FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 49 (2024) No. 3
ISSN 0867-6356
DOI: 10.2478/fcds-2024-0012 e-ISSN 2300-3405

Deep Learning Model for Detection and Severity Analysis of Car Accidents

Tanusree Chatterjee [*], Priya Roy [†], Kamalesh Karmakar [‡], Shiladitya Munshi [§], Sanjib Roy [¶]

Abstract. On-road car accidents are immensely unfortunate but quite common occurrences worldwide. Instant data-centric and informed decisions of crisis management are rarely experienced due to the absence of real-time car accident detection and severity analysis mechanisms. On this background, the current paper presents a deep learning model for car accident detection and analysis of its severity so that the crisis management activities might follow without any delay saving invaluable human lives. The existing works lack in using time-series data, the proper learning model for accurate prediction, and minimizing the time taken in post-accident scenarios for the victims to receive immediate medical help. This paper introduces the Long Short Term Memory (LSTM) model in conjunction with the Gradient Boosted Regression Trees (GBRT) technique for the determination of car accidents with different levels of severity. The proposed model works with the accelerometer and gyroscopic data collected through an application installed in the smartphones of the users inside the car. The LSTM-GBRT hybrid model is proposed to achieve higher accuracy than LSTM which deals with time-variant data. The satisfactory performance of the proposed technique has been reported and the results are extensively investigated in comparison with another hybrid technique such as LSTM with Random Forest (RF) as well. The statistics confirm the superiority of the proposed model over other parallel models in terms of several performance metrics, like Accuracy, Precision, etc.

Keywords: Deep Learning, Long Short Time Memory, Gradient Boosted Regression Tree, Car Accident Analytics

[*]Future Institute of Technology, Kolkata, West Bengal, India, tnsr.chatterjee@mail.com
[†]Future Institute of Engg. & Management, Kolkata, West Bengal, India, priyaroycse@gmail.com
[‡]Techno International New Town, Kolkata, West Bengal, India, k.karmakar.ju@gmail.com
[§]Techno India University Tripura, Agartala, Tripura, India, shiladitya.explorer@gmail.com
[¶]Future Institute of Technology, Kolkata, West Bengal, India, snjry2007@mail.com

1. Introduction

According to the World Health Organization (WHO), every year, approximately 1.3 million people die due to road accidents globally, i.e., every 26 seconds, one person is killed. Where 93% of road accidents occur in middle- and low-income countries like India, Bangladesh, South Africa, Afghanistan, etc., unfortunately, 60% of the world's vehicles are in these countries [1] [2]. In most such countries, road accidents cost 3% of their Gross Domestic Product. On the roads of the most populous country like India, 1.5 lakh people die every year due to accidents. Around 1130 accidents and 422 deaths occur every day; in turn, 47 accidents and 18 deaths occur every hour [3]. In 2022, 155622 deaths due to road accidents were registered in India [4] which is a 16% rise as compared to 2020[5].

There are several causes of increasing road accidents and related casualties, such as excessive number of vehicles, bad road conditions, improper conditions of drivers, etc. But, the most unfortunate reason among these all, is the delay in receiving immediate medical help for accident victims when the accidents happen on the sparse regions of the National Highways [1] [6]. The reason is the victims' relatives and the emergency service providers do not receive the message on time. As per reports, only near to 50% of road accident victims receive immediate medical help within the golden hour but the rest are unable to get the same within the proper time. The lack of knowledge of the actual accident and its severity are the causes behind it [9] and as an unfortunate result, many of the victims of the accidents die.

Several methods (software and hardware) have been proposed so far ([13], [11], [10], [7], [20],etc.) towards the solution approach of the aforementioned problem. However, very few of them have worked with time series data to validate the performance and results, which is a very crucial type of data in such accident scenarios. Some of them use image processing which is itself a very costly affair. Moreover, unlike the works, the proposed model focuses on post-accident scenarios analyzing the data with the proposed hybrid deep learning model collected inside cars by a smartphone application. It is done to minimize the time taken from an accident occurrence to the first medical help received by the accident victim, irrespective of the outside condition, without any image analysis and exclusive hardware support. Thus it results in a user-friendly and cost-effective approach. A detailed discussion of such related works is presented in the later section of the paper.

However, the concerning states of car accident scenarios motivate us to develop such a data-centric artificially intelligent solution that detects an accident after it occurs and analyses the severity of the same. The contributions of the work can be mentioned as follows.

- To propose a hybrid deep learning model with Long Short-Term Memory(LSTM) [12] and Gradient Boosted Regression Trees (GBRT) to detect an accident after it occurs.

- To analyze the severity of the accident and classify the data in different severity classes that help an informed decision to be made augmenting proper communication for immediate medical assistance.

- To validate the proposed model by testing with real-life dataset after accumulating and preprocessing the data collected from the in-built sensors of the smartphone of the users inside a car. The sensors used in this framework are Accelerometer, Gyroscope, and Geo-location sensors.

The work proposes to notify the accident, once detected (along with the severity), the emergency service providers such as the nearest hospitals, and police stations as well as the relatives of the accident victim. As a result, the victims, even in a sparse region, can receive immediate medical help.

In this context, the rest of the paper is organized as follows: The section 2 discusses notable research works of the same domain in the recent past, followed by the section 3 which elaborately reports the problem statement, related considerations, solution requirements, and preliminary discussions on LSTM and GBRT, along with the justification for utilizing these models. The next section i.e. section 4 introduces the deep learning model with dataset preparation. Further, it introduces the LSTM with GBRT-based learning algorithms for the detection of car accidents and the severity of the impact. The results of the extensive experiments with the proposed deep learning model and its comparison with other deep learning techniques are presented and explained in section 5. The current study officially ends in section 6 where conclusive remarks and the future direction of the presented research have been noted.

2. Literature Review

In [18], authors propose a visual notification system that notifies drivers about accident-prone features in any location based on real-time images obtained via dashcam. The work proposes a model that classifies an accident hotspot and a non-hotspot. This attentive driving scheme uses Class Activation Maps (CAMs) for feature extraction combined with the Convolutional Neural Network (CNN) based attention model. The model learns complex accident-prone features for identifying an image as an accident hotspot with an accuracy of up to 90%.

The authors in [13] propose a deep learning framework referred to as Pattern Recognition Chain (PRC) for car accident detection using basic multimodal sensors in cars. The PRC consists of several phases. The sensors are chosen in the data acquisition phase based on their availability. Data preprocessing uses sensor calibration, normalization, and segmentation to prepare the data for further analysis, and the feature extraction phase derives the most relevant information from each data segment. The classifiers are trained to distinguish between accident and non-accident events. The testing and evaluation on the real (accident) dataset show that CNN with an SVM classifier outperforms other approaches.

In [11], authors propose a deep-learning-based IoT system, namely Deepaccident, which includes an in-vehicle infotainment (IVI) telematics platform with a self-collision detection sensor, a front camera, and a cloud-based management system. When an accident is detected, the information is updated in the cloud server, and the related emergency notification is sent via a web call to the emergency centers. It detects high-speed head-on collisions and any single-vehicle accidents. The comparative study shows that Deepaccident outperforms other schemes in terms of accuracy and abnormal acceleration detection.

Driven by the idea of reducing the response time of reporting a car accident, the work in [7] uses Dynamic Time Wrapping (DTW) and Hidden Markov Model (HMM) to examine the severity of the accident from the accelerometer data of the car passengers' smartphones. The accidents are notified to the first responders as well. While the car accidents have been simulated with a hardware model, the paper reports a control-based multi-level two-class classifier system. Based on the control logic, the DTW or HMM is chosen as the classifier. However, the success percentage at different levels in the study has been reported at up to 98%.

Authors in [15] have proposed a framework that employs a stacked sparse autoencoder(SSAE) deep learning model to predict the severity of accidents based on several factors such as condition, the speed limit of the car, road & weather condition, driver's characteristics, etc. Primarily, the Catboost model is used to analyze and remove the factors with low correlation. Then K-means algorithm is used to classify the dataset based on the geographical information. Finally, the SSAE predicts the injury severity in each data class. The performance is evaluated in terms of Precision, Recall, and F1.

Authors in [21] and [27] use the LSTM model and LSTM-GBRT models respectively to predict an accident. The work in [21] decides the deployment of a wearable bike airbag system analyzing the input values from three acceleration-axis and three angular velocity-axis sensors based on the driver motion states. The study claims that the model has a higher judgment accuracy than the existing CNN. On the other hand, [27] proposes a road traffic accident prediction model with LSTM-GBRT to predict accident safety level indicators for future accidents. Here LSTM has been used to capture the time-dependent information in data (such as passenger traffic, highway mileage, population, death toll, etc.)and GBRT has been used for high robustness of ensemble learning in model training. The work compares the error rate, root mean square error, etc. with other existing models to identify the deviation from the true value and establishes that the proposed model outperforms the other existing models in predicting accidents.

