the user can analyze the operation pattern using a platform[12] capable of the data analysis.

V. CONCLUSION

We propose a new IoT platform and device for GS1-based drowsy driving monitoring. Firstly, the indirect drowsy driving monitoring system is implemented using driving pattern data of the vehicle using the CAN Protocol. In addition, GS1 EPCIS which is a global IoT standard has been used to standardize vehicle driving data, thereby enhancing data reproduction and readability. This reproduced data improved the understanding of data interpretation in machine learning. Secondly, this paper shows that drowsy driving monitoring is implemented with a deep learning LSTM algorithm, which improves accuracy and satisfies real-time time constraints. Finally, we show that the drowsy monitoring system can be implemented as an open hardware platform, machine learning SW framework, EPCIS server and smartphone.

The proposed method improves the readability and reusability of vehicle data. And we will contribute to the implementation of rapid prototyping product by autonomous driving and big data because we have implemented the open source hardware and machine learning framework at present.

Currently, it is not easy to acquire real data for drowsy driving, so we use open data set. After real data acquisition, we will improve the accuracy of machine learning prediction and real-time environment optimization.

REFERENCES

[1] Paul R. Davidson ,Richard D. Jones and Malik T. R. Peiris, “EEG-Based Lapse Detection With High Temporal Resolution” 2007 IEEE Transactions on Biomedical Engineering, 2007, pp. 832-839.

[2] Singh, Sarbjit and Papanikolopoulos, N.P. ―Monitoring Driver Fatigue Using Facial Analysis Techniques‖, IEEE Intelligent Transport System Proceedings (1999), pp. 314-318.

[3] Martin W¨ollmer, Christoph Blaschke, Thomas Schindl, Bj¨orn Schuller,Berthold F¨arber, Stefan Mayer and Benjamin Trefflich “On-line Driver Distraction Detection using Long Short-Term Memory” 2011 IEEE Transactions on Intelligent Transportation Systems,2011,pp. 574-582.

[4] K. Traub et al. “The GS1 EPC global architecture framework, version 1.6,” GS1 EPCglobal, Tech. Rep. GS1, 2014.

[5] GS1 General Specification, GS1 Standards, 2017.

[6] J. Byun and D.Kim, “Oliot EPCIS: New EPC information service and challenges towards the Internet of Things,” 2015 IEEE International Conference on RFID (RFID), San Diego, CA, 2015, pp. 70-77

[7] Gillberg, Mats, Göran Kecklund, and Torbjörn Åkerstedt. "Relations between performance and

subjective ratings of sleepiness during a night awake." Sleep: Journal of Sleep Research & Sleep Medicine (1994)

[8] Fabian Friedrichs and Bin Yang "DROWSINESS MONITORING BY STEERING AND LANE DATA BASED FEATURES UNDER REAL DRIVING CONDITIONS" 18th European Signal Processing Conference, 2010, pp. 209-213.

[9] Azim Eskandarian, Riaz Sayed, Pierre Delaigue, Jeremy Blum, and Ali Mortazavi "Advanced Driver Fatigue Research" U.S. Department of Transportation Federal Motor Carrier Safety Administration, 2007 pp. 98-108.

[10] D. HAUPT, P. HONZÍK, P. KUČERA and O. HYNČICA, "CLASSIFICATION OF DRIVER'S DROWSINESS FROM STEERING WHEEL MOTION pp. 428-43 UNDER REAL TRAFFIC CONDITIONS" in Proceedings of the 18th International Conference on Soft Computing, 2012, 3

[11] PICAN CAN-Bus Board
http://skpang.co.uk/catalog/pican-canbus-board

[12] Thing+ - Internet of Things Cloud platform
https://thingplus.net/