This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

RoadSense: Smartphone Application to Estimate Road Conditions using Accelerometer and Gyroscope

Azza Allouch, Anis Koubâa, Tarek Abbes, and Adel Ammar

***Abstract*—Monitoring the road condition has acquired a critical significance during recent years. There are different reasons behind broadening research on this field: to start with, it will guarantee safety and comfort to different road users; second, smooth streets will cause less damage to the car. Our motivation is to create a real-time Android Application RoadSense that automatically predicts the quality of the road based on tri-axial accelerometer and gyroscope, show the road location trace on a geographic map using GPS and save all recorded workout entries. C4.5 Decision tree classifier is applied on training data to classify road segments and to build our model. Our experimental results show consistent accuracy of 98.6%. Using this approach, we expect to visualize a road quality map of a selected region. Hence, we can provide constructive feedback to drivers and local authorities. Besides, Road Manager can benefit from this system to evaluate the state of their road network and make a checkup on road construction projects, whether they meet or not the required quality.**

***Index Terms*—Road monitoring, Accelerometer, Gyroscope, Pothole, Real time, Machine learning, Android.**

I. Introduction

According to statistics provided by World Health Organization (WHO), road accidents have become one of the top 10 leading causes of death in the world. Specifically, road accidents claimed nearly 1.25 million lives per year (2015). Studies in [1] show that most road accidents are caused by poor condition of roads. Bad roads are a big problem for vehicles and drivers, this is because the deterioration of roads leads to more expensive maintenance, not only for the road itself but also for vehicles. Accordingly, road surface condition monitoring systems are very important solutions to improve traffic safety, reduce accidents and protect vehicles from damage due to bad roads. Both road managers and drivers are interested in having sufficient information concerning road infrastructure quality (safe or dangerous road).

Consolidated approaches for monitoring road surface conditions involve the adoption of costly and sophisticated hardware equipments such as ultrasonic [2] or specific accelerometers with data acquisition systems [3]. These approaches incur a high installation and maintenance cost and require large manual effort, which can induce error while deploying or collecting the data. Another alternative is to use sensing technologies to gain this information to solve the problem of road surface condition monitoring. These days, smartphones are widely utilized. The greater part of them are equipped with various sorts of sensors like camera, accelerometer, GPS, gyroscope, microphones, etc. Thus, smartphone based road condition monitoring is one of such helpful applications to monitor street conditions.

This paper introduces a road condition monitoring framework which is based on sensors (accelerometer, gyroscope and GPS) built in smartphones to give us the quality of different road sections using machine learning techniques. The contributions of this paper are manifold and can be summarized as follows:

- As a first contribution, we design a machine-learning algorithm (C4.5 Decision tree) to classify road segment as compared to previous works that use simple thresholds, SVM and fuzzy logic. Our tests show that our system is able to detect and classify events related to road conditions with an accuracy of 98,6%.
- Our proposed system, unlike existing solutions that require external hardware, is an inexpensive simple yet efficient solution that is able to monitor road quality. It is realized on Android smartphones and is highly portable and easy to maintain. Our application provide constructive feedback to drivers and local authorities by plotting the evaluated road location on a Map and saving all recorded workout entries.
- Creating an Android application that allows real-time and automatic collection and analysis of accelerometer and gyroscope data in order to get reliable road surface labels in contrast to previous works that mostly use offline methods (videos, images for data labeling).
- While most of previous works employ unimodal accelerometer data, we are using gyroscope sensor in conjunction with accelerometer sensor to derive more accurate road quality prediction.

The rest of this article is organized as follows. Section II presents a background on the three machine learning algorithms used in the paper. Section III introduces some recent research works related to the monitoring of road surface conditions. Section IV describes the general idea and the

Manuscript received March 13, 2017

A. Allouch is with the National School of Electronics and Telecommunication, Sfax 3018, Tunisia, (email: azza.allouch@coins-lab.org).

A. Koubâa is with the Department of Computer Science at Prince Sultan University and research associate in CISTER Research Unit, ISEP- IPP, Portugal, (email: akoubaa@coins-lab.org).

T. Abbes is with the National School of Electronics and Telecommunication, Sfax 3018, Tunisia, (email: tarek.abbes@enetcom.usf.tn).

A. Ammar is with the College of Computer Science and Information Systems, at Al Imam Mohammad Ibn Saud Islamic University (IMSIU), KSA, (email: AhAmmar@imamu.edu.sa).

proposed architecture. Experimental results of the proposed work are presented in Section V. In Section VI, we conclude the paper and we give some perspectives.

II. Background

A. Related Works

In recent years, road condition monitoring has become a popular research area. There has been some works in this field. The most relevant published papers that are analysed are listed in Table I.