Works in [24]-[14] use LSTM and CNN to predict accidents. In [24], the deep learning model is used to engineer the previous sequence of accident data i.e. crash picture frames for predicting the next certain hours of traffic accident picture frames. The experimental results show that with seven-day crash sequences on Cobb County sparse data, the model can predict the accident image of the next day. The model in [26] predicts the frequency of traffic accidents during a tourism season by analyzing the time-series data of daily frequencies of traffic accidents with fatalities and injuries in a tourist place. The experimental results show a lower root mean square error (RMSE) compared to other existing models. [14] proposes a real-time crash risk prediction model on arterials using LSTM-CNN. It explicitly learns from the various features, such as traffic flow characteristics, signal timing, and weather conditions. The LSTM captures the long-term dependency while CNN extracts the time-invariant features. The comparative performance shows that the model outperforms other models like

LSTM, XGBoost in terms of Sensitivity, False alarm rates, etc.

In [10], a hardware system has been developed that consists of an Arduino Mega as a microcontroller and several sensors, like a force sensor, a Flame sensor, an Accelerometer, etc. It also has a Global Positioning System (GPS) and a Global System for Mobile Communications (GSM) transmitter module. The microcontroller collects the sensory data and estimates the speed, rotation, smoke levels, etc. of vehicles. Different classifiers, such as the Gaussian Mixture Model (GMM), Naive-Bayes Tree (NB), Classification and Regression Trees (CART), etc. are used to classify an accident, along with the severity level of the accident based on the G-force value and fire occurrence. The performance results in terms of precision, recall, and harmonic mean value show that CART is the most prominent classification model. The authors in [20] develop a hardware system with Arduino UNO which consists of a Limit switch, accelerometer, GSM module, and GPS system. Here, the accelerometer and limit switch are used to estimate the acceleration and force of the vehicle, respectively, and detect if an accident occurs based on certain threshold values. The exact location of the accident is sent using the GPS. The GSM is used for connecting to the network and sending notifications to emergency services and to the relatives of the victims via messages.

Most of the above works focus on accident detection or prediction. There are some software-oriented systems, and few are hardware-based.
The model in [18] has been learned using a dataset created with Google Street View images without considering different conditions such as weather, time of data, and image classification, combining the CNN backbone with an attention-based module. Using only visual features has made the process more complex. Authors in [13] use a standard framework, Pattern Recognition Chain but do not explain the reason for selecting the same. It uses the sensory data from the built-in sensors of cars, so the scheme does not work for cars where the sensors are not present. The findings lack any comparative study with other results. However, both [18], [13] do not focus much on notification management systems. In [11] the vehicle head-on photos used by the method are very irregular, and the accuracy of the image recognition of the original learning model could be insufficient. The image analysis itself increases the complexity of the system, and it cannot work without the IVI telematics platform, which is not considered to be a cost-effective system. The authors in [7] report an interesting way of detecting one of three severity levels (low, mid, and high) of car accidents; the lesser number of accident instances and comparatively high time requirement for classification might surface as concerns. Moreover, as the study reports stage-wise success percentages, it is hard to compute the overall success percentage from the reported experiments.

The comparative performance results of the proposed method in [15] with other Baseline models do not pose any remarkable difference. The accuracy result of the proposed model is not clear as well. Anyway, it does not also work with time-series data which is crucial for real-life accidents. In [21], airbags are operated by judging the bike accident situations using sensor thresholds and driver's motions only. The use of so many sensors exclusively makes the model a costly affair. It does not also consider the accident scenarios by vehicles other than bikes. The model proposed in

[27] does not include a few important features related to vehicles and may exclude a few features used as input for future accident predictions. However, none of the works in [15]-[26] consider the measurement of accident severity and so do not take into account any severity classes of accidents. The performance evaluation and the comparative analysis with existing models in terms of important parameters such as Accuracy, F-score, Precision, etc. are lacking in the works [24], [26].

The works in [10] and [20] develop hardware systems for accident detection. The [10] uses too many sensors, thus making the integrated system unnecessarily complicated and costly, and the selection criteria of the classifiers are not clear. The authors in [20] use only threshold values of the sensory data for accident detection.

The aforementioned scenarios motivate us to develop a deep learning model for car accident detection which would be driven through a mobile application. The app uses in-built sensors of the smartphone, such as an accelerometer, gyroscope, and GPS, and can run on any smartphone, irrespective of the operating system. It continuously collects sensory data and sends the data to the LSTM-GBRT based deep learning model running on a remote server in a sliding window time frame (every 15 seconds) only if an accident is suspected based on certain adaptive threshold values as reinforced by the deep learning model. The adeptness of the threshold value can be attributed to the dynamically changing road conditions near the data collection GPS location. The thresholding system has been brought into the picture to reduce the unnecessary data exchange from the smartphone client to the remote server where LSTM-GBRT based deep learning model is being trained continuously. After receiving the data, it is then analyzed by the model to classify the data into any of the four classes representing *No Accident*, *Minor Accident*, *Major Accident* and *Fatal Accident*. After confirming an accident occurs within 20 seconds of buffer time the decision can be notified through the mobile application to different chosen points. The notification management might follow any standard technology and hence is out of the scope of the current study.

3. Research Methodologies

The current section discusses the problem formulation and the solution requirements to offer a generic guideline for deep learning model design principles. On the basis of these guidelines or objectives, subsequent subsections report the justifications for selecting LSTM and GBRT modeling for accident detection and severity prediction. However, for the completeness of the paper, the current section ends with preliminary-level background studies on LSTM and GBRT.

3.1. Problem Formulation

The core objective of the proposed LSTM-GBRT based model is to

Table 1. Symbol Table

The symbol for Raw Sensory Data	Details
$A_x(t), A_y(t), A_z(t)$	Accelerometer data of x, y and z axis at time t
$G_x(t), G_y(t), G_z(t)$	Gyroscopic data of x, y and z axis axis at time t
¡$lat, long$¿	latitude and longitude of a location
Symbol for Derived Data	**Details**
$\overline{A}(t)$	Mean of $A_x(t), A_y(t), A_z(t)$
$\overline{G}(t)$	Mean of $G_x(t), G_y(t), G_z(t)$
SD_{acc}	SD of $(A_x(t), A_y(t), A_z(t))$ over the time period $t-\Delta t$ to t
SD_{gyro}	SD of $(G_x(t), G_y(t), G_z(t))$ over the time period $t-\Delta t$ to t
$G-Force$	The gravitational force equivalent of acceleration
$GyroForce$	The gravitational force equivalent of angular momentum
$\overline{A_x^m}(t)$	SMA of $A_x(t)$ over the time period $t-\Delta t$ to t
$\overline{A_y^m}(t)$	SMA of $A_y(t)$ over the time period $t-\Delta t$ to t
$\overline{A_z^m}(t)$	SMA of $A_z(t)$ over the time period $t-\Delta t$ to t
$\overline{G_x^m}(t)$	SMA of $G_x(t)$ over the time period $t-\Delta t$ to t
$\overline{G_y^m}(t)$	SMA of $G_y(t)$ over the time period $t-\Delta t$ to t
$\overline{G_z^m}(t)$	SMA of $G_z(t)$ over the time period $t-\Delta t$ to t
The symbol for Vehicle Related Data	**Details**
V_{id}	vehicle id
V_T	vehicle type
t	timestamp of data collection
Symbol for severity classes	**Details**
C_{na} or c_0	Represents no accidents
C_{minor} or c_1	Represents minor accidents
C_{major} or c_2	Represents accidents with major collisions
C_{fatal} or c_3	Represents accidents with fatal impacts

- detect accidents with the highest accuracy possible
- predict the severity level of the accident to gauge the urgency of immediate medical attention and civic help

Table 1 reports some notations & symbols with their detailed meaning pertaining to the official specification of the problem statement. These symbols will seamlessly be used in the entire article with respect to the proposed deep-learning modeling algorithms. On the basis of Table 1, the following paragraph defines the problem statement.

Problem Statement: Given m number of labeled instances having

- raw acceleration and gyroscopic data $A_x(t), A_y(t), A_z(t)$ and $G_x(t), G_y(t), G_z(t)$ respectively
- derived data like $\overline{A}(t)$, $\overline{G}(t)$, $\sigma_A^2(t)$, $\sigma_G^2(t)$, Standard Deviation (SD), Simple Moving Average (SMA), etc.

