iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36134

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Railway Track Monitoring System

Charu Latkar[1], Soubhig Shome[2], Rishan Rai[3], Prof. Ravishankar Holla[4]
[1, 2, 3]*Department of Electronics and Communication Engineering, R V College of Engineering, Bengaluru, India*
[4]*Assistant Professor, Department of Electronics and Communication Engineering, R V College of Engineering, Bengaluru, India*

Abstract: ***For the protection and proximity of railway networks it is substantial to Promptly detect and identify faults in the railway tracks. In this paper, railway track fault diagnosis is approximated from the vertical and lateral acceleration using a MPU6050. MPU6050 consisting of three sensors namely gyroscope, magnetometer and accelerometer are used to distinguish line and level as symetricities in a railway track.***
A GSM module is used to notify the location of faults on tracks. Arduino Microcontroller is interfaced using Arduino UNO IDE. The results show that the condition of railway track irregularity and railway track striation can be approximated constructively. The processed data is uploaded to the open source cloud provider thingspeak.com.
The use of various Machine Learning Algorithms are proposed to accomplish the above tasks based on the commonly available measured signals. By considering the signals from multiple railway tracks in a geographic location, faults are diagnosed from their spatial and temporal dependencies.
The irregularities in the railway tracks are detected using the Inertial Monitoring Unit, providing the necessary data about future deformities using Machine Learning.
Using Python 3.0, a generative model is developed to show that the AdaBoost network can learn these dependencies directly from the data. Seven different classification algorithms used for this project are Logistic regression,Naive Bayes Algorithm,Support Vector Machine, Ensemble Machine (Average) learning Algorithm, XGBoost Classifier, Extreme Machine Learning and AdaBoost Classifier.
Among the above 7 classification algorithms, AdaBoost Learning has given the highest accuracy,i.e of 93.93 %. The AdaBoost Machine Learning Model is used throughout the model.
Keywords: MPU6050, Machine learning algorithms, React Native, Arduino Uno, RTMS

I. INTRODUCTION

With the rise and development in technology, railway transport has gained an increasingly significant role in passenger and freight transport. Hence, Persistent tracking and inspection of railway tracks is essential for trouble-free and intactness of railway track system. The detection of faulty behavior and the determination of the cause(s) of that behavior are the main objectives of the fault diagnosis task.
It is not possible to install additional monitoring devices on each track circuit as the railway track circuit is a large network. Inertial Monitoring Unit (Micro Processing Unit (MPU)6050) can be used to detect increased oscillations that are observed in a train body due to Horizontal Fissure and Square of Angular Breaks.
With the development of software techniques and the decreasing costs of computer and electronic hardware, the works in these fields have produced efficient results. To permit operational security of a railway circuit, it's essential to disclose the existence of trains in the division of a railway track. In the case of a fault in the railway tracks, to further prevent any casualties due to rail accidents, the railway section is reported as occupied. Hence, the detection system design is considered to be fail-safe. Due to this occurrence, trains are prohibited to cross the threshold of entering that distinct division. Due to this collision is circumvented but effectuates postponement of trains. Furthermore,d espite fail-safe design, there are situations in which the railway section can be erroneously delineated as free, which can potentially lead to dangerous situations.

II. COMPREHENSIVE SYSTEM OUTLOOK

The system is divided into four parts: The interfacing of Hardware, Integration of Internet Of Things (IOT), Machine Learning (ML) algorithms and App development part as shown in Fig. 1

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

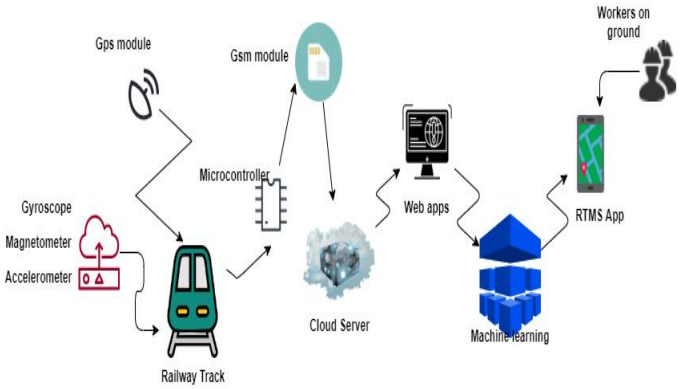


Fig. 1 RTMS outlook design

A. Interfacing of Hardware

1) MPU6050 consists of the three sensors: Gyroscope, Accelerometer and Magnetometer. The data from these sensors is acquired.
2) GPS module is used for tracking the location of the railway track.

