Efficient pothole detection using smartphone sensors

Kshitij Pawar[1],[*], *Siddhi* Jagtap[2],[**], and *Smita* Bhoir[3],[***]

[1]Department of Computer Engineering, Ramrao Adik Institute of Technology, India
[2]Department of Computer Engineering, Ramrao Adik Institute of Technology, India
[3]Department of Computer Engineering, Ramrao Adik Institute of Technology, India

Abstract. Road safety remains a casualty in India, with potholes wrecking asphalt pavements by the dozens. A study in 2017 recorded that potholes caused the budget for road safety to increase by a whopping 100.4 per cent, and even doubled the death toll from that of the year prior. To address this situation, an effective solution is required that ensures the drivers' safety and can prove beneficial for long term measures. This can be established by employing an apt pothole detection system which is simple yet functional. In this paper, the method for such a system is described which uses accelerometer and gyroscope, both built in the modern day smartphones, to sense potholes. Pothole induced vibrations can be measured on the axis reading, making them distinguishable. Our proposed Neural Network model is trained and evaluated on the data acquired from the sensors and classifies the potholes from the non-potholes. The neural network gives a classification accuracy of 94.78 per cent. It also presents a solid precision-recall trade-off with 0.71 precision and 0.81 recall, considerably high for a problem with class imbalance. The results indicate that the method is suitable for creating an accurate and sensitive supervised model for pothole detection.

1 Introduction

Roads with potholes have become nearly universal in India [1]. In fact, it has become second nature for drivers to swerve through lanes to avoid potholes, which causes nearby vehicles and other road users to panic as well. The sizes of these undesirable potholes may vary with location, but the damaged caused is just as severe. Potholes have the capability of causing accidents that may be serious or even fatal. The sudden, jarring hits against potholes affect the efficiency of the vehicles and impact the tyres, wheels, and drivers alike.

The government has mandated Road Safety Audit to provide safer transportation to road users [2]. Concerned directors inspect and take suitable short term and long term measures to ensure that the roads are pothole free. But these manual detection measures are not enough; establishment of a real-time pothole detection system is crucial for swift and effective pothole recognition.

With essentially this in mind, the pothole detection model is trained that uses the data acquired from the smartphone sensors, namely the accelerometer and the gyroscope, to classify the real instances of potholes. The accelerometer measures vibrations or inclination in the three directions, X, Y, and Z axes, and can help characterize the pothole on the basis of the magnitude of change in its reading. The gyroscope adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist. This vibrations based approach gives low response time, processing time and maintenance cost of the device.

Using these readings, our neural network model is trained and tested to accurately classify the potholes. For the purpose of the paper and the research, work has been done on a sufficiently large pre-existing dataset instead of gathering and modifying a dataset ourselves. The prime goal was to create an improved model from the ones already designed, that shows higher performance measures for categorization. Successful implementation has been done to increase the accuracy as well as sensitivity (recall) of the prototype, making the model highly suitable for a real-time pothole recognition system.

The rest of the paper is organized as follows. Section 2 gives a review of the related work. Section 3 explores the proposed methodology, and Section 4 evaluates and illustrates model metrics such as Accuracy, Precision, and Recall. Finally the paper is concluded in Section 5.

2 Literature Survey

Several pothole detection methods have been put forward, a few of which make use of the data obtained from the inbuilt smartphone accelerometer to discern potholes.

2.1 Existing Work

The undermentioned research papers related to work have been surveyed and inspected as follows.

---
[*]e-mail: kshitijvijay271199@gmail.com
[**]e-mail: siddhijagtap63@gmail.com
[***]e-mail: smitapatilbe@gmail.com

2.1.1 Real time pothole detection using Android smartphones with accelerometers [3]

Mednis et al proposed four pothole detection approaches in their paper to process the accelerometer data acquired using different Android OS based smart-phones. The performance analysis in context of different road irregularity classes showed true positive rates as high as 90%. The best results, displayed by the Z-DIFF approach, are however reliant on frequency and timing.

2.1.2 Pothole Detection System using Machine Learning on Android [4]

Kulkarni et al, in their paper describe a system using accelerometer sensor for detection and GPS for plotting. The algorithm assumes some threshold values on x-axis and z-axis, justified using a neural network technique which translates to an accuracy of 90%-95%, but bares no detail on other performance metrics.