- accident intensity labels C_{na} or c_0, C_{minor} or c_1, C_{major} or c_2 and C_{fatal} or c_3

to train the model and an unseen test set of n instances of raw sensor data; design a framework with the labeled data to

- detect car accident (if testing data does not fall into c_0); and
- predicts the severity level of the accident(as per the categorization of training data into any of c_1, c_2 and c_3) with the highest possible accuracy

3.2. Solution Requirements

Against the backdrop of the problem statement reported above, the proposed solution should meet the following criteria with the justifications mentioned:

- **Accident classes must be properly defined and labeled**: The incidence of a car accident is a binary phenomenon, but the intensity of the car accident is a non-binary incidence required to be modeled with multiple classes having closed boundaries. The solution to the problem. Hence, must work as a multi-class classifier, where the class definitions must be completed during data preparation.
- **Learning algorithm must consider the temporal change of the data**: A car accident could be considered as a chain of events where the car parameters change with time and ultimately result in an accident. Therefore the learning algorithm must be able to model the sequence of change of data on a temporal scale for better results.
- **The derived data extraction from the dataset has to be explicit and must conform to external labeling**: The severity of any car accident is highly a subjective narration and labeling of the severity is mostly done with instinctive approaches when most to most of them are correct. However, the features of car accidents distinguishing one severity level from the other severity level are generally abstracted. Hence, the proposed deep learning solution must possess a strong internal feature extraction mechanism from the training data that should conform to the data labels. item **Server side computation must be minimized to maintain scalability**: Accident is a rare incidence in a car's lifecycle. Most of the time, the huge and high-velocity sensory data uploaded to the server from the sensors within a particular car would result in a non-accident case. Therefore to minimize the load on the server, and to support the scalability of the system, only selected data above a threshold value should reach to server from the passenger's mobile phone. All the sensory data have to be computed locally within the smartphone, and if the data are above a threshold level, then only that data would be uploaded to the server. This threshold reinforced by the deep learning model must be adaptive to the car as well as the geo-fenced area through which the car is moving.

3.3. Choice of Learning Model

As motivated by the solution requirement, current research identifies the LSTM-GBRT pair as the perfect choice for the learning model. No specific learning model is expected to offer considerably good results with raw derived data only (as mentioned in Table 1) having no correlation and weights defined. Hence, LSTM [12] could be employed to extract the features with their correlation and weight defined from the raw data presented to the model. Gradient Boosting Regression Trees or GBRT can use these extracted features to carry out the multi-class classification task.

LSTM has been reported [25] [22] to perform well on datasets having temporal characteristics such as time-series data. Hence, the choice of LSTM as a framework component supports the solution requirement criteria.

GBRT [19] uses an ensemble classification that improves the model's performance on a dataset with multiple features. The additive model approach in GBRT builds the decision trees iteratively and gradually builds the model by reducing the loss function. In this work, the boosting technique in GBRT has been employed to minimize the residual errors during training and thereby enhance the accuracy of the model during classification.

So, based on the above discussions, the use of LSTM and GBRT together would emerge as the most prominent components of the proposed solution. For the sake of completeness, preliminary ideas about these two learning models are described in subsequent texts.

3.4. Preliminaries of the Learning Models Used

This research intends to analyze time-series data that are received from the sensors. A Recurrent Neural Network (RNN) uses information about previous events to classify the current event. However, RNNs would struggle to create a model if there are long-term dependencies. The LSTM is designed to learn long-term dependencies by remembering the information for a long period of time. Hence, the current research uses LSTM for feature extraction and GBRT for achieving high robustness of ensemble learning for model training. The working principles of LSTM and GBRT are presented below.

3.4.1. Long Short-Term Memory networks (LSTM)

The LSTM model is a variant of the recurrent neural network (RNN) that uses a specialized memory unit to train the data through a time backpropagation algorithm. The LSTM is used to overcome the problem of traditional RNN because the traditional RNN can not remember long-term dependencies.

A cell, an input gate (i_t), a forget gate (f_t), and an output gate (o_t) make up a typical LSTM unit. Three gates regulate the information flow into and out of the cell, and the cell retains values for arbitrary time periods. Time series with indeterminate duration can be categorized, examined, and predicted with the LSTM algorithm. The gates control memory, whereas the cells store information.

Given an input sequence $X = \{x_1, x_2, x_3, \ldots, x_m\}$ fed to the LSTM, three gates (input, forget, and output gates) control the information stored in a memory cell. The input gate i_t chooses which input values should be applied to the memory change. It is decided whether to pass through 0 or 1 data using the sigmoid function (σ). The *tanh* function also adds weight to the supplied data, ranking its significance on a scale from -1 to 1. The input gate is defined as

$$i_t = \sigma([x_t, H_{t-1}] \cdot W_i + B_i) \tag{1}$$

The memory cell state (C_t) are defined as

$$C_t = tanh([x_t, H_{t-1}] \cdot W_c + B_c) \tag{2}$$

The forget state identifies the information that has to be taken out of the block. A sigmoid function makes the decision. It examines the previous state (H_{t-1}), the input (x_t), and each number in the cell state C_{t-1} to create a number between 0 and 1.

$$f_t = \sigma([x_t, H_{t-1}] \cdot W_f + B_f) \tag{3}$$

The output is determined by the input and memory of the block. It is decided whether to pass through 0 or 1 data using the sigmoid function. And which numbers can pass through 0 and 1 is determined by the tanh function. Additionally, the *tanh* function gives the input values weight by multiplying them by the sigmoid output to determine their importance on a scale from -1 to 1.

$$o_t = \sigma([x_t, H_{t-1}] \cdot W_o + B_o) \tag{4}$$

$$h_t = o_t \cdot tanh(C_t) \tag{5}$$

In all the above equations,

- σ is the sigmoid function
- U is the weight matrix of the input
- W is the weight matrix of state transition; and
- *tanh* is the hyperbolic tangent function

To compute the output of the current timestamp, the $SoftMax$ activation is applied to the hidden state H_t.

$$Output = SoftMax(H_t) \tag{6}$$

The maximum score in the output is considered to be the prediction.

3.4.2. Gradient Boosted Regression Trees (GBRT)

The GBRT model [19] is designed by combining different models for ensemble learning to achieve high accuracy. In the Boosting Learning Ensemble Framework, bagging, boosting, and stacking learning frameworks are used. The training process of the boosting framework consists of some steps; in each step, the base model is trained, and the training set is transformed for the next step as per the summary strategy. After completion of all the steps, the prediction results of all the base models are combined to obtain the final result. The derivation of the result is obtained as

$$F(x) = \sum_{i=1}^{m} h_i(x) \tag{7}$$

where $h_i(x)$ is the product of the base model and its weight.

The objective of the training is to predict the true value of y for $F(x)$. Using the greedy approach, at a time only one base model is trained, and $F(x)$ is updated in a certain number of iterations as follows:

$$F^i(x) = F^{i-1}(x) + h_i(x) \tag{8}$$

In the GBRT model, a data set $D = \{x_i, y_i\}$ with n examples and m features is taken as input where $|D| = n$, $x_i \in R^m$ and $y_i \in R$. Here, a tree ensemble model is used to predict the output using the K-additive function as follows:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad where \ \ f_k\Gamma \tag{9}$$

where $\Gamma = \{f(x) = w_{q(x)}\}$ $(q : R^m \Longrightarrow T, w \in R^T)$. In this model, the leaves of each regression tree contain a continuous score w_i. Here, T is the number of leaves in a tree, q is the structure of each tree containing example instances at leaf index, and each f_k is an independent tree structure.

4. Proposed Deep Learning Model

This section presents the data collection, and pre-processing techniques and the proposed LSTM-GBRT-based algorithms, in detail.

4.1. Dataset Curation and Preparation

The data is collected from the accelerometer and gyroscope sensors, along with the latitude and longitude of the location, by calling the Google Map API to mimic a natural environment. A vehicle continuously senses acceleration and gyroscopic data in the event of value change using the integrated hardware components of the smartphone.

Table 2. Prepared Dataset

Id	T	V_{id}	lat	long	$A_x(t)$	$A_y(t)$	$A_z(t)$	$\overline{A}(t)$	$\sigma_A^2(t)$	$G_x(t)$	$G_y(t)$	$G_z(t)$	$\overline{G}(t)$	$\sigma_G^2(t)$
1	1021200000	-	-	-	-	-	-	-	-	-	-	-	-	-
...														
n	1021200230	-	-	-	-	-	-	-	-	-	-	-	-	-

As the data is time-variant, the accelerometer $< A_x(t), A_y(t), A_z(t) >$ and gyroscope $< G_x(t), G_y(t), G_z(t) >$ data along with derived data are stored in the following format, as shown in Table 2. Here, t represents the timestamp, V_{id} is the vehicle ID, and ¡lat, $long$¿ represents the latitude and longitude of the location from which the instance is collected.