B. Integration of IOT

1) Arduino Uno microcontroller is interfaced with MPU6050 and Gps module.
2) The acquired data is fed to Arduino Uno microcontroller using a GSM 2G module which acts as a medium of communication between the Arduino Uno and the cloud.
3) The data fed to Arduino is then sent to cloud via thingspeak.com which is an open source IOT platform to upload and download data in the cloud.
4) The 2G Module needs a Voltage source of 3.3V to 4.4V which can supply a maximum of 2A current.

C. ML approach

The data is then downloaded to the laptop and using **Python 3.0** the data is pre-processed.

1) Subsequently the system is trained using **seven** ML algorithms to give the final output.
2) The **AdaBoost** ML algorithm gives the best accuracy output.

D. App Development- RTMS App

1) The App was developed in **React Native** environment.
2) The App was designed in **NodeJs** using JavaScript programming language
3) Several libraries were imported and the google maps plugin was used.

III. IMPLEMENTATION OF THE PROPOSED MODEL

This paper deals with the sensor data and machine section. First, it acquires the sensor data and by interfacing with the data machine it uploads to the cloud. The next step is identifying the issue and matching it with the solution. Once it is identified using ML, it can be automatically fixed by alerting the engineers about the track defects via RTMS App.

A. Implementation of the Hardware design

1) *Read the Sensor Data* **:** Power up the Micro controller and start reading the Sensor data. The MPU6050 has approximately 10 seconds adjusting period. In those 10 seconds, it set and calibrates the original position of the sensor as 0 degrees in all axes and starts evaluating the yaw, pitch and roll with respect to the reference position.
2) *Obtaining the data:* Using GSM Module (SIM800L) which uses a 2G spectrum to transmit and receive data, the Xnorm, Ynorm and Znorm are uploaded to thingspeak.com (cloud website) as Field 1, Field 2 and Field 3 respectively.

B. Implementation of IoT

1) *Import Libraries and datasets :* In Python 3.0, the first step is to import the relevant and useful libraries and Datasets into the Script.

©IJRASET: All Rights are Reserved

TABLE 1
DATASETS

XFULLY DEFLATED EMPTY SERIES-II(TC) AT 90 KMPH									
Lateral Riding					Secondary Spring Deflection				
KM FROM	KM TO	SPEED	Max Accn (in g)	RI as per OREC C-116	Max on loading (mm)	Max offloading (mm)	Bogie Rotation in	Lateral movement in	Events
5600	5400	50	0.07	2.15	9.17	8.45	0.58	12.46	GS No 21-R-1002.1 Start,End GS No 20 R 979 -Start
5400	5200	70	0.06	2.11	8.30	10.44	0.48	10.33	GS No 20-Curve End GS No 19 R-5002.1-Start
5200	5000	50	0.06	2.00	10.56	9.72	2.45	28.26	Station, CS No 18-R132.9 Start
5000	4800	60	0.09	2.05	10.22	9.61	2.43	38.94	GS No 18 End GS-17-R-226.4 Start
4800	4600	60	0.05	2.01	14.19	11.39	1.69	35.45	Srirampura Station-A706.0 GS-17 End GS-16-R1015.9-Start
4600	4400	60	0.08	2.20	9.25	11.20	1.91	38.78	GS-15-R-212.1 Start,End
4400	4200	80	0.05	1.99	9.18	15.16	0.51	13.54	GS-14 R-997.9 9 m Start, End
4200	4000	70	0.05	1.97	12.09	10.64	0.59	29.93	Station, Platform GS No 13R-1050-Start
4000	3800	70	0.16	2.25	12.76	10.72	1.47	41.78	Kuvempu Road Station(KVPR) -3974.0 GS-13-R-1050-End