2.1.3 A Real-Time Pothole Detection Approach for Intelligent Transportation System [5]

Wang et al overcame the high false-positives predicament faced by single threshold approaches by combining and improving the Z-THRESH and G-ZERO approaches as depicted in the fifth approach of their paper. Although the algorithm gives an accuracy of 100%, the limitation of this study is the sample size.

2.1.4 Intelligent Pothole Detection and Road Condition Assessment [6]

Bhatt et al, in their paper provide a solution to detect road conditions by capturing data on a car's movement from accelerometer and gyroscope sensors in the phone. They trained several classifiers on aggregate metrics from which the best performing classifier, support vector machine (SVM), achieved the highest test accuracy of 92.9%. Adjusting the classification threshold improved the precision-recall trade off to give a precision of 0.74 and a recall of 0.42.

2.1.5 Pothole Detection using Machine Learning [7]

Song et al, describe the process for their approach in their paper, which includes acquiring road status information using gyroscope and accelerate sensors, and data pre-processing for a restrained Inception V3 and Transfer Learning. The model showed 100% classification rate but the consideration of only five real instances for each class, collected 20 times for each instance, might suggest overfitting.

2.2 Analysis of Literature Survey

An analysis of various papers on pothole detection methods by considering the type of equipment and methods employed has been done and the research gaps are identified.

After analysing Table 1, it has been observed that while accuracy is high, the recall score is low and potholes are relatively misclassified as compared to non-potholes.

Table 1. Analysis of Literature Survey

Authors	Year	Equipment & Methods	Research Gaps Identified
Mednis et al [3]	2011	4 Android OS based smart-phones	Best results are reliant on frequency and timing
Kulkarni et al [4]	2014	Threshold values on x-axis and z-axis have been taken for the neutral network	Other performance metrics not mentioned
Wang et al [5]	2015	Combining the Z-THRESH and G-ZERO approaches	Limitation of the study is the small sample size
Bhatt et al [6]	2017	SVM classifier, improved using a threshold	Recall of 0.42
Song et al [7]	2018	Data pre-processing done for a restrained Inception V3 and Transfer Learning	Considered only five real instances, collected 20 times for each instance, suggesting overfitting

3 Proposed Work

The proposed work focuses on building an efficient pothole detection system, that is a classifier with increased accuracy, while also ensuring that the process is not too complex and time-consuming. Figure 1 presents the proposed architecture.
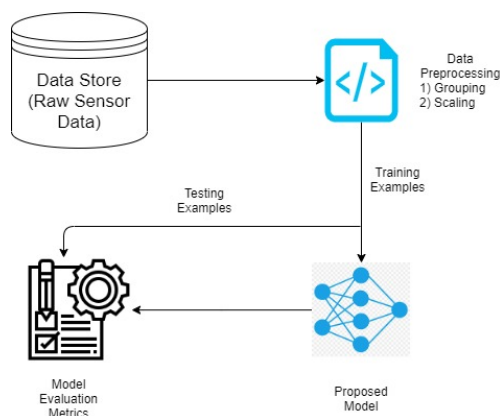

Figure 1. Proposed Architecture

The architecture has four major stages: data collection, pre-processing of data, building the model, and evaluating the classifier. The raw data was collected from the smartphone's in-built sensors: accelerometer and gyroscope. After the data was cleaned and transformed to

improve its quality, it was fed to the proposed model, that is the neural network. Finally, model was evaluated on a number of evaluation metrics to make out its performance.

3.1 Data Collection & Pre-processing

The dataset is primarily the vibration based sensor readings collected from a smartphone's in-built sensors, that is, the accelerometer and gyroscope (Figure 2), as the phone was mounted on the windshield of the vehicle to detect said vibrations.
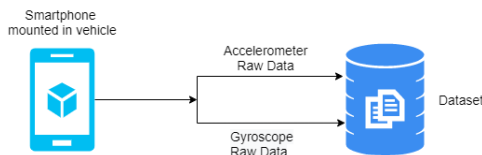

Figure 2. Raw Data Collection

The corpus of data is the result of five trips over a selected road with sufficient instances of potholes to make inferences. It consists of a total twenty-four input features and the target feature which labels the instance as pothole or not. The inputs are essentially the minimum, maximum, mean and standard deviation values of the x, y and z directions of the accelerometer as well as the gyroscope.

