V-tracer: a Vehicular Trace Generator for Future Predictive Maintenance

Mirialys Machin, Piedad Garrido, Francisco J. Martinez
iNiT Research Group
University of Zaragoza, Spain
{mmachin, piedad, f.martinez}@unizar.es

Julio A. Sanguesa
Centro Universitario de la Defensa
Zaragoza, Spain
jsanguesa@unizar.es

***Abstract*— In this paper we present V-tracer, a vehicular trace generator aimed at generating realistic data about mobility of vehicles, as well as their daily operation and wear. The objectives of our approach are two: first, gathering real traces obtained by in-vehicle on-board units (OBUs), and second, as the first target is hard to achieve, generating synthetic data. The final goal will be getting all the information that would be very useful to infer and predict vehicle failures. The traces provided by our generator may be used to perform the predictive maintenance of vehicles in the near future.**

I. Introduction

Intelligent Transportation Systems (ITS) are intended to improve the operation and safety of transportation through the use of solutions based on information technology and telecommunications [1]. Although recent ITS-based proposals have mainly focused on transport safety applications, other services such as those related to predictive maintenance could be developed. Predictive maintenance services will reduce the maintenance cost by suggesting drivers when they have to replace any component just before its malfunction [2].

To carry out an efficient and accurate predictive maintenance it is necessary to gather and process a massive quantity of data related to vehicles and their behavior, such as diagnostic trouble codes (DTCs), mobility patterns, driving styles, etc. These data would allow extracting information useful to accurately predict some vehicle component faults before the end of their lifetime, or failure takes place.

As gathering real traces obtained by in-vehicle onboard units (OBUs) is currently hard, we consider necessary to generate synthetic data that accurately resemble the data that would be obtained by real vehicles. However, data related to vehicles, their components and weather conditions, are currently distributed in different sources. Additionally, several data are difficult to obtain because they are proprietary or are protected by confidentiality agreements.

In this paper, we describe the implementation of V-tracer, a vehicular data generator primarily aimed at obtaining data related to vehicles mobility, their daily operation, and wear. V-tracer is able to combine real information obtained from in-vehicle OBUs and a synthetic vehicular data generator to obtain realistic traces including routes, weather conditions, and failures. Our data generator will be capable of automatically generating vehicle-related data that, either is currently unavailable, or it would take much more time to gather in real scenarios. In the future, researchers may use these data to build better and more accurate vehicle predictive maintenance systems. Existing works focused their attention on generating data for particular objectives, which are not related to vehicles [3]. Even though some of them are capable of generating more generic data [4], [5], these approaches cannot be directly extrapolated to the vehicular environment.

II. Our Proposal: V-tracer

In this section, we describe V-tracer, a system which generates synthetic data traces related to vehicle environment. The software has been implemented using the Java programming language and the PostgreSQL open source database.

As shown in Figure 1, the vehicular generator architecture can be divided into three different parts. Specifically, we can identify: (i) the input data reading process, (ii) the parameters adjustment process, and finally, (iii) the output data process, where the output data is generated and stored in a database.

II-A. Input data

The information used as the input data of the system is scattered in several sources and stored in different formats. In fact, all the information related to vehicles must be obtained from different sources and has to be unified (we make this in CSV files) to be processed by the vehicular generator. This information includes vehicles' make, model, fuel consumption, their sales percentage [6], fuel tank capacity [7], tires used, the routes followed, the weather conditions (i.e., minimum and maximum temperatures, relative humidity, and precipitations) [8], and finally, the most common vehicles' failures [9].

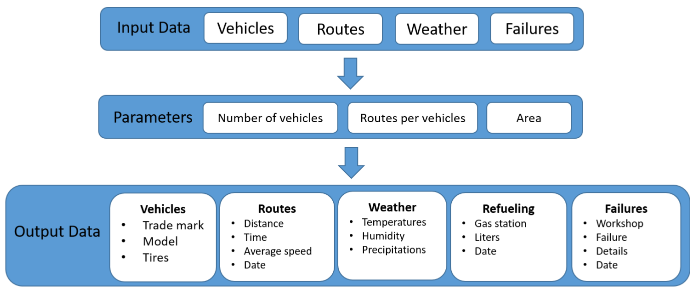

Fig. 1. Vehicular data generator architecture.

II-B. *Parameters*

During the second step, the user provides the parameters to the system. Basically, these parameters will include the number of vehicles to be generated and the number of routes for each one. Also, the user will have the possibility to choose the kind of routes to generate (rural or urban) and the region in Spain. According to the parameters specified by the user, the system will generate the traces during the last step.

II-C. *Generating output data*

The third step comprises the generation of vehicular traces. After receiving the number of vehicles, the system will randomly generate them according to the data of sales in 2016 [6], and it also assigns them different routes. More specifically, a set of routes will be generated for each vehicle including useful information regarding refueling in gas stations, the weather conditions, visits to garages, as well as vehicles' failures.