C. Implementation of ML Approach

1) *Naive Bayes Theorem:* The library used for evaluating the predicted values with Naive Bayes theorem is MultinomialNB from sklearn.naive_bayes.
2) *Logistic Regression:* The library used for evaluating the predicted values with Naive Bayes theorem is Logistic Regression from sklearn.linear_model.
3) *Support Vector Classifier*: The library used for evaluating the predicted values with Support Vector Classifier is SVC from sklearn.svm.
4) *Ensemble model (Average):* The library used for evaluating the predicted values with Ensemble Modelling (Average) is statistics .
5) *XG Boost Classifier:* The library used for evaluating the predicted values with XG Boost Classifier is: XGBClassifier from xgboost and loadtxt from numpy.
6) *Extreme Learning Machine (ELM):* The library used for evaluating the predicted values with Extreme Learning Machine is elm.
7) *Adaboost Algorithm:* The library used for evaluating the predicted values with Adaboost Algorithm is : AdaboostClassifier from sklearn.ensemble and datsets from sklearn..

©IJRASET: All Rights are Reserved

D. Implementaion of App Development

1) Prerequisites to develop an app as shown in Fig. 2 are:

a) Wireless Internet connection for a mobile device can connect to.
b) NodeJs, Expo from the Git.
c) Minimum Operating system requirement: Android version lollipop and ios version 10.0.
d) JavaScript,VS code editor and Windows powershell.

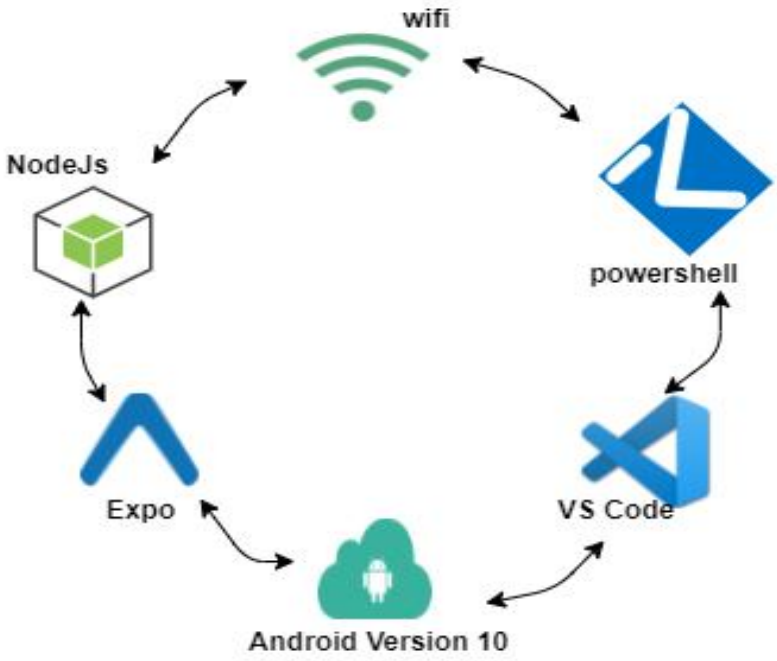


Fig. 2 Prerequisites for App Development

2) Project Setup

a) To develop the app VS code editor is used and NodeJs and npm packages are installed.
b) Plug in react native bridge is used to communicate between nodejs and react native code.
c) The app is bootstrapped with expo repository which allows to set up react native project.

3) Plug ins and Repository used

a) React native bridge is used to communicate between nodejs and react native code by Native level interactions.
b) The app is bootstrapped with expo repository which allows to set up react native project.
c) Configure expo using app.json.
d) This allows the name icon, version, splash screen, slug, background etc to the app.

4) Dependencies

a) In the app.js file the dependencies are given and the code is written.
b) The dependencies versions used is react-native-maps : v 0.27.1

5) Configuring The Google Maps Is Done Using Following Steps

a) Integrating maps in react native using react native maps.
b) The latitudes and longitudes are given as inputs to obtain the required location using google maps as shown in Fig.3
c) The react native maps package is then linked to the app.i.e linking app.js to map.js