Figure 3 and and Figure 4 show the plots [6] for the normalised accelerometer values of the x, y and z directions for both the Good Road Points (the non-pothole occurrences) as well as the Bad Road Points (the pothole occurrences).
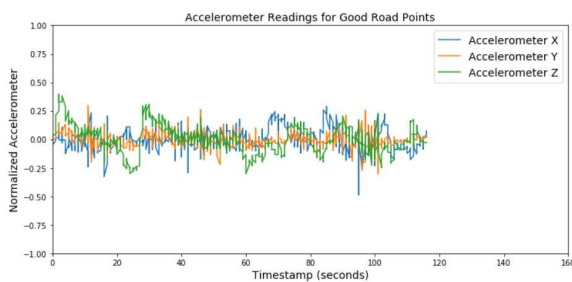

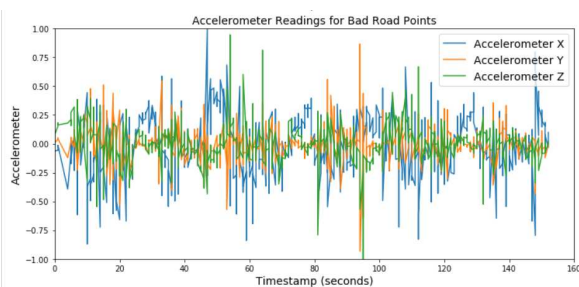**Figure 3.** Accelerometer Readings for Good Road Points

Figure 4. Accelerometer Readings for Bad Road Points

It can be clearly seen that instances of potholes have high variations compared to that of the non-potholes and this disparity benefits the model greatly.

The normalized gyroscope readings also indicate this variance and are thus instrumental features together with the accelerometer readings.

3.2 Feature Engineering

3.2.1 Grouping

Each instance of the dataset was originally taken every fifth of a second, which resulted in turbulent fluctuations. Hence, in an attempt to counter this the instances were grouped by intervals of two seconds each, in order to make better sense of the data points.

3.2.2 Scaling

The features did not have a certain range and differed from each other, hence scaling was applied to each. Standardization (or z-score normalization) scales the values by the mean while taking into account standard deviation, consequently reducing the effect of the outliers in the features.

$$x = (x - mean(x))/std(x)$$

This aggregation provides a finer signal for pothole classification as can be seen in Figure 5 and Figure 6. The plots show the accelerometer standard deviations for the Good Road and Bad Road Intervals. The sharp spikes for the pothole occurrences can now be better perceived.
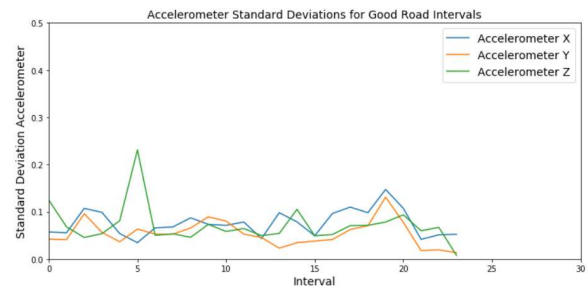

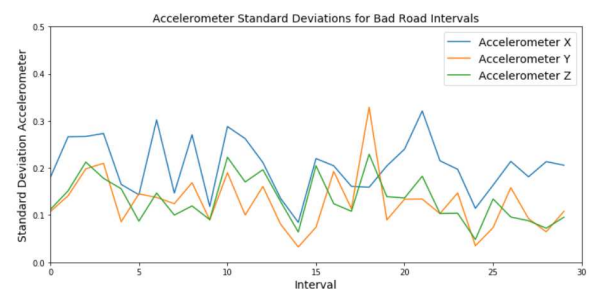**Figure 5.** Accelerometer Standard Deviations of Good Road Intervals

Figure 6. Accelerometer Standard Deviations of Bad Road Intervals

3.3 Proposed Model

The proposed model is a Neural Network, comprising of two hidden layers and one output layer. Each hidden layer has a dense layer with ReLU activation function, allowing faster backpropagation, and a dropout layer, to prevent overfitting, by randomly setting a fraction rate of input units to zero at each update.

The output layer, on the other hand, employs the sigmoid activation function in order to attain the probabilities to define the class of each input (pothole or not).

Further, the model uses the binary cross entropy loss function to compile, and trains over 140 epochs with a batch size of three.