The Pothole Patrol [4]; is a sensing application that reports the road surface conditions. It require the integration of particular hardware equipment; for each vehicle an embedded computer running Linux is used for data processing, a Wi-Fi card for transmitting gathered data, an external GPS for localization, and a 3-axis accelerometer to monitor road surface. It uses machine-learning algorithm to detect potholes. Nericell [3] is a system developed by Microsoft to monitor roads and traffic conditions. It requires a very complicated hardware and software setup. It uses several external sensors such as a microphone, GPS, Sparkfun WiTilt accelerometer. The detection is not very accurate (False positive rate less than 10% and false negative rate between 20% and 30%), the system may confuse between smooth, uneven and rough roads. Mednis et al. [5] proposed a real time system for detecting potholes. The system employs Android OS based smartphones having accelerometer sensor and simple algorithms to detect events from acceleration data. Experimental results show a true positive rate equal to 90%. The drawbacks of this work is that the system uses only accelerometer sensor and data are collected through specialized hardware.

Perttunen et al. [6] use a Nokia N95 mounted to a rack on the wind-shield, with accelerometer and GPS to collect data. Labeling is done with a camcorder attached to the headrest of the front passenger seat. However, labeling driving data using video is a time consuming and error prone work. The author [7] describes a pothole detection system. The neural network technique is used for justifying the threshold values and the accuracy is from 90% to 95%. Smartphone accelerometers and gyroscopes are used in [8], [9] to detect road surface anomalies, using an audiovisual data labelling technique with a labeller sitting beside the driver inside the car to mention everything relevant he saw or felt. Then, SVM is used for anomaly detection and classification with an accuracy of 90%. Moazzam et al. [10] used a low-cost Kinect sensor to capture and calculate the approximate volume of a pothole. The use of infrared technology based on a Kinect sensor for measurement is still a novel idea, and further research is needful to decrease error rates. For methods by image processing, Zhang et al. [11] have made use of stereo camera images coupled with a disparity calculation algorithm to identify potholes. Although camera-based approach have been popular in the general field of pothole detection [12]. A recent study uses near infrared (NIR) camera to classify several road conditions; the evaluation has been done in laboratory conditions, and field experiments [13]. However, a drawback of video processing methods is that they are often considered too computationally expensive for smartphone-based implementations. For methods by ultrasonic, an ultrasonic transducer is equipped on vehicles [2]. The ultrasonic waves are continuously emitted to road surfaces and anomalies are detected by the reflection time. However, to obtain high accurate results, an expensive measurement device is required. In [14], the authors present SmartRoad a crowd-sourced road sensing system that can detect traffic lights, traffic regulators, stop signs and road anomalies. Further, S-Road Assist [15] detect road surface and traffic conditions using threshold-based heuristics on data gathered from smartphone sensors. Typically, related studies will threshold the standard deviation of measurements from smartphone-embedded accelerometers to detect road anomalies[16], [17].

Previous works [3], [5], [6], [8], [9] indicate that labeling the road surface condition accurately is a difficult task. We tried to overcome this problem by developing an android application called Road Data Collector that automates data collection and labeling. In the literature, uni-modal accelerometer sensor have been applied. In our work, we are using both gyroscope and accelerometer sensors to derive more accurate road quality prediction. Moreover, while in previous work simple thresholds on various features have been used in anomaly detectors, we use C4.5 decision tree [18] to detect road quality. While most existing solutions require external hardware [4], [3], [10], [2], our solution is a real time android application implemented on a smartphone device. Thus, it greatly reduces the overall cost of the system. We show that our system is able to detect and classify events related to road conditions with an accuracy of 98,6%.

B. Machine Learning Algorithms

We employed in our research three different machines learning algorithms. C4.5 classifier is a simple decision tree for classification. It creates a binary tree to model the classification process. Once the tree is built, it is applied to each tuple in the dataset and leads to assign a class for that tuple [19][20]. While building a tree, C4.5 ignores the missing values. C4.5 allows classification via either decision trees or rules generated from them. Support vector machines (SVM) are supervised learning methods for classification and regression. The SVM classifier is firstly trained, and then unknown samples go through the classifier to be categorized. In the training process, a hyperplane is constructed to classify data into one of the two categories (i.e., category 0 or 1). In the testing process, it predicts whether a new sample falls into one category or the other. Naive Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses Bayes theorem and assumes that all attributes are independent of each other given the class variable.

III. System Design

Our goal is to derive a road quality recognition system that detects, analyzes, identifies and predicts the state of road segments using smartphone sensors. Our system does not depend on any pre-deployed infrastructures and additional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

TABLE I: Comparison with related research in Road Surface Monitoring

Reference	Smartphone sensors	External hardware	Data labeling	Detection methods
[4]	Not used	Accelerometer, GPS	Manually	Threshold/ Machine learning algorithm
[3]	Accelerometer,microphone,GPS	Not used	Not Mentioned	Threshold
[5]	Accelerometer	Not used	Not Mentioned	Threshold
[6]	Accelerometer,GPS	Not used	Video	SVM
[7]	Accelerometer,GPS	Not used	Not Mentioned	Threshold/ Neural Network
[10]	Not used	Kinetic sensor	Not Mentioned	Three-dimensional (3D) reconstruction
[11]	Not used	Stereo camera images	Not Mentioned	Image processing algorithms
[12]	Not used	Camera images	Manually	Three-dimensional (3D) reconstruction
[13]	Not used	Near infrared (NIR) camera	Manually	Video processing methods
[2]	Not used	Ultrasonic sensors,GPS	Not Mentioned	Threshold
[14]	GPS,Power	Not used	Manually	Threshold on relative roughness
[15]	Accelerometer,GPS	Not used	Not Mentioned	Threshold
[16]	Accelerometer,GPS	Not used	Not Mentioned	Threshold the standard deviation of measurements
[17]	Accelerometer	Not used	Not mentioned	Threshold
[8], [9]	Accelerometer,Gyroscope	Not used	Audiovisual	SVM
RoadSense	Accelerometer,Gyroscope,GPS	Not used	Automatic	C4.5 decision tree