©IJRASET: All Rights are Reserved

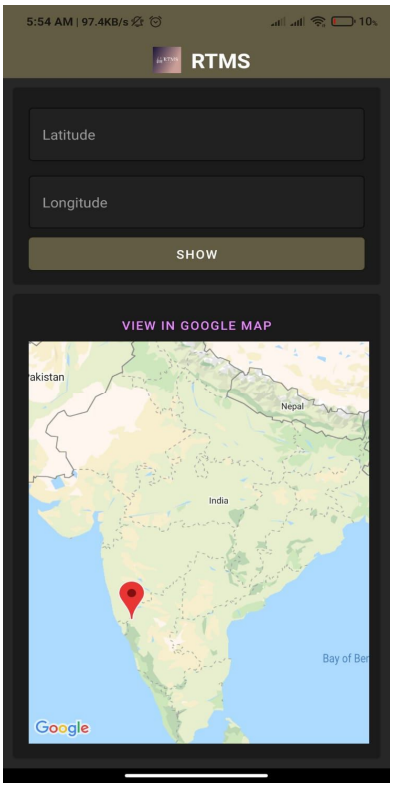


Fig. 3 Geocoordinates

6) Setting up Google maps to locate the crack:

a) The values for latitude and longitude are obtained from the output of the ML algorithms.

b) The coordinates obtained are thus sent to the workers via whats app or any other messaging tool.

c) Using the RTMS app the workers on the field can evaluate the address of the crack as shown in Fig.4

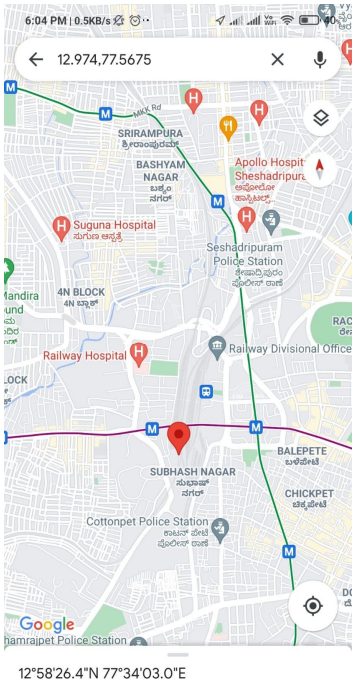


Fig. 4 Google maps

©IJRASET: All Rights are Reserved

IV. RESULTS

TABLE 2
Results Of Ml Algorithms

ALGORITHM	ACCURACY (in %)
Naive Bayes	55.42
Logistic Regression	58.84
Support Vector Classifier	85.04
Ensemble modelling(average)	66.43
XG Boost Classifier	90.90
Extreme Machine Learning	88.43
Adaboost Classifier	93.93

Table 2 represents the comparison of accuracy of various prediction algorithms. From the table, it can be concluded that AdaBoost has the highest accuracy of 93.93%. Hence, AdaBoost prediction Algorithm is finalized to be used to predict any faults in the Railway Tracks. The predicted values replace the original values of dependent variables. The stretches of Railway Tracks that are predicted to develop faults are hence obtained. Fig 7 and Fig 8 show the locations of the track where faults are predicted. Once the location of faults have been determined, the same is sent to the Railway Track maintenance workers as a text message over Whatsapp. Screenshot of the text message sent using Whatsapp is as seen in Fig 5



Fig. 5 Screenshot of the text message sent using Whatsapp

The RTMS App is built in the React Native environment usingNodeJs. The App used several packages and libraries such as Expo, Formik and React NativeBridge. The App was built in version 1.0.0. The geo-coordinates (latitude and longitude) were entered on the App screen for finding the location of the crack as shown in Fig 6. The google maps - geolocation plugin was used to link geo-coordinates to google maps. This gave the location of the fault present in the crack as shown in Fig 7.

