Grenze International Journal of Engineering and Technology, June Issue

Intelligent Accident Rate, Fatality Analysis and Mapping using OpenStreetMaps

V Govardhan Pranav Suraj[1], A Advyth Vaman[2], S Kunal Achintya Reddy[3], V Poornima[4] and Krishnaprasad TR[5]
[1-5]Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, Kollam, India
Email: pranavvasireddy77@gmail.com, advythvaman06, kunalachintya, vpoornima1322}@gmail.com
{krishnaprasadtr}@am.amrita.edu

***Abstract*—Every year, India loses nearly 1.5 million citizens due to road accidents. In this work, we try to implement a two pronged approach to reduce the number road accidents in India. Firstly, we create a Machine learning model to predict road accidents subject to different road categories in India, using data from publicly available government dataset. Parallelly, we have a web application that we can use to decentralize the mapping of accident hot-spots via OpenStreetMaps. The hot spot markers vary in their intensity depending upon the accidents occurring in that locality. We also categorize the accidents into National highways, State highways and Other roads for more accurate measures so that various institutions can use them to make policies to reduce the mortality of road accidents in India. The results derived from the model are also presented that demonstrates the high efficiency of the model.**

***Index Terms*— Accident marker, Web Application, National Highway, State Highway, Machine Learning, Accident rate, Fatality, SVR.**

I. INTRODUCTION

A road accident can instantaneously destroy entire families. It is one of the major problems on Indian roadways. Every year, there is a continuous increase in the severity of the death rate due to accidents. Over 1.5 million people in this country lost their precious lives due to these mishaps. India, which is struggling with the rapid increase in the number of cars on the road and the rise in traffic accidents, needs to act quickly to improve road safety and stop this escalating problem [1].

This research sets out to lay out the intricate network of variables that cause traffic accidents in India and suggests a diversified strategy to anticipate and lessen these occurrences. Drawing on insights from various academic papers and research, we aim to provide a nearly accurate model that predicts accident rates and gives preventative measures.

The heart of our research is a toolkit of machine learning techniques that are being adopted by a growing number of different fields. By using machine learning regressions, which include methods like Bayesian Ridge, Elastic Net, and others, we try to explore the complex problems of accident prediction. These approaches, which are shown by their capacity to negotiate large datasets, present a possible means of analyzing the dynamics and patterns of road accidents in India [2].

The careful curation of large datasets from Indian government libraries, which form the basis of our predictive

Grenze ID: 01.GIJET.10.2.660

models, is central to our research technique. Various characteristics, including accident frequency, time trends, death rates, and vehicle density, are held in these datasets. We hope to build predictive algorithms that can identify the underlying causes of these occurrences and predict accident rates by using the abundance of these records [3].
Our creative inclusion of a geographic information system (GIS) into a web application enhances our analytical skills. By enabling users to report accidents in real-time, this innovative method adds insightful, practical information to our collection. Our objective is to improve road safety measures by cultivating a shared responsible and aware culture by utilizing community-driven data collection [4].
A special feature in the web application is the increasing intensity of the marker on the map which is proportional to the number of accidents reported at that place. This helps the user to regard the map more understandably and plan their travel accordingly.
This study is an attempt to tackle road safety, one of India's important public health issues. Through the integration of state-of-the-art technology and community outreach, our goal is to help the government create a more secure and adaptable transportation environment in India.