hardware. In our system, road conditions could be detected and identified by smartphones according to readings from accelerometer and gyroscope sensors. The life cycle of our system is divided into 2 phases: training and prediction. We will detail in this section these phases.

A. Training phase

In the training phase (Fig. 1), we train the classifier model using machine-learning techniques based on the collected data. During a preprocessing stage, a low pass filter is applied to remove high frequency components, and then we compute magnitude of accelerometer and gyroscope values. In the Feature Extraction stage, effective features are extracted from specific types of road conditions patterns on acceleration and rotation around gravity. Afterwards, the features are selected in the training phase and a classifier model would be generated which can realize fine-grained identification. Finally, the classifier model is generated and saved.

1) Collecting Data from Smartphone Sensors: The Data collection phase is the most important one; since it is responsible for collecting road information. We develop an Android-based App to collect readings from the 3-axis accelerometer and gyroscope sensor. The sensors data of road surface quality were collected using accelerometer and gyroscope sensors built in the Galaxy mobile phone, mounted on the car dashboard as shown in Fig. 2, along the vehicle path. The sampling frequency of the sensors was 50 Hz. Several data collection drives were performed with a varied speed, the road condition label is pre-set before the collection starts. Once the user stops the data acquisition, the application stores the learning data-set as an Attribute-Relation File Format (arff) file. For the work reported in this paper, a drive of about 40 minutes in length (25km) was selected among the drives, as it seemed to be a smooth segment. It was also hard to classify the anomalies into different categories at the same time. So, the classification in this study was not targeted to recognize different road anomalies, but to differentiate potholes from the smooth road. In total, we obtain 2000 samples of data.

2) Preprocessing: Accelerometer data readings usually contained irrelevant data (noises). Therefore, a pre-processing phase should be applied in order to reduce noise and improve the road quality recognition. Due to several factors such as jerks or vibrations, turning, veering, braking, and as well as subtle changes in sensor orientation, a considerable amount

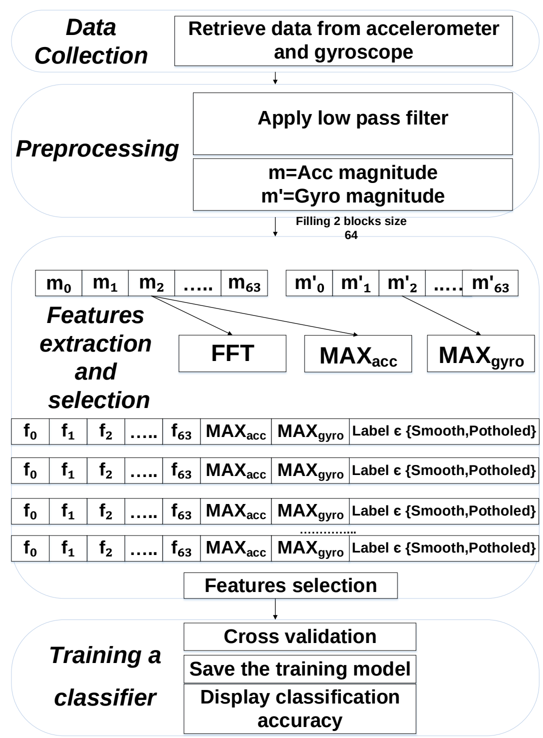

Fig. 1: Training phase.

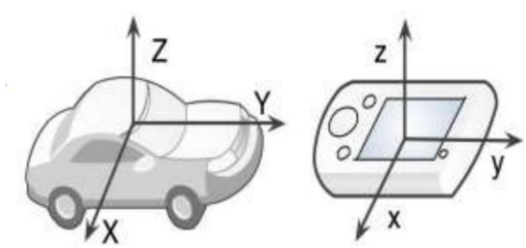

Fig. 2: Phone orientation inside the car.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

of noise is added to these signals. A Low-Pass Filter can be helpful to remove high frequencies signals (noise) in the input signal by applying a suitable threshold to the filter output reading. Once the noise reduction is completed, the filtered accelerometer samples (x, y, z) and the filtered gyroscope samples (x', y', z') are then preprocessed where each sampled smoothed vector was combined into a single magnitude [21] where $(m = \sqrt{x^2 + y^2 + z^2}), (m' = \sqrt{x'^2 + y'^2 + z'^2})$.