The generation and the assignation of a route to a vehicle involves different issues, such as determining the weather conditions, or if it is necessary to visit a gas station. The routes including their distance, average velocity, and duration of each trajectory will be generated using the Google Maps Geocoding API [10]. Users will be able to select the number of routes per vehicle and the area where the vehicles can travel. In addition, refueling is automatically generated for each vehicle when the fuel tank is below 25 % of its capacity, although this threshold can be changed. Additionally, the system can generate failures including some details (i.e., date, affected component, problem description, and the location of the garage). Therefore, a set of vehicles along with their routes, the refueling made according to the specified parameters, and the weather in each route will be obtained at the end of the execution. All these data may be used by any vehicles' predictive maintenance system.

III. Results: Simulation Example

Our vehicular data generator was executed with the following parameters: (i) 50 vehicles. This number was chosen to obtain a reasonable sample to test the performance of our proposal, (ii) 5,331 routes, approximately 100 routes per vehicle, and (iii) half of the routes were generated all over the country (i.e., Spain), and the other half were generated in different regions: Aragón, Andalucía, Galicia, and Cataluña.

During the routes generated for the vehicles, 1,233 refueling operations were carried out with a total of 48,876 liters of fuel. In addition, the system generated 81 failures: 13.5 % were in the engine, 29.6 % in the electric system (ES), 12.3 % in the transmission, 9.9 % in the fuel injection system (FIS), 14.8 % in the wheels, steering system, suspension and brakes (WSSB), and the remaining 19.7 % in other systems. Figure 2 shows a comparison between the data generated by the system and the real data obtained by the RAC [9]. As shown, there is a good agreement between the real and the generated data, except for the failures in the electric system and in the transmission.

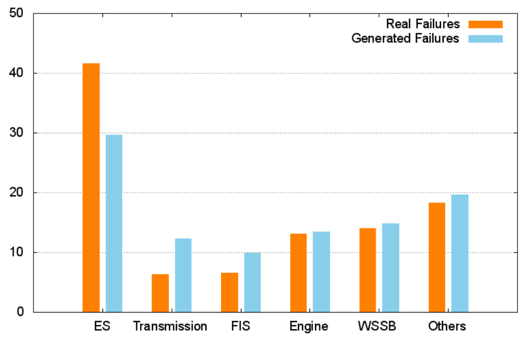

Fig. 2. Real failures [9] and generated failures.

IV. Conclusions

In this paper, we described a synthetic data generator specially designed for vehicle environments. The main objective is to provide realistic vehicles' traces that could be used to develop and train predictive maintenance systems in the future. Unlike other applications that obtain data only from the vehicles, our system is able to generate synthetic traces (i.e., routes, vehicles with their characteristics, weather conditions, refueling and failures) quickly (about 4,000 routes/h) and easily, without the obligation to install any additional equipment and thus saving considerable time and money to obtain data.

V. Acknowledgments

This work has been partially supported by the 2016 Mobility Scholarship for Latin American PhD Students Program granted by the University of Zaragoza and the Santander Bank.

References

[1] Economic Committee, "Directive 2010/40/UE of the european parliament and council," 2010, Available at http://195.76.37.162/NR/rdonlyres/77CCCE9E-3548-41FE-BF7A-AF03023445F2/115152/LexUriServ.pdf.

[2] D. Goyal, A. Saini, S. S. Dhami, and B. S. Pabla, "Intelligent predictive maintenance of dynamic systems using condition monitoring and signal processing techniques—a review," in *International Conference on Advances in Computing, Communication and Automation (ICACCA)(Spring)*, July 2016, p. 1–6.

[3] L. Kelch, T. Pögel, L. Wolf, and A. Sasse, "Traffic generator for hsdpa network simulations," in *International Conference on Connected Vehicles and Expo (ICCVE)*, 2014, p. 325–330.

[4] R. Malhotra, Poornima, and N. Kumar, "Automatic test data generator: A tool based on search-based techniques," in *5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2016, pp. 570–576.

[5] J. Singh and K. Singh, "Designing a customized test data generator for effective testing of a large database," in *International Conference on Advanced Computer Theory and Engineering*, 2008, p. 84–88.

[6] Institute for the diversification of savings and energy (IDAE). Cars Database, 2016, available at http://coches.idae.es/portal/BaseDatos/MarcaModelo.aspx.

[7] Informative cars, 2015, available at http://carerac.com.

[8] National Oceanic and Atmospheric Administration (NOAA), 2016, available at http://www.noaa.gov/.

[9] Reial Automòbil Club de Catalunya, "Las 4 averías de coche más habituales," 2017, Available at http://blog.racc.es/coche-y-moto/las-4-averias-de-coche-mas-habituales/.

[10] Google Maps API. Google Developers, "What is Geocoding?" 2017, Available at https://developers.google.com/maps/documentation/geocoding/intro?hl=en-419.