Due to the time-variant nature of the data, some features like mean and variance are generated using a window-based technique. In order to properly detect the drastic change in the mean and variance of the sensor data, the windows are overlapped. The mean of the values of all the axes of the accelerometer $\overline{A}(t)$ and gyroscope $\overline{G}(t)$ are calculated as follows:

$$\overline{A}(t) \leftarrow A_x(t) + A_y(t) + A_z(t) \tag{10}$$

and

$$\overline{G}(t) \leftarrow G_x(t) + G_y(t) + G_z(t) \tag{11}$$

For the time-series data analysis Simple Moving Average (SMA) of the accelerometer and gyroscope data are calculated as follows:

$$\overline{A_x^m}(t) \leftarrow \frac{1}{n} \sum_{u=t-\Delta t}^{t} A_x(t) \tag{12}$$

...

$$\overline{G_x^m}(t) \leftarrow \frac{1}{n} \sum_{u=t-\Delta t}^{t} G_x(t) \tag{13}$$

Here, the captured data is non-evenly spaced time series data with the independent variables $< A_x(t), A_y(t), A_z(t), G_x(t), G_y(t), G_z(t) >$. Moreover, in the captured data, the timestamps are not evenly spaced for the accelerometer as well as the gyroscope. Furthermore, the data captured from the accelerometer and gyroscope are not synchronized. Hence, using a windowing technique, the variance of the Accelerometer data ($\sigma_A(t)$) and gyroscopic data ($\sigma_G(t)$) are calculated as

$$\sigma_A^2(t) \leftarrow \frac{1}{n} \cdot \sum_{u=t-\Delta t}^{t} (A_x(u) - \overline{A_x})^2 + (A_y(u) - \overline{A_y})^2 + (A_z(u) - \overline{A_z})^2 \tag{14}$$

and

$$\sigma_G^2(t) \leftarrow \frac{1}{n} \cdot \sum_{u=t-\Delta t}^{t} (G_x(u) - \overline{G_x})^2 + (G_y(u) - \overline{G_y})^2 + (G_z(u) - \overline{G_z})^2 \tag{15}$$

where Δt is the window size in time and n is the number of instances within the interval, $\overline{A_x}$ is average of $A_x(t)$ over the time period $t - \Delta t$ to t and similarly others.

The standard Deviation of accelerometer and Gyroscope data are referred to as SD_{acc} and SD_{gyro} and they are computed using the following two equations:

$$SD_{acc} \leftarrow \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2} \tag{16}$$

where x_i is SD_{acc_x}, y_i is SD_{acc_y} and z_i is SD_{acc_z} respectively and n is number of instances.

and

$$SD_{gyro} \leftarrow \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2} \tag{17}$$

where x_i is SD_{gyro_x}, y_i is SD_{gyro_y} and z_i is SD_{gyro_z} respectively.

The G-Force values for the accelerometer and gyroForce are also computed as

$$G - Force \leftarrow \sqrt{x_{acc}^2 + y_{acc}^2 + z_{acc}^2}/9.81 \tag{18}$$

and

$$GyroForce \leftarrow \sqrt{x_{gyro}^2 + y_{gyro}^2 + z_{gyro}^2}/9.81 \tag{19}$$

All the original and derived data, as mentioned above, constitute the feature vector for the dataset. The dataset preparation technique thus discussed, can formally be presented as Algorithm 1.

Algorithm 1: Preprocessing and Data Preparation

input : Collected raw input Accalarometer and Gyroscope data as $A_x(t), A_y(t), A_z(t)$ and $G_x(t), G_y(t), G_z(t)$ respectively

output: Feature Vector V

1. calculate mean $\overline{A(t)}$ and $\overline{G(t)}$ of accelerometer and Gyroscopic data
2. calculate moving average $\overline{A_x^m}(t)$, $\overline{A_y^m}(t)$, $\overline{A_z^m}(t)$, $\overline{G_x^m}(t)$, $\overline{G_y^m}(t)$, $\overline{Z_z^m}(t)$ of Accelerometer and Gyroscopic data over the time period $t - \Delta t$ to t
3. calculate variance $\sigma_A^2(t)$ and $\sigma_G^2(t)$ of Accelerometer and Gyroscopic data over the time period $t - \Delta t$ to t
4. prepare the V using the raw and derived features presented in Table 2

Algorithm 1, as mentioned above constitutes a feature vector. In the present study, 4 different datasets have been prepared with 4 different data collection criteria. The data for the first dataset ds_0 was collected from an old car moving on a smooth and less crowded road having frequent speed breakers, the data for the second dataset ds_1 was collected on the same circuit but from a new car. Data for ds_2 and ds_3 were collected on a very rough surface and busy road with frequent speed breakers from an

old and a new car respectively. In all four data collection procedures, the skill level of the driver was ensured to be constant.
The sizes of the datasets under consideration are as follows. There are 17950, 10960, 14646 and 1116 instances in the ds_0, ds_1, ds_2 and ds_3 respectively. The above results in the preprocessed dataset size of $3.2MB$, $1.8MB$, $2.6MB$, and $228KB$ respectively.

During data collection, there were four classes identified as c_0 representing no accidents, c_1 representing minor accidents resulting in car body dents only, c_2 representing major accidents resulting abrupt seizing of the car movement or derailments and c_3 representing fatal accidents where the car got tossed up and down. The data pertaining to the event of no accident and minor accidents (member of class c_0 and c_1) were collected in real scenarios through old and new car in different conditions as described earlier. The data for major and fatal accidents (member of class c_2 and c_3) were collected from simulated environment.

The simulated environments were created by attaching smartphones with battery-operated LAEGENDARY Fast Remote Controlled Cars with the following specifications

- Model : Tornado 1:12
- Speed: 48 KM/Hour
- ESC / Servo Torque: 60A / 2.2 KG
- Car Dimension: 6.1 x 13.7x 11.2 in

The simulation environment was matched with the road and traffic conditions by designing artificial speed breakers and using six remote-controlled cars. The c_2 and c_3 data were equally divided in four datasets ds_0 to ds_3.

The Accelerometer and Gyroscope values resulted in the events of no accident and the other three types of accidents were clustered by running a standard $K - Means$ algorithm with $k = 4$ and labeled accordingly by running a Python program.

In the current study, the data has been generated by the sensors, where data retrieval is event-driven. Thus, the sensors do not work in a synchronized way, resulting in a considerable amount of missing values in the prepared dataset. As the data retrieval of a sensor is triggered by a value change event, the missing values in a dataset instance are filled with the values read at the prior clock.

4.2. Proposed LSTM-GBRT Model

As per the deep learning theoretical perspectives, LSTM generates and identifies the feature vectors on the temporal scale. The feature vectors as identified by the LSTM are not optimized and thus have the scope of readjusting the important factor through the introduction of newer weights by inspecting their correlation indices. Hence, in the proposed model, GBRT has been introduced as a boosting algorithm followed by the LSTM operations. GBRT computes the weight factors of the feature vectors as per its own algorithm for the betterment of the result and, this significant betterment