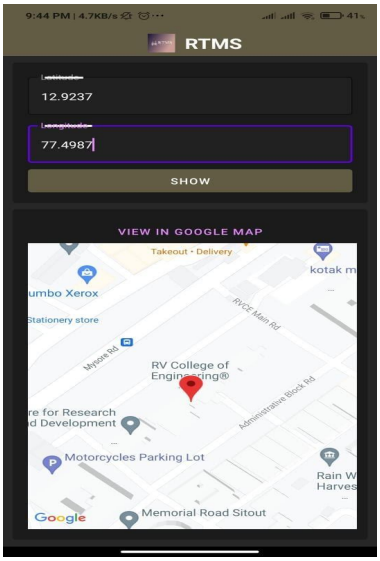


Fig. 5 Screenshot of coordinates entered in RTMS App

©IJRASET: All Rights are Reserved

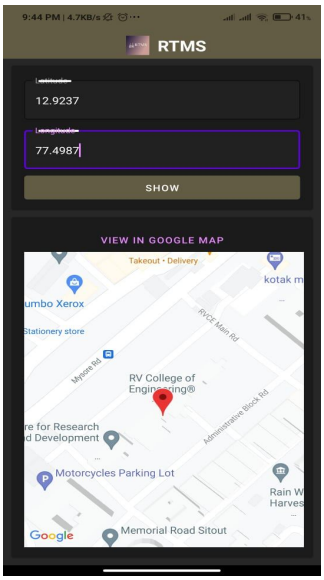


Fig. 6 Screenshot of coordinates enetered in RTMS App

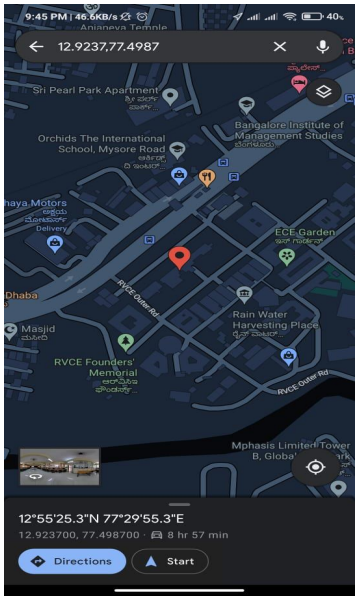


Fig. 7 Screenshot of Google maps showing the location of faults on the railway tracks.

V. CONCLUSIONS

The proposed model uses different Machine Learning tools to process the available data and predict possible defects and faults in the railway tracks. The model consists of two parts: Hardware (Embedded System) and Software (Machine Learning and App Development). The Embedded System consists of sensors (Inertial Monitoring Unit), GPS Module, Microcontroller and a GSM Shield. The data is collected from these sensors while the train is running and is transmitted to the cloud. This data is then downloaded from the cloud into a computer/laptop. A total of 3006 datasets were fed and seven different Machine Learning algorithms were used. AdaBoost algorithm gave the best accuracy of 93.93% and was chosen as the best out of the seven algorithms. Since it provided the best accuracy, AdaBoost can be applied to prevent train derailments. The coordinates of the derailment that are obtained from the GPS module are sent through a communication app viz. Whatsapp. The geo-coordinates were then converted into a physical address and navigated to google maps using google maps API in the RTMS app which was developed in React Native environment and thus the faults in the track were identified.

©IJRASET: All Rights are Reserved

REFERENCES

[1] Chellaswamy C., Geetha T.S., Vanathi A. and Venkatachalam K. 'An IoT Based Rail Track Condition Monitoring and Derailment Prevention System'. 1 Jan. 2020 : 81 – 107.

[2] Gbadamosi, Abdul-Quayyum, Lukumon O. Oyedele, Juan Manuel Davila Delgado, Habeeb Kusimo, Lukman Akanbi, Oladimeji Olawale, and Naimah Muhammed-yakubu. "IoT for predictive assets monitoring and maintenance: An implementation strategy for the UK rail industry." Automation in Construction 122 (2021): 103486.

[3] Ranjith, A., and S. P. Vijayaragavan. "Internet of Things (IoT) Based Automated Calamity Avoidance System for Railway Sectors." Journal of Computational and Theoretical Nanoscience 17.12 (2020): 5399-5408.

[4] Benazer, S. Sakena, M. Sheik Dawood, Sulochanan Karthick Ramanathan, and G. Saranya. "Efficient model for IoT based railway crack detection system." Materials Today: Proceedings 45 (2021): 2789-2792.

[5] Bhojwani, N. L., A. S. Ansari, S. S. Jirge, M. B. Baviskar, and D. N. Pawar. *"Railway Track–Crack Detection System by Using Arduino Microcontroller*."International Journall of Scientific Reasearch and Engineering Development, vol. 3, 2020.