II. RELATED WORKS

The annual accident death rate in India is increasing day by day. This is the result of various factors such as rash driving, road rage, natural calamities, etc., One of the main reasons, however, is the absence of a proper availability of information and enforcement. The following papers have done extensive research on various factors and how they play a significant role in the cause and effect of road accidents.
The paper [5] has evaluated an ensemble of machine learning models to predict road accident severity based on the 2023 NZ road accident dataset. They analyzed the predicted results and applied an explainable ML technique to evaluate the importance of road accident contributing factors. In [6], the authors mentioned using Apriori and Naive Bayesian techniques to predict the different categories of vehicle accidents that are self-reported or recorded by the state. The datasets used were the state-recorded police reports or FIRs regarding the accidents.
The objective of the paper [7] is to use data mining to help develop a model that, in addition to helping identify the accident-prone areas in the nation based on various factors, also helps establish the relationship between these factors and casualties. This is achieved by grouping similar objects to smooth out the heterogeneity of the data. The paper [8] presents a brief survey on the design and models used for the detection of accidents. It also presents a survey on the analysis methods used for various reasons that cause accidents, the conventional IoT-based models that are used in detecting driver drowsiness, fatigue, drunken driving, and distractions.
The paper [9] presents a comprehensive review of the application of ML techniques in detecting the severity of accidents. They developed a web application that can be used by the user to predict the fatality of an accident in real-time.

III. PROPOSED MODEL

Our proposed model given in Fig 1, seeks to address two challenges as given below.

- Localised mapping of accidents onto a publically available open and free forum such as the OpenStreetView maps. The intensity of the markers on the map reflects the danger of the location concerning motor vehicle accidents.
- We address accident rate and fatality analysis along the national highways, state highways, and other roads in India using Support Vector Regression (SVR) after conducting an exploration of the data available.

Being a country having a growing aspiring population there is an increasing use of motor vehicles. Upon the user uploading the image of a supposed accident, the server side runs a CNN image classification algorithm to determine if the image uploaded is indeed that of an accident or not. This is done to avoid spamming the server, thus preventing incorrect information from being mapped. Since this validation is not the primary purpose of this work, we are only presenting the result of the training accuracy and validation accuracy of the accident images and are not explaining the CNN image processing. The details of the dataset are presented in the 'Dataset' section.

A. Web tool for accident mapping

Our first step is to receive the image sent by the user into the server. Once the image is validated to be that of an accident, we use the Flask web development framework and integrate it with OpenStreetMaps. The system architecture comprises a frontend as given in Fig 2, developed using the Flask web development framework [10],

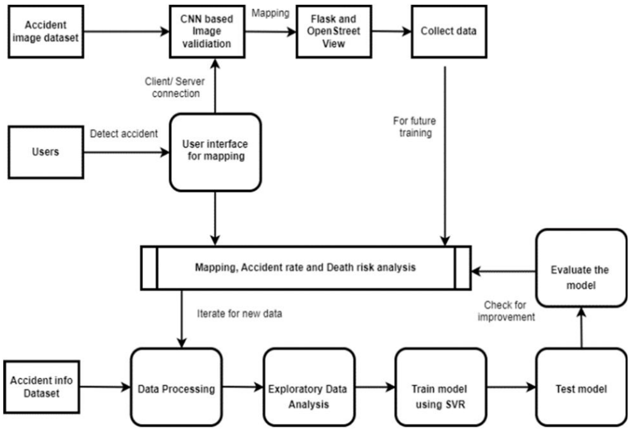


Figure 1. Proposed model

which handles user interactions and interfaces with the Geolocation Module—the Geolocation Module interfaces with the OpenStreetMap [11] API to retrieve map data and visualize accidents. We are not using Google Maps as it requires payment on many occasions. OpenStreetMap is open source and free everyone can use it. User-uploaded images are geotagged to provide context to accident reports. The module provides stakeholders with actionable insights into accident occurrences, enabling proactive measures to be taken to improve community safety. As given in Fig 3, the marker is placed over the location where the image was sent from. The intensity of the marker increases as the number of accidents geotagged from that location increases. This will help in further analysis of the reason for the increase of accidents in a particular area relative to others, like different types of roads, barriers in the road, etc. The image in Fig 3 is for demonstrative purposes. The place where a higher number of accidents are reported is having greater intensity when compared with others. The idea is to have these markers displayed in OpenStreetMaps when users type in a particular location or when a user has to traverse between places A and B.