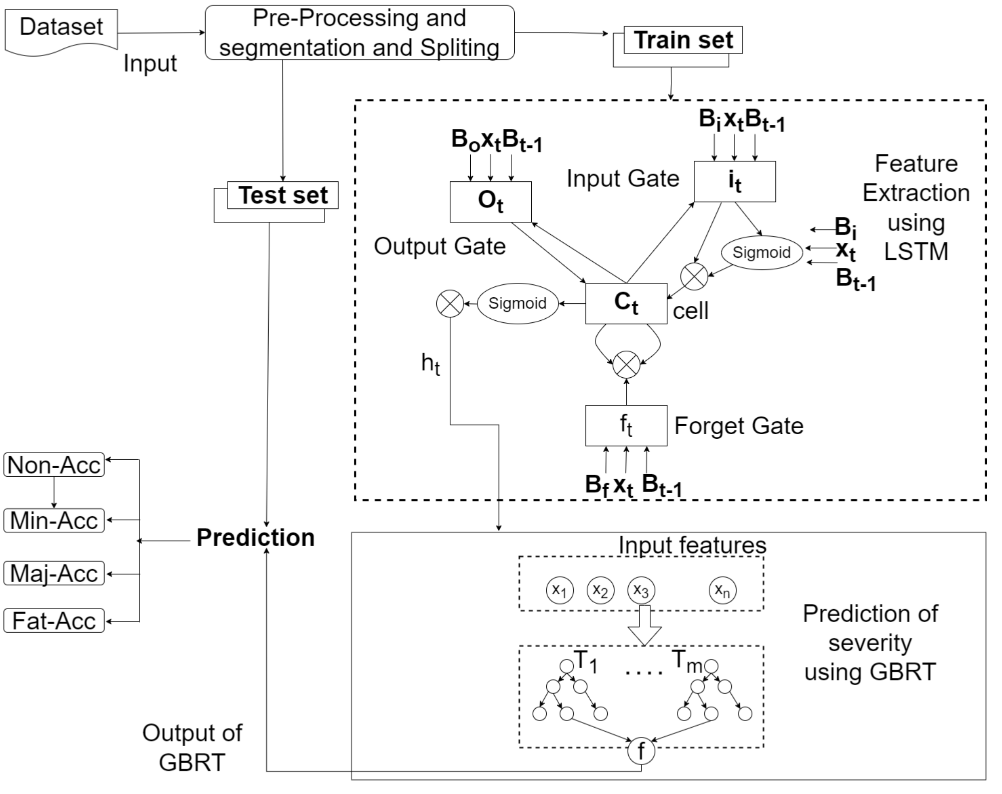


Figure 1. Schematic framework for severity analysis

in terms of several performance metrics has been demonstrated in the later section (subsection 5.4).

The proposed model is designed by combining LSTM and GBRT models as reasoned by the above text. The working principles of the proposed models are presented in section 3.4. The schematic diagram of the proposed model is shown in Fig. 1. Here, the LSTM model is used for feature extraction, and the extracted features are sent to the GBRT model as input. The curated dataset is split into Train and Test sets. The Train set is fed to the LSTM layer for uprooting the features from the supplied dataset. The output of the LSTM (h_t) is fed to the GBRT layer for discernment of severity that happens due to car accidentes. The proposed algorithm is presented in 2.

The number of nodes in the hidden layer cannot be precisely determined theoretically. To choose the number of hidden layer nodes, experiment with various hidden layer node counts using LSTM's single layer and gauge the degree of departure using the error rate and root mean square error. The 3-layer LSTM model has the best capacity to fit. The learning rate, the number of estimators, the maximum depth of the tree, the number of split nodes in the sample, the minimum sample needed for the

Algorithm 2: Feature extraction and severity discernment

input : X_t: instant input at LSTM layer,
w_i: weight of the corresponding cell,
c_t, c_{t-1}: fresh and prior condition of corresponding cell,
h_t, h_{t-1}: instant and former outcomes,
B_i: bias of the i^{th} node,
N: number of iterations,
M: number of predicted class

output: $F(x) = argmax < f^N(x), y^k >$

Pass h_{t-1}, X_t and bias information through sigmoid layer.
Sigmoid layer performs the input information (i_t) using Eq.1
i_t is fed to $tanh$ layer to get current moment information using Eq.2
Current and prior memory information is combined in Sigmoid layer using $c_t = f_t \cdot c_{t-1} + i_t \cdot C_t$, where C_t is computed as per Eq.2
The final output is fed to GBRT layer using Eq.5, where o_t is calculated by Eq.4
$x_i = h_t$ and y_i labelled as corresponding class in the input data
while $t < N$ **do**
  for $j = 1 \dots M-1$ **do**
    Compute $w_i \leftarrow \frac{1}{2}e^{-\frac{1}{2}<f^t(x_i),y_i>} \sum_{k=1}^{M}(y_i - y^k)e^{-\frac{1}{2}<f^t(x_i),y_i>}$
    Compute greatest risk decrease using $g* = \arg\max \sum_{i=1}^{N} g(x_i, w_i)$
    Compute optimal step size using $\alpha* = \arg\min R[f^t(x) + \alpha g*]$
    Update $f^{t+1}(x) = f^t(x) + \alpha * g * (x)$
    $t = t + 1$

leaf nodes, and the loss function are the hyperparameters of the GBRT layer. The values of the hyperparameters used in the proposed scheme have been presented in Table 3.

4.3. Adaptive and Dynamic Threshold Management

The proposed framework has been designed to support a large number of cars at a time and each car would emit sensory data at an average speed of 100 KB/sec for each sensor. If all the emitted data are sent (as client request) to the application server and then to the computing cluster, the infrastructure will become overloaded unnecessarily, yielding lower performance and throughput.

In order to restrict client requests, the proposed system ensures that the sensory data are compared locally against updated, dynamic, and adaptive threshold values stored as pushed data by the application server as guided by Algorithm 3. The algorithm uses GeoJSON object of type *LineString* as defined in [28]. The coordinates of the *LineString* starts from the current ¡$lat, long$¿ denoted as $Start_{x,y}$ and ends at a ¡$lat, long$¿ of a point 200 meters straight ahead, computed through Google Map

Table 3. Values of hyperparameters used in the proposed scheme

hyperparameters	**values**
Learning rate	0.03
LSTM layer depths	3
The depth of tree	4
Min sample leaves	3
Min child weight	1
Gamma	0
Scale position weight(GBRT)	1
Optimizer	Adam

API, denoted as $End_{x,y}$.

Spearman's rank correlation analysis of the four data sets ds_i, i = 0 to 3, having ordinality in different accident classes, reveals that derived data $F-Force$, SD_{acc} and SD_{gyro} have positively monotonic nature with accident severity classes c_0 to c_3, hence these three derived data are identified as the threshold limiter.

Further, the algorithm uses three constant values, preset by the proposed framework as

- k_{gf} - damping factor of threshold $G-Force$ in percentage value
- k_{acc} - damping factor of threshold SD_{acc} in percentage value
- k_{gyro} - damping factor of threshold SD_{gyro} in percentage value

Algorithm 3: Management of Adaptive and Dynamic Threshold of Sensory Data

input : Car Type V_T, GeoJSON object G_j and a constant triplet ¡$k_{gf}, k_{acc}, k_{gyro}$¿

output: Threshold Triplet¡$G-Force^h$, SD^h_{acc}, SD^h_{gyro}¿

1 Query MySQL cluster for $min(G-Force)$, $min(SD_{acc})$ and $min(SD_{gyro})$ given V_T and any ¡$lat, long$¿ between the range of $Start_{x,y}$ & $End_{x,y}$ of the *LineString* type G_j and accident class is c_1

2 Compute $G-Force^h \leftarrow min(G-Force) - (min(G-Force) \cdot (k_{gf}/100))$

3 Compute $SD^h_{acc} \leftarrow min(SD^h_{acc}) - (min(SD^h_{acc}) \cdot (k_{acc}/100))$

4 Compute $SD^h_{gyro} \leftarrow min(SD^h_{gyro}) - (min(SD^h_{gyro}) \cdot (k_{gyro}/100))$

5 Form the triplet ¡$G-Force^h$, SD^h_{acc}, SD^h_{gyro}¿

6 End

Algorithm 3 takes the stored information of all the c_1 data for a particular Geofenced area covering a range of ¡$lat, long$¿ values and a specific car type to compute a dynamic threshold of ¡$G-Force^h$, SD^h_{acc}, SD^h_{gyro}¿ triplet. Once the pre-processed sensory data at the client ends crosses this triplet value at time t_i, the sensory data and other derived data at time instance t_i are sent to the application server for processing.

Crossing of threshold triplet value at time t_i pertains to any of the following facts

- $G - Force(t_i) > G - Force^h$
- $SD_{acc}(t_i) > SD_{acc}^h$ and
- $SD_{gyro}(t_i) > SD_{gyro}^h$

When a new car type is introduced to the proposed model or a car is passing through a new geo-fenced location for which data is not available (because the threshold triplet does not exist), all the sensory data will be uploaded to the model for further processing. However, the adaptive nature of the algorithm ensures that as sensory data parameters keep being stored for accident events involving multiple car types across multiple locations, the unproductive client request will be restricted with time.

5. Results and Analysis

The current section details the experiments with the proposed model along with their multi-facet results and significant analysis.

5.1. Objective of the experiment

The objective of the experimental procedures with the proposed model can be formalized as

- To measure the accuracy of accident detection and its severity for each dataset as described in Subsection 4.1.
- To measure the performance metrics of the proposed LSTM-GBRT-based algorithm
- To consider the comparative analysis with the other AI classification models on accuracy and other performance criteria

Adaptive threshold management have not been taken into consideration due to the fact of their tremendous dependency on application level and infrastructural diversification. The experiment focuses only on the detection of accidents and severity analysis leaving the threshold management which actually means that the the experiment does not intend to balance the load on the deep learning model.