3) Feature Extraction: When machine-learning algorithms are processed, representative tuple of features rather than raw data is a more effective input. Thus, it is necessary to extract effective features from road conditions patterns. In order to reduce input data, the raw data obtained from the sensors is first segmented into several windows, and features such as frequencies are extracted from the window of samples. This step of Feature Extraction serves as inputs into the classification algorithms for recognizing roads quality. As mentioned in [22], the window size is a cardinal parameter that influences both computation and power consumption of sensing algorithms. A sliding window of window length 64 is applied for feature extraction stage, which proved helpful in extracting frequency-domain features. In contrast to heuristic features, time and frequency-domain features can describe the information within the time-varying signal. Unlike time domain, the frequency-domain features need another phase of transforming the received data (in time domain) from the previous phases of the pipeline. This stage generates frequency-domain features using very fast and efficient versions of the Fast Fourier Transform. After computing magnitude m and m', the work-flow of this training phase buffers up 64 consecutive magnitudes $(m_0 ... m_{63})$, $(m'_0 ... m'_{63})$ before computing the FFT resulting in a feature vector $(f_0 ... f_{63})$ or a vector of Fourier coefficients. FFT transforms a time series of amplitude over time to magnitude across frequency [23]. Since, the x, y, z accelerometer and x', y', z' gyroscope readings and the magnitude are time domain variables; it is necessary to convert these time-domain data into the frequency domain as they can represent the distribution in a compact manner that the classifier will use to build a model in further phase of this pipeline. While computing the Fourier coefficients, the training phase also stores the maximum (MAX_{acc}) magnitude of the $(m_0 ... m_{63})$, the maximum (MAX_{gyro}) magnitude of the $(m'_0 ... m'_{63})$ and the road supplied label (Smooth, Potholed) using a native android application called Road Data Collector.

At the end of Feature extraction pipeline, a feature vector consists of the following features or attributes: $(f_0 ... f_{63})$, MAX_{acc} magnitude, MAX_{gyro} magnitude, label. The data set is then saved as an arff file. This file format is then analyzed by Waikato Environment for Knowledge Analysis or WEKA [24], a knowledge analysis and machine-learning tool. Therefore, those values can be used as features for training. However, not all of them are equally effective for road conditions detection and identification.

4) Feature Selection: Some features in the processed data-set might contain redundant or irrelevant information that can negatively affect the recognition accuracy and the classification performance. Then, implementing techniques for selecting the most appropriate features is a suggested practice to reduce computations and simplify the learning models. In this work, we applied a Correlation based Feature Selection (CFS) approach [25], taking advantage of the fact that this method is built in WEKA. CFS works under the assumption that features should be highly correlated with the given class but uncorrelated with each other. CFS algorithm reduces the number of features from 66 to 25. Finally, we identify 25 effective features that are able to capture the patterns of different types of road conditions.

5) Training a Classifier Model to Identify road conditions: After feature extracting and selecting, we obtain a tuple of features for each road condition. Then a classifier model is trained based on the tuples for all road conditions through machine learning techniques [26] to identify road conditions. For each road condition, the input into the machine learning algorithm is in the form of (25-dimensional features, label), where the 25-dimensional features are the tuples obtained from the Feature Extraction and the label is the type of the road. Since, the training data is a labeled data set; supervised learning algorithms are used to infer a model from the labeled data. For the selection of the base classification technique, we experimented with a set of widely used algorithms (Decision tree C4.5, Support Vector Machines, and Naive Bayes) and picked the one that yielded the highest classification accuracy. In a word, the classifier trains the inputs and then generates a classifier model which conducts identification to the two types of road conditions.

B. Prediction phase

The prediction phase (Fig. 3), is installed on smartphone, which senses real-time vehicular dynamics to detect and identify road conditions. Our system first senses the readings of accelerometer and gyroscope embedded on smartphone. After getting real-time readings from sensors, the preprocessing is performed on sensors readings. Afterwards, our system extracts features from patterns of the road conditions, then predict the road quality based on the classifier model trained in the training stage and show the road location trace on a geographic map. Finally, a history of all reported road condition is saved.

IV. Performance Evaluation

The overall goal of this evaluation is to determine the accuracy and effectiveness with which our system is able to detect road conditions. In this section, we evaluate our system in two steps:

- Analytical Validation :Evaluate the performance of different classifiers based on several metrics.
- Experimental Validation : Test the feasibility of the whole system in real driving environments.

A. Analytical Validation

The learning data-set with the selected features is passed as input to various classification algorithms in order to select the most appropriate model. To evaluate the effectiveness of different classifiers, cross validation is used as the evaluation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

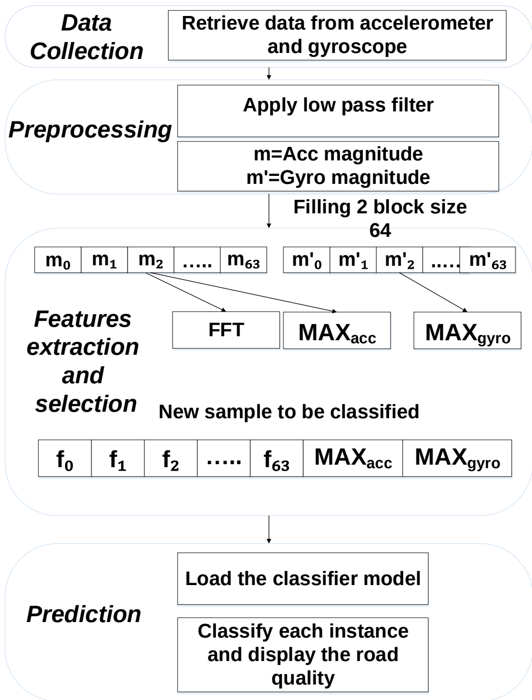

Fig. 3: Prediction phase.