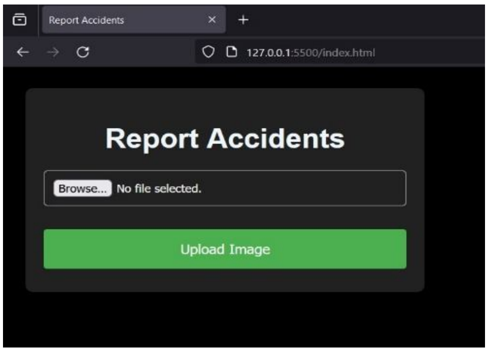


Figure 2. UI available to users to upload images of accidents

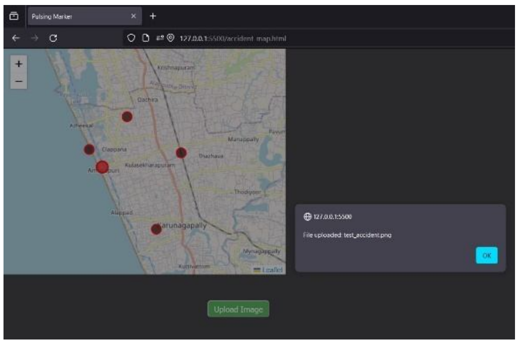


Figure 3. Mapping of accident spots

B. Support Vector Regression (SVR)

Support Vector Machine is used for both classification (categorizing) and regression (predictions) based on data. SVM can manage both linear and non-linear datasets. It aims to create a dividing line, called a hyperplane, that best separates the data points into their respective groups as given in Fig 4. Support Vector Regression (SVR) is an extension of the Support Vector Machine (SVM) algorithm. SVR operates on identifying a hyperplane that best fits the data points while minimizing the error margin. It uses kernel functions to map the input data into a higher-dimensional space, which helps to find a hyperplane that separates the data points. Data points from the training dataset that lie closest to the hyperplane are known as support vectors. SVR depends on support vectors to define the regression function. These support vectors play a crucial role in determining the position and orientation of the regression hyperplane.

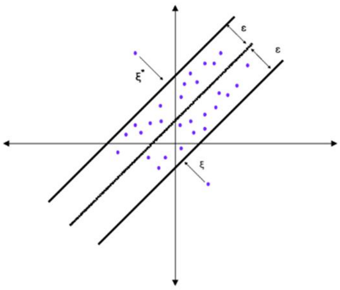


Figure 4. Support Vector Regression

C. Equations

The equation of optimal hyperplane is

$$w^T * \text{x+b} = \text{f(x)} \quad (1)$$

w is the weight vector and determines the direction of the hyperplane, x is the feature vector, and b is the bias term which is added to account for the displacement from the origin (intercept). The objective function in Equation 10 represents the primary goal of SVR. It tries to minimize the size of the tube while ensuring that the predicted outputs remain as close as possible to the desired outputs.

The term $\frac{||w||^2}{2}$ represents the regularization term, which penalizes large weights in the model. The ||w|| represents the magnitude of the normal vector w to the surface being approximated. It determines the orientation of the hyperplane that divides the data points in feature space. By minimizing ||w||, the tube surrounding the surface becomes narrower, thereby maximizing the margin between the data points and the surface.

$$\min \frac{1}{2} \left||w|\right|^2 \quad (2)$$

The epsilon-insensitive tube specifies that for each training data point (xi, yi), the predicted output f (xi) should lie within yi − ε and yi + ε, where ε is the margin of tolerance for errors.

$$y_{i} - f(x_i) \leq \varepsilon \quad (3)$$

$$f(x_i) - y_i \leq \varepsilon \quad (4)$$

To simplify, let's denote f (xi) = $w^T x_i + b$, where w is the weight vector and b is the bias term.