4 Results and Discussions

A total of seven tests were implemented for refining the model by changing a number of parameters and measures. The different test cases along with the various hyperparameters used to tune the respective neural network are given in Table 2. The test cases are nothing but experimental trials of neural networks, each leading to better or worse results, in an attempt to find the model parameters suited best for the problem. The SMOTE case was also implemented as a solution to counter the class imbalance problem.

The pothole detection problem, like many problems, is one of severe class imbalance. The instances of potholes are very few compared to the instances of non-potholes. Synthetic Minority Oversampling Technique, or SMOTE, is an algorithm used to tackle the underlying class imbalance of the dataset by simply oversampling the instances of the minority class. The new examples are then randomly changed to create synthetic but reliable records, reducing the class disparity.

Table 2. Test Models

Test Case	Dense Layers	Dropout Layers	Epochs	Batch Size
1	1	1	700	9
2	4	1	700	12
3	3	2	100	1
4	3	2	100	2
5	3	2	100	5
6	3	2	300	3
Proposed Model	**3**	**2**	**140**	**3**
SMOTE	3	2	40	32

Ultimately, among the eight cases implemented, the seventh case which is the proposed model gave the best results, with sufficiently high classification metrics as required for the problem.

4.1 Results

The results obtained from the different test cases are shown in Table 3. The proposed model gives a high accuracy of 0.9478 and a prominent precision-recall trade-off, with 0.7105 precision and 0.8181 recall.

Table 3. Results

Case	Accuracy	Precision	Recall	F1 Score
Baseline	**89.877**	**1**	**0**	**0**
1	0.9141	0.6315	0.3636	0.4615
2	0.9018	0.5217	0.3636	0.4285
3	0.9171	0.6071	0.5151	0.5573
4	0.9171	0.6667	0.3636	0.4705
5	0.9141	0.5806	0.5454	0.5625
6	0.9018	0.5172	0.4545	0.4838
Proposed Model	**0.9478**	**0.7105**	**0.8181**	**0.7605**
Expected	**0.9263**	**0.7368**	**0.4242**	**0.5384**
SMOTE	0.8957	0.4864	0.5454	0.5142

The comparison between model evaluation metrics of our proposed model with that of the Bhatt et al support vector machine (SVM) model are illustrated in Figure 7. It is observed that our proposed model achieved significantly better recall, implying that the model is able to recognize the actual potholes adequately well.
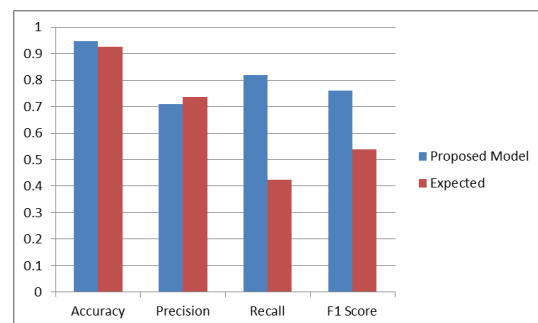

Figure 7. Comparison of expected and proposed model

For this specific problem with class imbalance it is observed that neural networks are better equipped for the job and give better results due to scarcity of dataset and large number of input features i.e., 24. Several key parameters about the structure of the input data are required to be set in case of SVM, which leads to it being a stiff and inflexible architecture. However, in case of neural networks loss is automatically minimized using Gradient Descent without any special configurations. SVM can also suffer during

scaling i.e., when the size of the dataset increases on being deployed in production, whereas neural networks do not suffer from the increased size of dataset and scale quite efficiently too.

4.2 Analysis

The seventh test gave the best results with high train and test accuracy over passing epochs, as well as low output loss.

Figure 8 and Figure 9 explore the values of accuracy and loss over epochs for the train and test sets respectively.
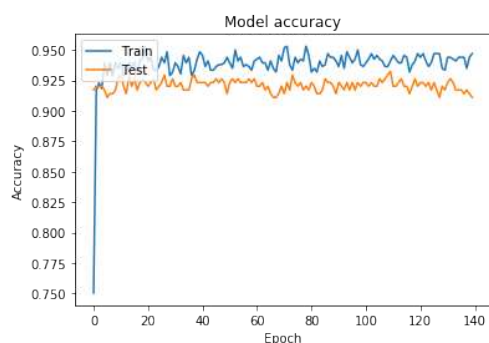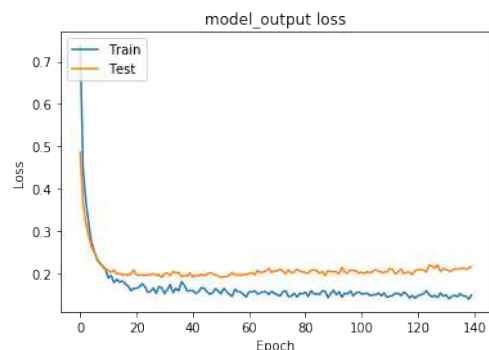