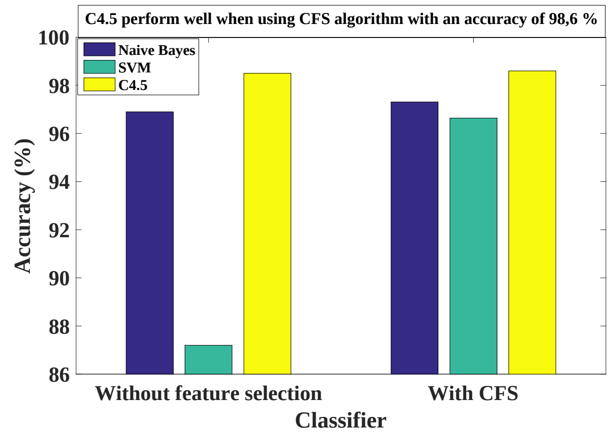

Fig. 4: Accuracy of each algorithm.

method. More specifically, the smartphone sensor dataset is divided into test and training sets.

The goal of cross-validation is to define a subset of data or a validation set to test the model for the training phase, in order to scrutinize the problems such as over-fitting. In order to determine whether a classifier is better than another, a 10 fold cross validation were performed, where 1/10 of both data were used only for testing purpose. We have used several measures in order to evaluate the performance of classifiers used.

1) Classification accuracy: The accuracy can explain the overall classification performance for all the activity classes as the follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Fig. 4 shows the classification accuracies of different machine learning algorithms with and without feature selection. As is evident from Fig. 6, C4.5 is the most accurate classifier compared to SVM and Naive Bayes with the average accuracy of 98.50%, CFS algorithm increases the average accuracy by 0.1%. We can see that C4.5 perform well when using CFS algorithm; this high accuracy was achieved by fusing data from accelerometer and gyroscope sensors to reduce the number of false positives. Support Vector Machine performs poorly in accuracy (95.25%) and Naive Bayesian Classifier is a close competitor in terms of accuracy (96.90%).

2) TP Rate and FP Rate: True positive rate is the percentage of positive cases correctly classified as belonging to the positive class.

$$TPRate = \frac{TP}{(TP + FN)} \tag{2}$$

False positive rate is the percentage of negative cases misclassified as belonging to the positive class.

$$FPRate = \frac{FP}{(FP + TN)} \tag{3}$$

From the above results of Table II, in true positive element rate, C4.5 > naive bayes > SVM ; in false positive element rate, naive bayes > SVM > C4.5.

TABLE II: Performance comparison of Three Algorithms using different metrics

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
C4.5	0.985	0.064	0.985	0.985	0.985	0.987
SVM	0.953	0.265	0.951	0.953	0.950	0.844
Naives Bayes	0.969	0.049	0.972	0.969	0.970	0.976

3) Precision, Recall and F-measure: The precision is another metric which can explain about the ratio of the correctly classified positive instances to the total number of instances classified as positive and is mathematically calculated as follows:

$$Precision = \frac{TP}{(TP + FP)} \tag{4}$$

The recall also known as the true positive rate or sensitivity, is a measure of how good is the classifier to correctly predict actual positives samples

$$Recall = \frac{TP}{(TP + FN)} \tag{5}$$

the F-measure is a measure of a test's accuracy. The F-measure can be interpreted as a weighted average of the precision and recall

$$F - measure = \frac{2 \times precision \times recall}{(Precision + Recall)} \tag{6}$$

Both precision and recall are therefore based on an understanding and measure of relevance. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

In precision C4.5 >SVM >naive bayes; in recall ratio, C4.5 >naive bayes >SVM (Table II). Since F-measure is the harmonic mean of precision and recall, hence to know which classifier is the best in terms of precision and recall, we can calculate the F-measure value (Table II). The average F-measure value of C4.5 is the biggest among the three, that is 0.985. Nave Bayes has average F-measure of 0.969 and SVM of 0.950. Therefore we can say that C4.5 is the best in terms of precision and recall .

4) ROC Curve: A receiver operating characteristics (ROC) analysis is another metric used in machine learning to evaluate and compare classifier performance through the AUC (Area under ROC curve). ROC performance of a classifier is usually represented by a value, which is the area under the ROC curve (AUC) [27]. AUC has known as a statistical measure for evaluating classification, ranking models and selecting the best classification method. The value of AUC is between 0 and 1. From Table II, The AUC of C4.5 is 0.987, followed by Nave Bayes 0.976 and SVM 0.844.

5) Confusion Matrix: The most straightforward way to evaluate the performance of classifiers is based on the confusion matrix analysis. A confusion matrix contains information about actual and predicted classifications done by a classification system.