Substituting f (xi) into the constraints, we get:

$$y_i \text{-} (w^T x_i + b) \leq \varepsilon \quad (5)$$

$$(w^T x_i + \text{b}) \text{ - } y_i \leq \varepsilon \quad (6)$$

The width of the epsilon-insensitive tube is ε, which plays an important role. A small ε value indicates a lower tolerance for error, brings data points closer to the boundary and more data points become support vectors. Conversely, the Larger value of ε allows predictions to deviate further from the desired output, resulting in fewer support vectors.

$$|y_i - (w^T x_i + b)| \leq \varepsilon \quad (7)$$

This equation represents the epsilon-insensitive tube around the regression function f(x). It ensures that the absolute difference between the predicted output and the actual output is less than or equal to ε for all training data points.

The main aim of SVR is to minimize the error between the predicted output of a function and the actual output. To achieve this, it uses an approach called epsilon-insensitive loss function. This loss function $L_\epsilon(y, f(x, w))$ penalizes predictions that deviate from the desired output by more than a specified margin ε.

f (x, w) is the linear combination of the input features x with the weights w and adjusted by the bias term b.

$$f(x, w) = < w, x_i > +b \quad (8)$$

$$L_\epsilon(y, f(x, w)) = \begin{cases} 0, & |y_i - (w^T x_i + b)| \leq \epsilon \\ |y - f(x, w) - \epsilon, & |y_i - (w^T x_i + b)| > \epsilon \end{cases} \quad (9)$$

To adopt a soft margin slack variables ξ and $\xi *$ are introduced to account for points that lie outside the epsilon-insensitive tube. These variables represent the distance of data points from the margin boundaries and allow for a certain degree of tolerance for errors. The optimization algorithm can control the tolerance for points lying outside the tube by adjusting the values of ξ and $\xi *$.

Based on Equations 2 and 9, the optimization problem in Equation 10 is obtained;

$$\min_{w,\xi,\xi*} \frac{1}{2}||w||^2 + C \sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{10}$$

c is a regularization parameter that controls the balance between maximizing the margin and minimizing the error. when c is large, the optimization algorithm minimizes the error term in the objective function, resulting in more accurately fitting the training data, which may lead to increased model complexity. Conversely, when c is small, the regularization effect is weaker, and the optimization algorithm prioritizes maximizing the margin. This results in a simpler model with a larger margin, which may generalize better to unseen data but accuracy might lead to compromise in the precision of fitting the training data.

IV. DATASET

Year	National Highways			State Highways			Other Roads		
	Accidents	Killed	Injured	Accidents	Killed	Injured	Accidents	Killed	Injured
2003	31.4	38.6	30.1	22.4	28.2	26.7	46.2	33.3	43.2
2004	30.3	37.5	30.8	23.5	26.9	24.9	46.2	35.6	44.3
2005	29.6	37.3	31.3	23.6	27.2	25.7	46.8	35.4	43.0
2006	30.4	37.7	30.8	18.5	26.8	24.9	51.1	35.6	44.4

Figure 5: Sample Dataset

The dataset for accidents on roads is collected from the Government of India's dataset. The dataset for accident image verification is from [12]. A sample dataset on accident data on roads is presented below
The features used in the dataset are the number of accidents, Number of people killed, and Year.

A. Dataset exploration

To examine the relationship between the different features, we constructed a scatterplot analysis as shown in Fig 6 using the seaborn [13] and matplotlib [14] library.