[6] Chellaswamy, C., Santhi Ponraj, and K. Venkatachalam. *"Deep learning based intelligent rail track health monitoring system." International Journal of Innovative Technology and Exploring Engineering 8.12 (2019): 693-702.*

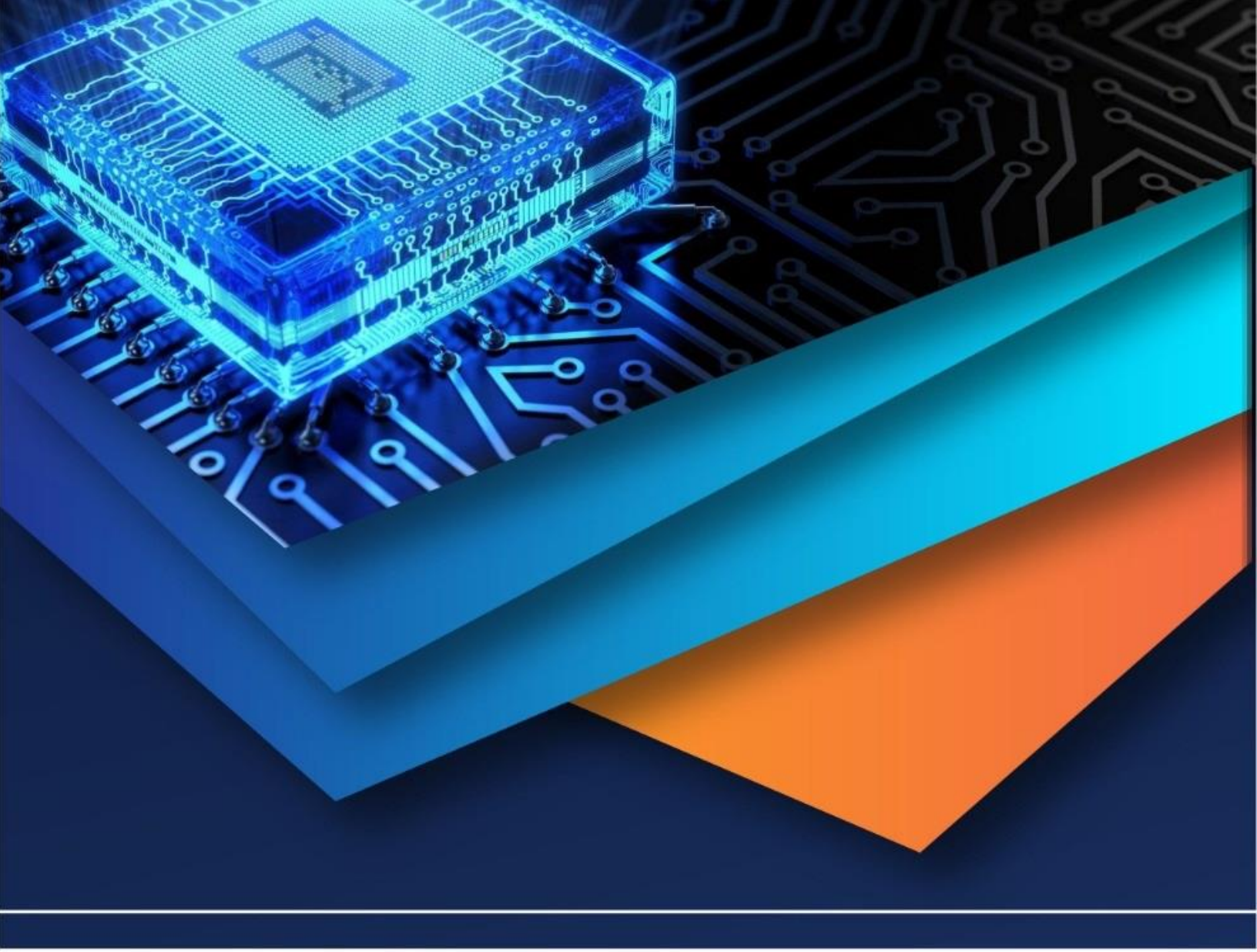
[7] Ni Y-Q, Zhang Q-H. A Bayesian machine learning approach for online detection of railway wheel defects using track-side monitoring. *Structural Health Monitoring*. June 2020. doi:10.1177/1475921720921772

[8] H. Brito, A. Gomes, Á. Santos and J. Bernardino, "JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development," *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1-6, doi: 10.23919/CISTI.2018.8399283.

[9] A. Sterling, "NodeJS and Angular Tools for JSON-LD," 2019 IEEE 13th International Conference on Semantic Computing (ICSC), 2019, pp. 392-395, doi: 10.1109/ICOSC.2019.8665625.

[10] K. Shah, H. Sinha and P. Mishra, "Analysis of Cross-Platform Mobile App Development Tools," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), 2019, pp. 1-7, doi: 10.1109/I2CT45611.2019.9033872.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)