The confusion matrix of three algorithms was shown in Table III. In the table, the value of diagonal lines should be mainly focused and the greater it is, the more the examples are correctly classified. From Table III (a), it could be found that there were 1938 groups of samples. The number of correctly classified examples of SVM algorithm was 1846, 92 wrong examples; the number of correctly classified examples of naive Bayesian classifier was 1878, 60 wrong examples(Table III (b)).

Table III (c) shows the confusion matrix for the classification results obtained from C4.5 Classifier. From this table, The number of correctly classified examples was 1911, 27 wrong examples. It can be seen that the first row has 1690 instances corresponding to class smooth, where 1680 are correctly classified and the other 10 were misclassified as potholed.

These false negatives can be explained by the presence of rough road considered as smooth during the learning phase. However, the second row has 248 instances corresponding to class potholed, suggesting that 17 instances were misclassified as smooth. These false positives are due to the absence of some kind of pothole during the learning phase.

The experiment we carried out reveals that C4.5 Decision Tree Classifier outperforms SVM and naive Bayesian. It is the best in all performance parameters. So we decided that C4.5 should be the choice for the optimal classifier.

Results of analyses are good enough to motivate us to use C4.5 in order to discover patterns of road conditions and predicting the quality of an unknown road.

Then, Decision Tree C4.5 algorithms are a pruned decision tree that works well for recognizing tasks, well-known for their low computations and reasonable accuracies. The main reason C4.5 decision tree was chosen to serve, as a model for classification, is that it produces simpler rules and removes insignificant parameters before it begins a process of tree induction. Moreover, Decision tree can handle datasets that may have errors, or missing values. Decision Tree is believed to be a better solution to be integrated into smartphone applications. As a decision tree model, it is easy to compute when implemented as a set of IF-THEN rules.

TABLE III: Confusion Matrix for all classifiers

(a) SVM

Class Label	Smooth	Potholed
Smooth	1673	17
Potholed	75	173

(b) Naive Bayes

Class Label	Smooth	Potholed
Smooth	1643	47
Potholed	13	235

(c) C4.5

Class Label	Smooth	Potholed
Smooth	1680	10
Potholed	17	231

6) Impact of smartphone sensors: From Fig. 8, the overall accuracy of C4.5 classifier is 96,74% using only accelerometer features. Besides, Fig. 8, shows that gyroscope features are more determinant to identify road conditions. As exhibited by the results, the gyroscope features is efficient enough to recognize road conditions, which are also simple and practical to be extracted from smartphone sensor readings.

However, when we combine the accelerometer with the gyroscope features, the framework accuracy was up to 98,6%. So we can say that gyroscope sensor is important to confirm the detection results in addition to the accelerometer sensor.

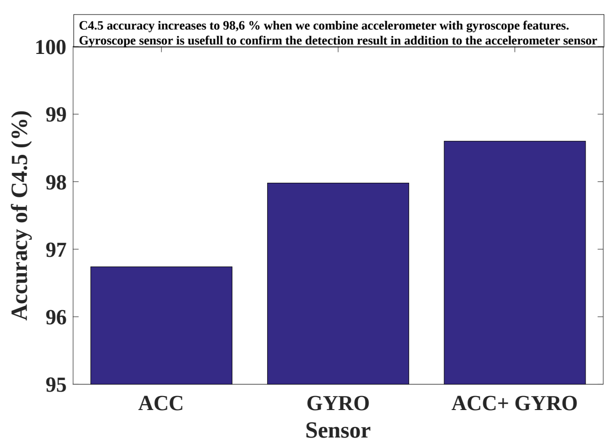

Fig. 5: Impact of smartphone sensors.

B. Experimental Validation

In this section, we first present the prototype of our application, then we evaluate and test the performance of our system in real driving environments

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2702739, IEEE Sensors Journal

1) Application Oviewer: Once the classifier was chosen, the C4.5 model is created outside the application that is using Weka data mining tool. The model is converted into rules and incorporated into the application as an activity. Building this application involved some work to display information in real time on map and to implement the database, which display all recorded road conditions. We develop the road condition monitoring application RoadSense and test it on Android Smartphone Samsung galaxy alpha. The smartphone provides an accelerometer sensor and a gyroscope sensor. We have used Android platform because it is open source and has a fairly versatile API (Application Programmer Interface). When the application is started, the application calls Android's location listener, which takes the user location and route in a moving vehicle from the phone's inbuilt GPS. After the system is started, the monitoring daemon keeps running in background as a Service in Android, collecting and recording the readings of sensors. These readings are processed, and then passed to the WEKA Classifier activity to predict the road conditions status. When road condition is detected, the map notification component works to send the road condition directly on map and show the route traveled by the driver in real time. Once the road condition is detected, it needs to be stored somewhere. The Android API supports development of a SQLite database. All event detected are stored in the database. The initial design included a list view that read the entries in the database and displayed them using the List View format. Users could remove events from the list, which would delete them from the table. Using this application, the driver can get knowledge of the road before traveling. We expect to visualize a road quality map of a selected region. Providing constructive feedback to the driver and local authorities is crucial. Road Manager can benefit from this system to evaluate the state of their road network and make a check on the road construction project, whether they meet required quality or not.