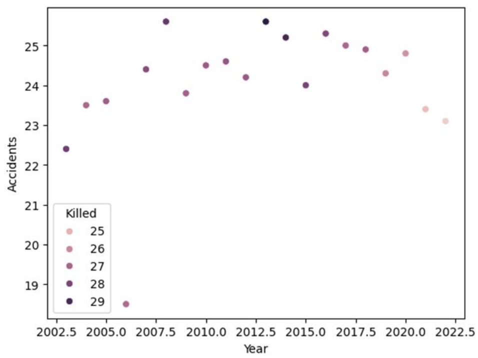


Figure 6. State highways - Accidents, People Killed, Year

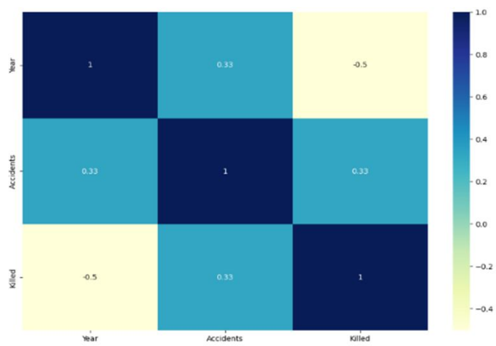
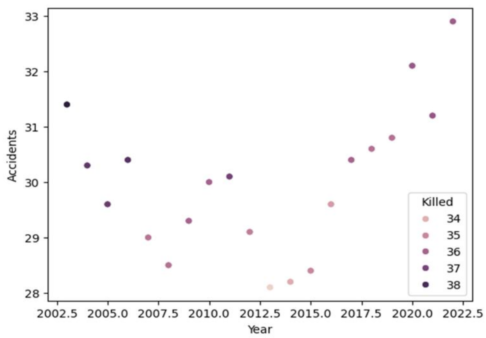


Figure 7. State highways – Heatmap – Correlation

The scatterplot reflects that for state highways, as the number of years progresses, the number of people being killed due to accidents decreases, although not in a strictly linear format. To get a better scalar understanding of the relationship between features, we use a correlation heatmap analysis as given in Fig 7. From the heatmap, we can infer that the number of people killed on state highways as the years progress is declining whereas the number of accidents shows an increase, though not a strong proportionality. We explore the inter-feature relationship via a scatter plot, for National Highways. A preliminary analysis of Fig 8 shows us that the number of people killed during 2002 is the highest and keeps reducing as we approach 2012 and then starts to increase again towards the year 2022, although it does not reach the high levels of 2002. We also do a correlation analysis of different features from the National Highway perspective. We find that just like State highways, on National highways also the number of people getting killed in accidents is reducing as the years progress, but the number of accidents per year is showing an increasing trend although at a slower rate. The heatmap metric of 0.37 for Accident-Years shows a positive relationship, but as it is on the lower half of 0 to 1 it cannot be conduced as a strong positive relationship.



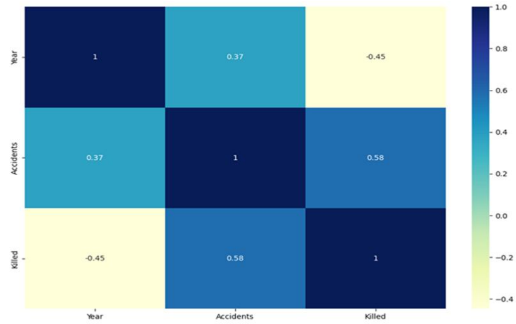
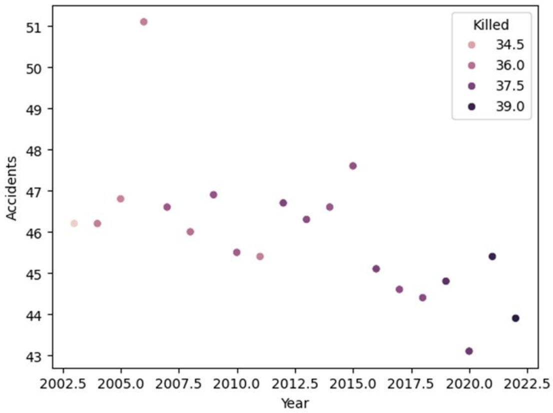


Figure 8. National Highways – Accidents, People Killed, Year

Figure 9. Heatmap-National Highway

Fig 10 represents the graph that shows the relationship between the number of accidents, number of people killed, and year on 'other' roads of India which are not highways or state highways.