Figure 8. Epoch vs. Accuracy

Figure 9. Epoch vs. Loss

4.2.1 Accuracy Metrics

The model shows an accuracy of 0.9478 which is higher than the baseline considered. Not only that, the precision-recall trade-off too is quite high with a precision of 0.71 and a recall of 0.81, accounting to a F1-score of 0.76.

Precision is the ratio of correctly predicted pothole observations to the total observations predicted as potholes. Recall (Sensitivity) is the ratio of correctly predicted potholes to all the observations that are actually potholes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 is more useful than accuracy, for a problem of uneven class distribution. Hence a high F1 score of 0.76 shows the superior skills of the classifier.

4.2.2 ROC-AUC Curve

Another evaluation metric considerd for the class imbalance is the ROC-AUC Curve.

ROC (Receiver Operating Characteristics) is a probability curve and AUC (Area Under the Curve) represents degree or measure of separability. The x-axis of the plot indicates the False Positive Rate and the y-axis indicates the True Positive Rate. It tells how much the model is capable of distinguishing between the binary classes. Higher the AUC, better the model is at predicting the positive classes.

The neural network gives an AUC of 0.972 for the training data, as shown in Figure 10. This indicates that there is a 97.2% chance that the model will be able to distinguish between the pothole class and the non-pothole class.
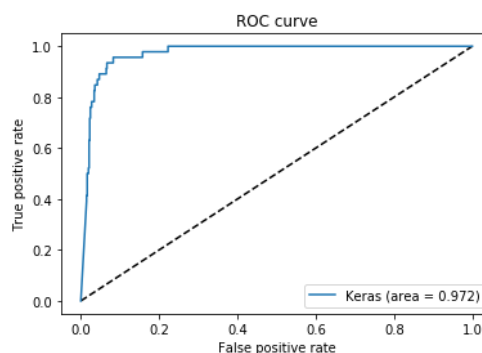

Figure 10. ROC Curve

5 Conclusion and Future Work

By crowd sourcing, one can map locations of potholes and provide such information to authorities and develop more detailed maps for safe travel.

In this paper, a coherent approach is proposed to recognize and detect a pothole on the road. The data was acquired from the smartphone sensors, making the approach cost-effective. A simple yet functional Neural Network for the class imbalance classification problem is utilized, improving the accuracy measures seen previously. The proposed model gives a sufficiently high accuracy of 0.94 and a significant recall of 0.81, making the model suitable for real-time systems.

Further improvement in model accuracy can be increased by collecting more sensor data over different sized potholes to better represent the variety in depths of potholes. Scope can be expanded to classify different types of potholes as well.

References

[1] A. Sherrif, "How Difficult Is It to Keep Our Roads Pothole Free" (2018)

[2] V. Singh, "The pothole problem in India" (2018)

[3] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, L. Selavo, "Real time pothole detection using Android smartphones with accelerometers", Int. Conf. DCOSS, 1-6 (2011)

[4] A. Kulkarni, N. Mhalgi, S. Gurnani, N. Giri, "Pothole Detection System using Machine Learning on Android", Int. Jou. Em. Tech. Adv. Eng. **14**, 7 (2014)

[5] H. Wang, C. Chen, D. Cheng, C. Lin, C. Lo, "A Real-Time Pothole Detection Approach for Intelligent Transportation System", Math. Prob. in Eng. **2015**, 1-7 (2015)

[6] U. Bhatt, S. Mani, E. Xi, J.Z. Kolter, "Intelligent Pothole Detection and Road Condition Assessment", ArXiv **abs/1710.02595** (2017)

[7] H. Song, K. Baek, Y. Byun, "Pothole Detection using Machine Learning", Adv. Sci. Tech. Let. **150**, 151-155 (2018)