5.2. Methodology and Performance Parameters

Each of the labeled datasets as mentioned in Subsection 4.1 is divided into 80%-20% ratio to be considered as training and testing data respectively. The split is guided to ensure that 80% of data from each of the classes c_i where $i = 0 \cdots 3$ (as mentioned in Table 1), are considered in the training dataset. 70% of this training data gets consumed by the LSTM model and the rest 30% has been used in the GBRT model. The testing dataset elements are used to test the accuracy and effectiveness of the proposed system by applying standard performance measurement techniques as mentioned below.

In order to have a standard performance evaluation of the proposed classification algorithm and framework in general, the following primary metrics have been identified.

- **True Positive** (t_p): The number of testing data classified correctly in a particular class
- **True Negative** (t_n): The number of testing data not classified correctly in a particular class
- **False Positive** (f_p): The number of testing data classified incorrectly to a particular class, while in reality, it should not belong to that class
- **False Negative** (f_n): The number of testing data not classified incorrectly to a particular class, while in reality, it should belong to that class

Based on the above discussion, the current study formalizes the following derived performance parameters.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{20}$$

$$Precision = \frac{tp}{tp + fp} \tag{21}$$

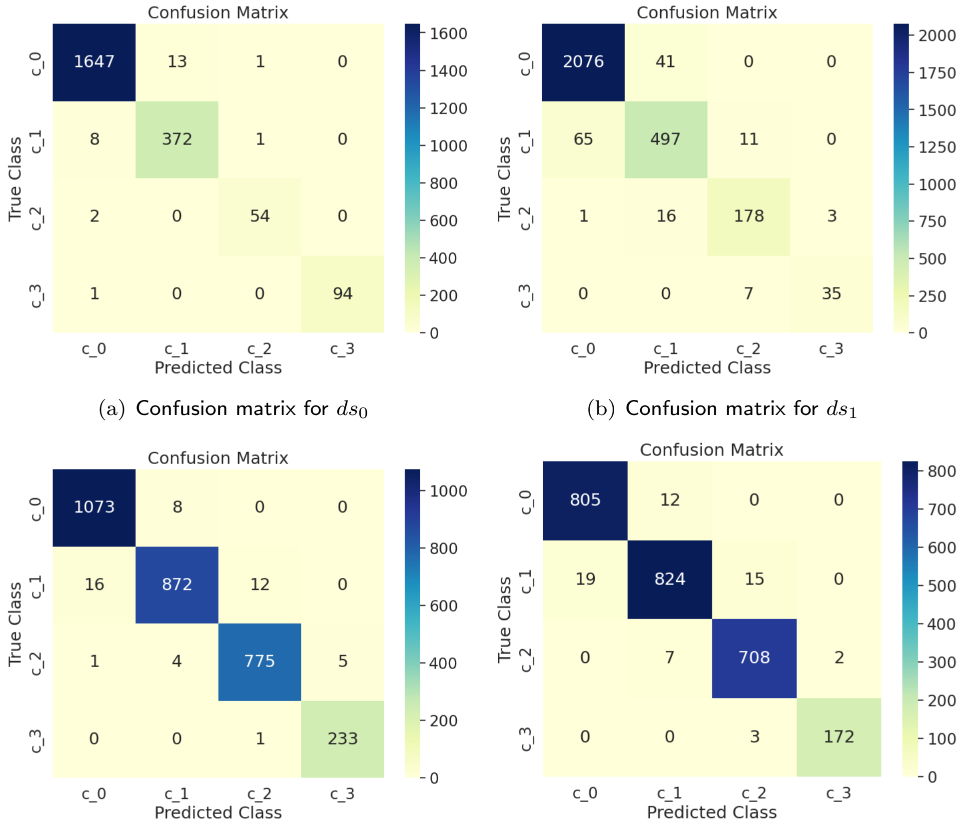
$$Recall = \frac{tp}{tp + tn} \tag{22}$$

$$F - Beta = \frac{(1 + \beta^2) * Precision * Recall}{\beta^2 * Precision + Recall} \tag{23}$$

$$G - mean = \sqrt{\Pi_{class1}^{class4} \frac{TP}{TP + FN}} \tag{24}$$

5.3. Experimental setup

The experiments on the proposed LSTM-GBRT classification are conducted on a Linux system having $x86\_64$ architecture, 2 GHz processor speed, along 13 GB of onboard RAM. Also, online GPU has been used along with Kaggle Notebook and TensorFlow for implementation purposes.



(a) Confusion matrix for ds_0

(b) Confusion matrix for ds_1

(c) Confusion matrix for ds_2

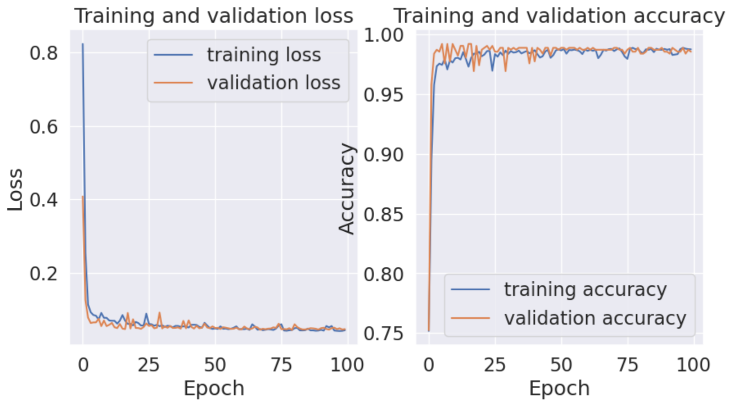
(d) Confusion matrix for ds_3

Figure 2. Confusion Matrices using Proposed LSTM+GBRT from Different Datasets

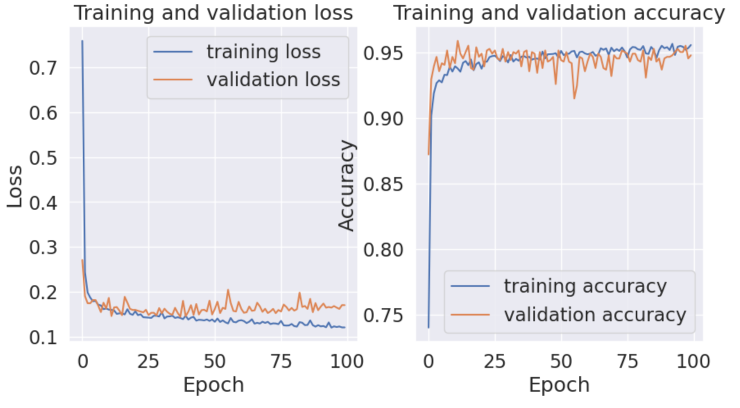
5.4. Experimental Results

The results of the experiment with the proposed framework are reported under two phases. The first phase presents the performance-related analytical reports of the proposed classification model, whereas the second phase illustrates its head-on comparison with the other classification models.

Analysis of the proposed framework: The proposed LSTM-GBRT classification algorithm was trained and tested with each of the datasets ds_i, $i = 0, \cdots, 3$. The data elements were classified into any one of the classes c_0, c_1, c_2, and c_3. Based on the classification, four confusion matrices have been generated and illustrated as Fig. 2. The interpretation of false positive and false negative cases using confusion matrices is very significant for severity prediction. The $(i, j)^{th}$ cell of the matrix represents the number of instances classified in class c_j, which actually should be classified in class



(a) Accuracy Measure of ds_0

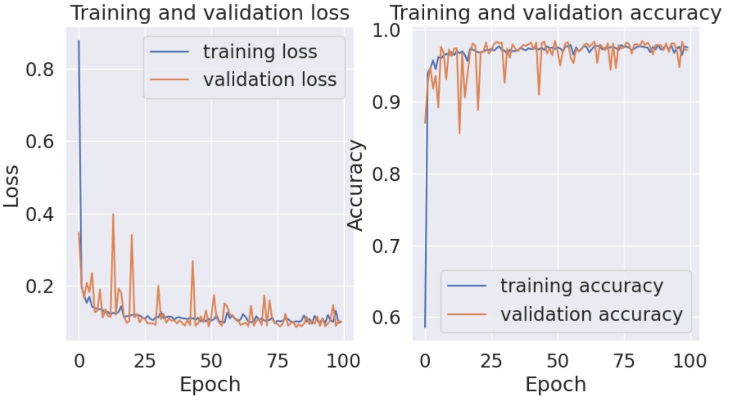


(b) Accuracy Measure of ds_1

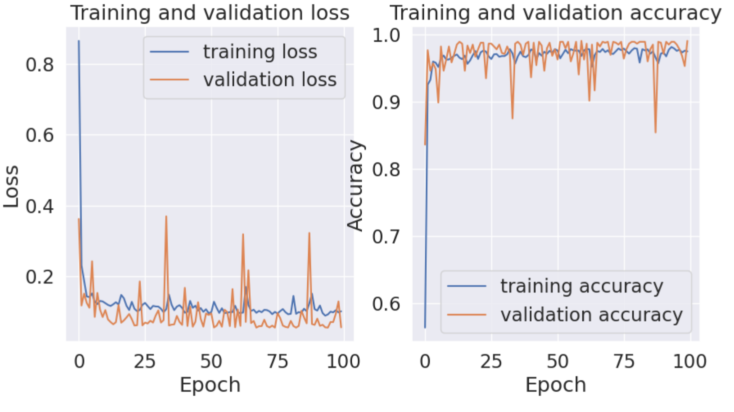
c_i. The proposed scheme has produced very few numbers of False negative cases. False-positive cases are quite hectic as they produce unnecessary chaotic situations by creating accidental severity messages to the victim's relatives.