An analysis of the graph shows that the number of accidents over the years has come down, but the number of people getting killed due to accidents is increasing. This is reflected in the heatmap correlation given in Fig 11. The heatmap analysis also shows that the Year-Killed correlation is 0.88 but the Year-Accidents is -0.62.



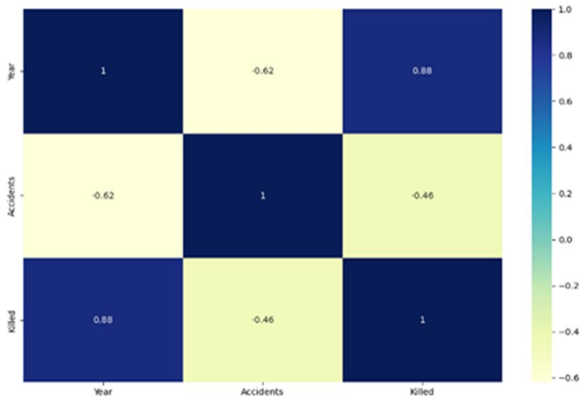


Figure 10. Other roads – Accident, People Killed, Year

Figure 11. Heatmap- Other roads

B. Result

MobileNetV2 pre-trained model is used to detect whether an image uploaded by the user is indeed that of an accident or not. By leveraging transfer learning, we harness the knowledge learned from a large dataset to accelerate the training process and improve the model's performance on our specific task. During training, we split the dataset into training (80%), validation (10%), and testing (10%) subsets. The dataset used comprised approximately 1200 images. On our test dataset, the CNN model achieved an accuracy of 92.5%, precision of 91.2%, recall of 93.8%, and F1-score of 92.4%. The results of accident image verification are given in Fig 12. Visualizing the training and validation loss as in Fig 13 and 14, as well as accuracy, over epochs allows us to monitor the model's progress and diagnose potential issues. We can see that in Fig 13 the training loss is reducing with every epoch whereas the training accuracy is increasing. The same can be observed in Fig 14 where the validation loss is steadily decreasing while the validation accuracy is increasing with every epoch. There are three features of particular importance in the dataset as can be understood from the Heatmap correlation analysis in Fig 7, Fig 9, and Fig 11.

They are 'Year', 'Number of accidents', and 'Killed/ Fatality (number of people killed)'. The input features are Year and Number of accidents while 'Fatalities' is the target feature.

The relationship between the three features is not strictly directly proportional to being linear, hence we use the Linear Regression (LR) and Support Vector Regression (SVR) in predicting the fatality. We use Python language and Jupyter Notebook for this prediction and analysis purpose.

The result is further categorized into

- National highway accidents.
- State highway accidents.
- Other road accidents.