2) Experimental environment and Snapshots: In this section, we describe our experiments made to test the performance of the whole system. We install the android application RoadSense on smartphone (Samsung galaxy alpha). We recruited a volunteer participant for our study, and deployed the smartphone on the vehicle dashboard. The phone operate in a completely autonomous manner, requiring no user intervention from the participant. The application is run by the driver in real driving environments. The drivers is asked to drive just as how he normally do. The entire experiment lasted for about 3 weeks. We choose different road circumstance like smooth and potholed. The RoadSense application detect in real time the road quality, and show the road location trace on a geographic map with some useful information including the number of potholes, the percentage of smooth type on the traveled road, the speed of the vehicle, and the distance traveled. The snapshots attached are the output result of our application. Fig.6 shows that when traveling a potholed road, the application successfully predict that the type of this road is potholed. An illustration overlaying Google map of the road traveled is shown in Fig. 7, indicating that the road segment is smooth. The application also keeps the record of all reported road condition. Our various experiment results show that our application can effectively and efficiently predict the road type.

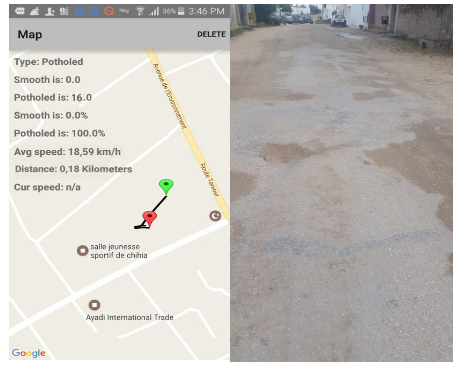

Fig. 6: Potholed road.

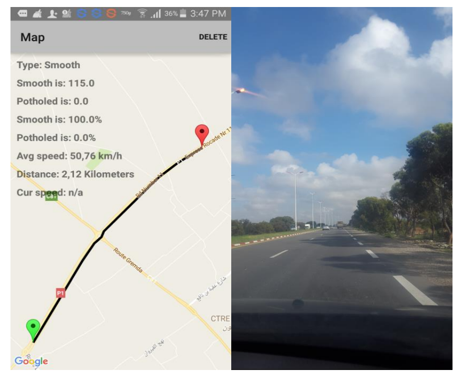

Fig. 7: Smooth road.

V. Conclusion

In this paper, we studied a machine-learning algorithm for prediction of road quality. It uses an accelerometer and gyroscope sensor for collection of data and GPS for plotting the road location trace in Google map. We have tested three classification algorithms: decision tree C4.5, SVM and Naive Bayes. Our experimentation shows the superiority of C4.5 in term of detection accuracy (98,6%). Our best results is obtained thanks to a grouping of two sensors; accelerometer and gyroscope. The smartphone-based method is very useful because it removes the need to deploying special sensors in vehicle. It has the advantage of high scalability as smartphone users increases day by day. Thus, we have developed a smartphone application RoadSense. The RoadSense application is an attempt to provide its users with better knowledge about the routes of their transportation. With further work in this field, it is possible for this project to play a proactive part in improving road conditions in developing countries. To this end, our system can be used to create a personal road type warning system that maintains a historical record of road conditions. As a future work, we aim to improve the road type detection algorithm through detecting other road anomalies and trying other machine learning classifiers.

ACKNOWLEDGMENT

The authors would like to thank the Center of Excellence and the Robotics and Internet-of-Things Unit at Prince Sultan University, Saudi Arabia for their support of this work.

REFERENCES

[1] W. H. Organization. (2015) Global status report on road safety. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2015/en.pdf/

[2] R. Madli, S. Hebbar, P. Pattar, and V. Golla, "Automatic detection and notification of potholes and humps on roads to aid drivers," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4313–4318, 2015.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst*, Raleigh, NC, USA, 2011, pp. 323–336.

[4] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst.Appl. Services*, New York, NY, USA, 2008, pp. 29–39.

[5] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst.Workshops*, Barcelona, 2011, pp. 1–6.

[6] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppnen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, and T. Ristaniemi, "Distributed road surface condition monitoring using mobile phones," in *Proc. Int. Conf. Ubiquitous Intell. Comput*, Berlin, Heidelberg, 2011, pp. 64–78.

[7] A. Kulkarni, N. Mhalgi, D. Sagar Gurnani, and N. Giri, "Pothole detection system using machine learning on android," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 7, pp. 360–364, July 2014.

[8] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Proc. Int. Workshop Modeling Social Media*, Cham, 2014, pp. 128–146.

[9] F. Seraj, K. Zhang, O. Turkes, N. Meratnia, and P. J. Havinga, "A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior," in *in Proc. ACM Int. Conf. Pervasive Ubiquitous Comput*, Osaka, Japan, 2015, pp. 1169–1177.

[10] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman, "Metrology and visualization of potholes using the microsoft kinect sensor," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst*, Netherlands, 2013, pp. 1284–1291.