Next, Fig. 3 shows the accuracy measures of the proposed LSTM-GBRT based algorithm running on each of the four datasets. The sub-figures of the Fig. 3 illustrate the distribution of training and validation loss and accuracy from left to right respectively. The training and validation accuracy curve represents whether the training has been performed appropriately and whether any probability of overfitting in this model exists or not. In the graphs (Fig. 3), it can be clearly seen that the training accuracy and validation accuracy almost overlap with each other after a certain number of Epochs for all the datasets. It proves the training has been performed perfectly for our model.

The average execution time of the proposed classification algorithm is measured in terms of the training and testing time of the models for the dataset as presented earlier. The average time shown (training and testing) per iteration takes all four datasets



(c) Accuracy Measure of ds_2



(d) Accuracy Measure of ds_3

Figure 3. Accuracy Measure of the Proposed LSTM+GBRT from Different Datasets

into consideration. It can be seen that, on average, the model spends around 140.56 milliseconds per iteration during training and only 14.23 milliseconds per classification for new unseen testing examples. This time computation has been reported for $14,833$ kb of training data and $3,708$ kb of testing data. However, the time taken during training and testing significantly varies with the increase in training and testing data volume.

In accident severity prediction models, prediction of accident severity is critical as the actions required for accident cases are different when compared with those of non-accident cases. Hence, along with Accuracy, more metrics, such as Precision, Recall, F-Beta, and G-mean are considered in this study. The impressive results of such metrics claim that the proposed model is very much comparative despite the use of imbalanced (time-series) data. Moreover, the intimation of false alarms to the victim's relative, and the emergency service providers such as hospitals, police

Table 4. Head-to-head Comparison of Proposed Model with other Classification Models

Model	Performance Metrics	Datasets			
		ds_0	ds_1	ds_2	ds_3
LSTM	Accuracy	90.33	81.59	89.21	86.84
	Precision	67.57	56.68	87.56	89.73
	Recall	63.72	53.28	78.66	83.89
	F-Beta	66.76	55.96	85.62	88.49
	G-mean	68.7	46.3	63.8	70
LSTM+RF	Accuracy	95.21	89.18	93.50	93.49
	Precision	70.21	58.55	94.20	94.51
	Recall	67.55	53.03	84.44	90.02
	F-Beta	69.66	57.36	92.07	93.58
	G-mean	75.6	45.6	64	80
LSTM+GBRT	Accuracy	98.81	95.09	98.43	97.74
	Precision	98.10	92.39	98.33	97.95
	Recall	98.04	89.51	98.61	97.90
	F-Beta	98.08	91.8	98.37	97.94
	G-mean	97.88	79.57	97.1	95.72

stations, etc., can create unnecessary panic and chaotic situations as well. So, to avoid the same, the proposed model has considered $\beta = 0.5$ to emphasize more on False positive cases. Thus the model is robust against alarm message generation by minimizing the False positive cases. Similarly, another performance metric, G-mean, has been introduced for the classification of imbalanced datasets. The higher value of the G-mean suggests that the proposed model has correctly classified the positive cases.

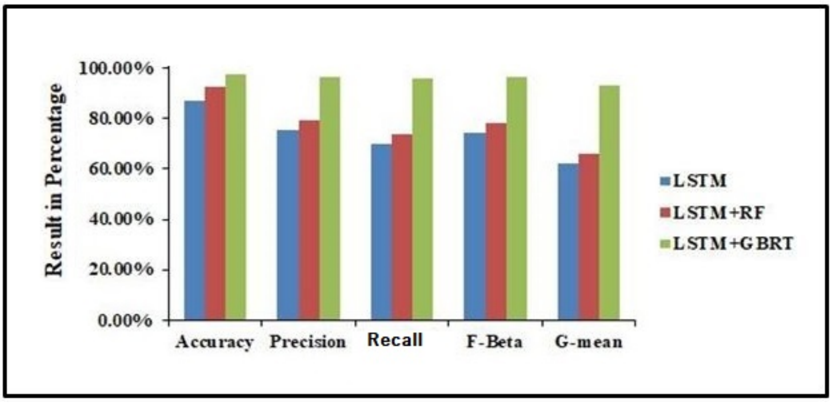


Figure 4. Average Performance Metrics using Different Classifiers

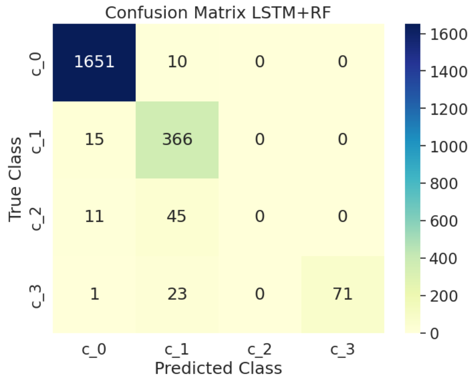
Comparative analysis: As stated in section 4.2, before experimenting with the proposed LSTM-GBRT model, the dataset has been tested with the LSTM model alone to realize the necessity of enhancement from LSTM to the combined model LSTM-GBRT. The comparative results in terms of all the performance metrics from four different time-series datasets using different classifier models such as LSTM, LSTM-RF (Random Forest), and LSTM-GBRT are depicted in Table 4. The results clearly state the fact that the LSTM classifier alone does not perform well enough in terms of all the performance metrics like Accuracy and others. Instead, LSTM when combined with other classifiers such as RF, GBRT produces much better results in terms of all parameters taken into consideration over here. It has been experimented to understand which classifier produces the best result with LSTM. However, the head-to-head comparison results of the combined model of LSTM-RF and LSTM-GBRT (Table 4) clearly indicate that the proposed classification model outperforms the other known model in every parametric criterion. Hence, it can be claimed that the current study has perfectly designed the best framework combining the LSTM and GBRT models for the detection of car accidents with severity measurements.

Fig. 4 represents the average performance metrics of the LSTM classifier, LSTM-RF, and proposed LSTM-GBRT obtained from Table 4, towards further comparative performance analysis among the models. All models have produced a good accuracy of more than 90% with four different time series datasets. However, the proposed model poses the highest. The maximum values of both Recall and Precision suggest that the model has correctly predicted the severity of cases. The high value of F-Beta suggests that the proposed scheme has efficiently reduced both false positive and false negative cases and thereby effectively predicts severe cases during car accidents.

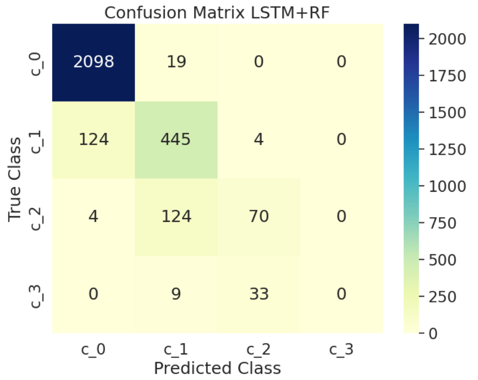
However, it is very evident from the results that on average, the proposed LSTM-GBRT portrays the best results in terms of all the performance criteria among all the classifiers under consideration.

For a better understanding of the results, the confusion matrices and the accuracy measure (training and validation), produced by RF with LSTM from all four datasets are represented in Fig. 5 and Fig. 6 respectively. The confusion matrices show that the false negative cases in the LSTM-RF model are higher compared to the same proposed LSTM-GBRT model. Additionally, the accuracy measure of the proposed model outperforms the same of LSTM-RF model as well.