Figure 12. Accident image verification result

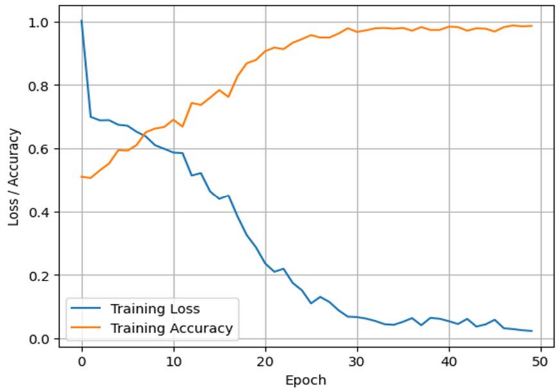


Figure 13. Training accuracy of accident images

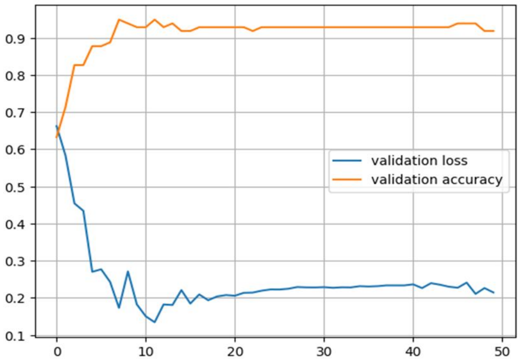


Figure 14. Validation of accuracy of accident images

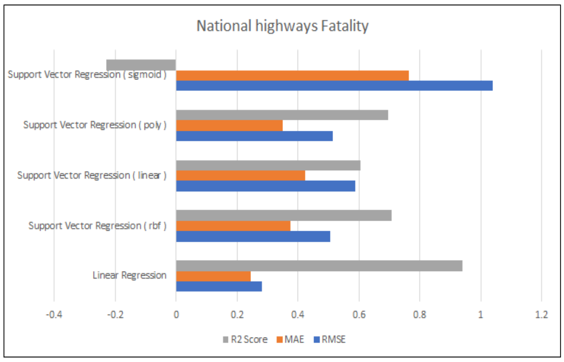


Figure 15. Fatality regression – NH

C. National Highway (NH) accident

In NH fatality analysis, we compare Linear Regression with Support Vector Regression (SVR). The results show that Linear Regression(LR) is better than SVR as the Root Mean Square Error (RMSE) score of 0.28 is much lower when compared to SVR(kernel-poly) which is 0.51. The Mean Absolute Error (MAE) is also the lowest for LR. The R-squared (R2) score of LR is 0.93, which is higher than the R2 scores of SVR.

D. State highway (SH) accidents

In SH fatality analysis, The results show that SVR (kernel-RBF) is the most accurate against other models mentioned earlier as its RMSE, MAE, and R2 scores are 0.302, 0.21 are lesser than others and its R2 score of 0.89 is higher than other models.

E. Other Road Accidents

In other roads fatality analysis, The results show that SVR(kernel-RBF) is the most accurate against other models mentioned earlier as its RMSE, MAE, and R2 scores are 0.523, 0.41 are lesser than others and its R2 score of 0.45 is higher than other models, except LR. However, the overall result parameters are heavily in favor of SVR(kernel-rbf).

V. CONCLUSION AND FUTURE WORK

In this work, we look at the data on accidents from National Highways (NH), State Highways (SH), and other roads in India. We explored the data and due to the non-linear nature of the data features, we use Support Vector Regression to predict the accident and death rate occurrences. We find that the SVR is very efficient as it gives an RMSE of 0.28 for NH, an RMSE of 0.32 for SH, and an RMSE of 0.5 for other roads. We also provide a UI option for the user to upload any images related to accidents and at the backend we verify whether the image is that of an accident or not using CNN. If it is then we map the accident location onto OpenStreetMaps.

In future work, we would like to enhance the current work by having more data, by providing a warning alert to the users when they approach a location known for its high rate of accidents.