[11] Z. Zhang, X. Ai, C. Chan, and N. Dahnoun, "An efficient algorithm for pothole detection using stereo vision," in *Proc. IEEE Int. Conf.Acoust., Speech Signal Process*, Florence, Italy, 2014, pp. 564–568.

[12] D. Rajamohan, B. Gannu, and K. S. Rajan, "Maargha: a prototype system for road condition and surface type estimation by fusing multi-sensor data," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1225–1245, 2015.

[13] P. Jonsson, J. Casselgren, and B. Thörnberg, "Road surface status classification using spectral analysis of nir camera images," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1641–1656, 2015.

[14] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 4, p. 55, 2015.

[15] H. Sharma, S. Naik, A. Jain, R. K. Raman, R. K. Reddy, and R. B. Shet, "S-road assist: Road surface conditions and driving behavior analysis using smartphones," in *Proc. 2015 Int. Conf.Connected Vehicles Expo*, Shenzhen, China, Oct 2015, pp. 291–296.

[16] K. Kulkarni, K. V. Prashant, S. Nasikkar, T. Ahuja, and N. Mhetre, "Predicting road anomalies using sensors in smartphones," *Imperial journal of interdisciplinary research*, vol. 2, no. 6, 2016.

[17] Y. Li, F. Xue, L. Feng, and Z. Qu, "A driving behavior detection system based on a smartphone's built-in sensor," *International Journal of Communication Systems*, vol. 30, no. 8, 2016.

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[19] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2002.

[20] A. K. Sharma and S. Sahni, "A comparative study of classification algorithms for spam email data analysis," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1890–1895, 2011.

[21] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc. 7th ACM MobiSys*, Krakow, Poland, 2009, pp. 179–192.

[22] S. A. Hoseini Tabatabaei, A. Gluhak, and R. Tafazolli, "A survey on smartphone-based systems for opportunistic user context recognition," *ACM Comput. Surv(CSUR)*, vol. 45, no. 3, pp. 1–51, jul 2013.

[23] A. T. Campbell. (2013) Smartphone programming. [Online]. Available: http://www.cs.dartmouth.edu/campbell/cs65/myruns/myrunsmanual.html

[24] Data mining software in java. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

[25] M. A. Hall, *Correlation-based feature selection for machine learning*, Hamilton, New Zealand, 1999.

[26] P. Harrington, *Machine learning in action*. Manning Greenwich, CT, 2012.

[27] R. Entezari-Maleki, A. Rezaei, and B. Minaei-Bidgoli, "Comparison of classification methods based on the type of attributes and sample size," *Journal of Convergence Information Technology*, vol. 4, no. 3, pp. 94–102, 2009.

Azza Allouch was born in Sfax, Tunisia, in 1991. She received her master degree from the National School of Electronics and Telecommunication of Sfax in 2016. Her master thesis versed on machine learning Algorithm for road condition monitoring using motion sensors on android platform. She is a member of The COINS (Cooperative Intelligent Networked Systems) research group.

Anis Koubâa Anis Koubâa received his B.Sc. in Telecommunications Engineering from Higher School of Telecommunications (Tunisia), and M.Sc. degrees in Computer Science from University Henri Poincaré (France), in 2000 and 2001, respectively, and the Ph.D. degree in Computer Science from the National Polytechnic Institute of Lorraine (France), in 2004. He was a faculty member at Al-Imam University from 2006 to 2012. Currently, he is a Professor in the Prince Sultan University, and Research Associate in CISTER Research Unit, ISEP- IPP, Portugal, and a Consultant in Gaitech Robotics (China). He becomes a Senior Fellow of the Higher Education Academy (SFHEA) in 2015. He has published over 150 refereed journal and conference papers. His research interest covers mobile robots, cloud robotics, robotics software engineering, Internet-of-Things, cloud computing and wireless sensor networks. Dr. Anis received the rector teaching award in Prince Sultan University in 2016, best research award from Al-Imam University in 2010, and the best paper award of the 19th Euromicro Conference in Real- Time Systems (ECRTS) in 2007. He is the head of the ACM Chapter in Prince Sultan University. His H-Index is 30.

Tarek Abbes is an assistant professor at the National School of Electronics and Telecommunication of Sfax, Tunisia. He received his Ph.D diploma in computer science from Henri Poincaré University Nancy, France in 2004. The topic of his thesis was network traffic classification and multipattern matching for intrusion detection. He obtained the DEA Diploma in Artificial Intelligence from Henri Poincaré University of Nancy France in 2001 and a Telecommunication engineering Diploma from High school of communication of Tunis in 2000. His research is concerned with formal verification of network equipments configuration, network intrusion detection and feature interaction in Telecommunication systems.

Adel Ammar received the Engineer degree from the École des Mines de Nancy, Nancy, France, in 2003, and the Ph.D. in Applied Data Processing, from the University of Paul Sabatier, Toulouse, France, in 2008. He is currently working as an assistant professor at the College of Computer Science and Information Systems, at Al Imam Mohammad Ibn Saud Islamic University (IMSIU). His research interests include machine learning, pattern recognition, statistical modeling, and intrusion detection.