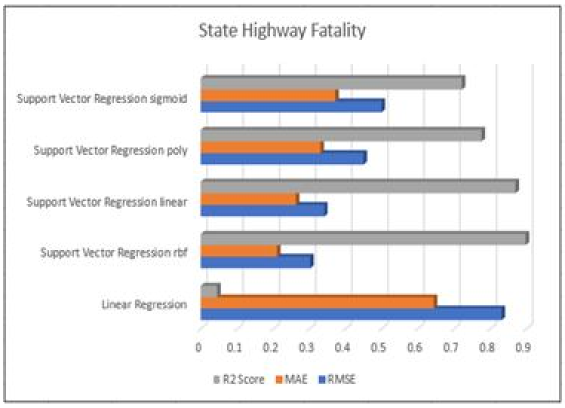


Figure 16. Fatality regression – SH

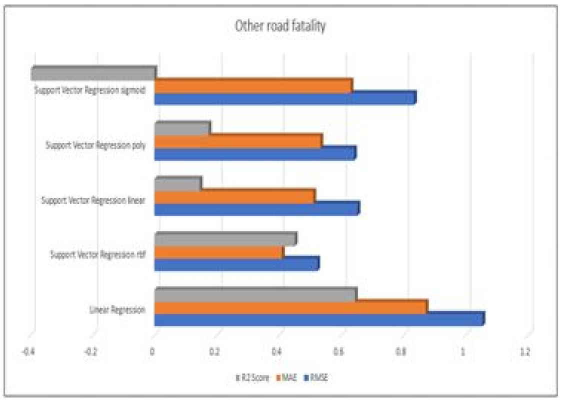


Figure 17. Fatality Regression – Other Roads

REFERENCES

[1] M. L. S. Kumar, U. S. Ashritha, Y. Sumanth, S. Hafeez, P. P.K, and A. Chandran, “Smart Accident Detection Sys-tem,” in 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, Aug. 2021, pp. 1675–1679. doi: 10.1109/ICESC51422.2021.9532702.

[2] J. P. Kuriakose and N. Jayasree, “Comparative Study of Machine Learning and Deep Learning Based Motor Vehi-cle Accident Judgment Classification,” in 2023 Fifth International Conference on Electrical, Computer and Com-munication Technologies (ICECCT), IEEE, Feb. 2023, pp. 01–06. doi: 10.1109/ICECCT56650.2023.10179758.

[3] Bhagyalekshmi A and Lasitha A, “A Major Challenge For Living: Road Traffic Accidents In Kerala, India In The Course Of Last Ten Years.,” International Journal of Scientific & Technology Research (IJSTR), vol. 9, no. 2, 2020.

[4] P. K. Binu, K. J. Sredhey, and R. S. Anuvind, “An Iot Based Safety and Security Mechanism for Passenger Vehi-cles,” in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, Jul. 2019, pp. 1502–1506. doi: 10.1109/ICICICT46008.2019.8993177.

[5] S. Ahmed, M. A. Hossain, S. K. Ray, M. M. I. Bhuiyan, and S. R. Sabuj, “A study on road accident prediction and contributing factors using explainable machine learning models: analysis and performance,” Transp Res Interdiscip Perspect, vol. 19, p. 100814, May 2023, doi: 10.1016/j.trip.2023.100814.

[6] M. Nidhi. R and M. Kanchana V, “Analysis of Road Accidents Using Data Mining Techniques,” International Journal of Engineering & Technology, vol. 7, no. 3.10, p. 40, Jul. 2018, doi: 10.14419/ijet.v7i3.10.15626.

[7] Jain, G. Ahuja, Anuranjana, and D. Mehrotra, “Data mining approach to analyse the road accidents in India,” in 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Di-rections) (ICRITO), IEEE, Sep. 2016, pp. 175–179. doi: 10.1109/ICRITO.2016.7784948.

[8] S. Savitha and N. Sreedevi, “A Study on Accident Detection Systems Using Machine Learning,” 2023, pp. 675–686. doi: 10.1007/978-981-19-7455-7_54.

[9] V. D. Chennupati, Y. V. Reddy, B. S. K. Kanth, K. V. Rishwanth, K. K, and K. L. Nisha, “A Machine Learning-Powered Web Application for Accident Severity Detection,” in 2023 5th International Conference on Inventive Re-search in Computing Applications (ICIRCA), IEEE, Aug. 2023, pp. 540–545. doi: 10.1109/ICIRCA57980.2023.10220796.

[10] M. Grinberg, Flask Web Development: Developing Web Applications With Python. O’Reilly Media, Inc., 2018.

[11] “OpenStreetMap,” https: //www.openstreetmap.org.

[12] C. K. C, “Accident Detection From CCTV Footage.” Kaggle, 2020. doi: 10.34740/KAGGLE/DSV/1379553.

[13] M. Waskom et al., “mwaskom/seaborn: v0.8.1 (September 2017).” Zenodo, Sep. 2017. doi: 10.5281/zenodo.883859.

[14] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” Comput Sci Eng, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.