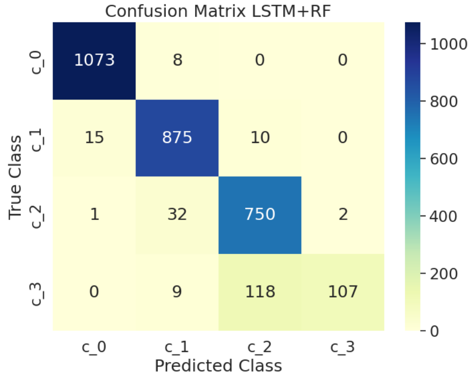
Comparison with Recent Studies: Similar works have been published in terms of severity discernment in car accidents so far. Very few among them have worked with time-series datasets. Therefore, in this context, we have collected and prepared four different time-series datasets to train the proposed LSTM-GBRT model. The model has been compared with other similar studies like a deep model that has been developed [23] for the prediction of severity cases in car accidents using modified particle swarm optimization. The deep model has produced a good accuracy of 84.63% along with decent results in other performance metrics. Similarly, another study [15] has suggested a deep model named stacked sparse autoencoder (SSAE) to predict injury severity in car accidents. The SSAE model has produced an accuracy of 81.2% in severity prediction discernment. On the other hand, our proposed LSTM-GBRT



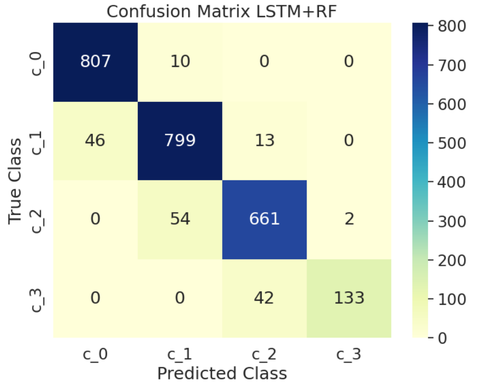
(a) Confusion matrix for ds_0



(b) Confusion matrix for ds_1



(c) Confusion matrix for ds_2



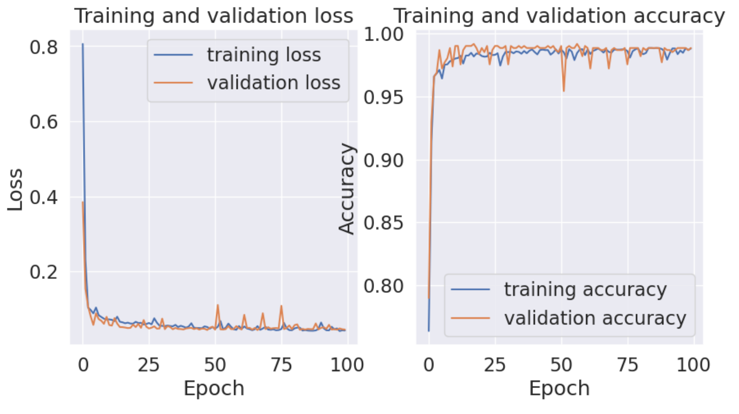
(d) Confusion matrix for ds_3

Figure 5. Confusion matrices using LSTM+RF from different datasets

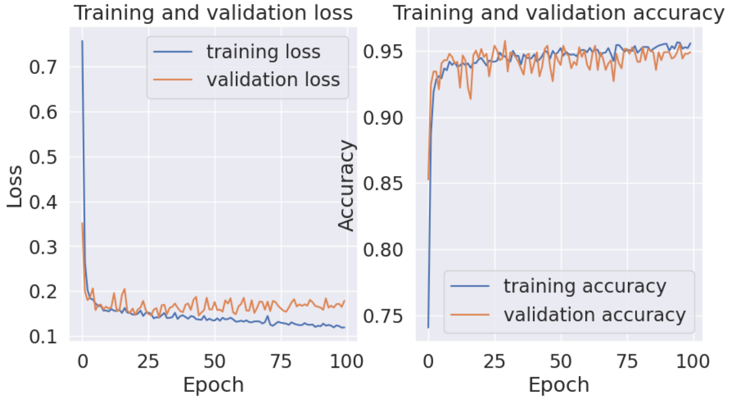
model has produced an accuracy of 98.7% along with good results in other performance metrics as well. Fig. 7 represents the comparative performance analysis of above mentioned two existing studies with the proposed scheme.
The work in [7] shows the accuracy measurement of predicting accident and non-accident cases on different data sets. It poses an accuracy of 86% in distinguishing no-accident from accident state. The proposed model distinguishes the states (no-accident, and accident with minor, major, and severe states) with 97.52% accuracy.

5.5. Insights gained

It is a noticeable fact from the performance results of the proposed LSTM-GBRT model that, it poses an accuracy of 97.52% on average, i.e. with only 2.5% loss in Training and Validation of the data. Additionally, the average Precision, Recall, F-

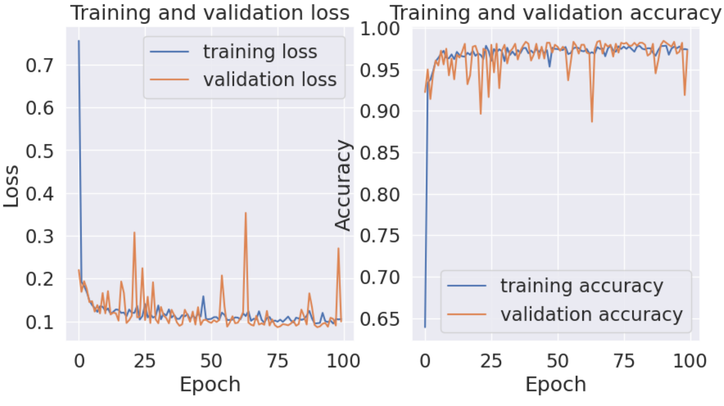


(a) Accuracy Measure of ds_0

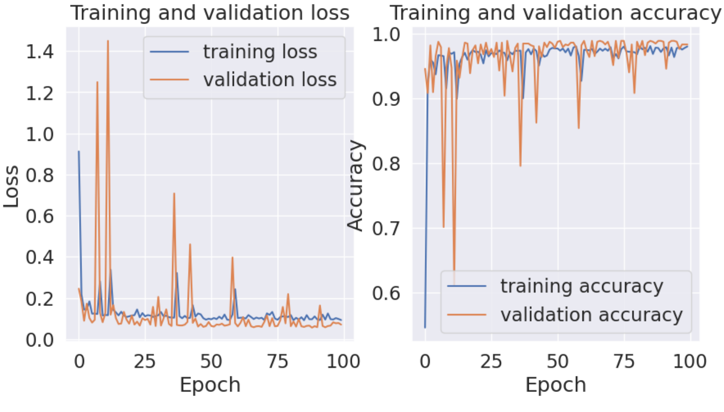


(b) Accuracy Measure of ds_1

Beta from all the datasets result as 96.69%, 96.02%, and 96.34% respectively. On being compared with other classification models, it has been found that the proposed model gains more than 10% accuracy over LSTM, and almost 5% over LSTM-RF model. In terms of other parameters such as Precision, Recall, F-Beta, and G-mean, the proposed LSTM-GBRT model outperforms the performance of other classifiers LSTM and LSTM-RF. Compared to LSTM and LSTM-RF, on average, the proposed model shows an improvement of 21%, 17% on Precision, 26%, 22% on Recall, 23%, 19% on F-Beta, and 30%, 28% on G-mean respectively. So, the reported results show that the LSTM-RF model on the time series data poses better results compared to the LSTM classifier model alone in identifying accidents and classifying the data in several severity classes. However, the proposed LSTM-GBRT model produces the best result in classifying such datasets in all the above performance criteria.



(c) Accuracy Measure of ds_2



(d) Accuracy Measure of ds_3

Figure 6. Accuracy Measure of the LSTM+RF from Different Datasets

5.6. Ablation Study

The proposed model uses LSTM for relevant feature extraction from time-series data and GBRT for severity classification purposes. The dataset used in this study is imbalanced. A small value is considered for both min child weight and scale position weight (Table 3) for faster convergence and gamma is set to 0 during the training phase with GBRT to minimize the loss reduction. On the other hand, instead of GBRT, the proposed model has been classified using a Machine learning technique, say Random Forest (RF). The extractor-classifier duo (LSTM-RF) produced slightly less accurate results during classification compared to LSTM-GBRT (Fig. 4) because RF consumes more time on training with imbalanced time-series data. If the proposed model ablated the classification mean (either RF or GBRT) then LSTM has alone produced a very less accurate result. The reason for the same could be attributed due

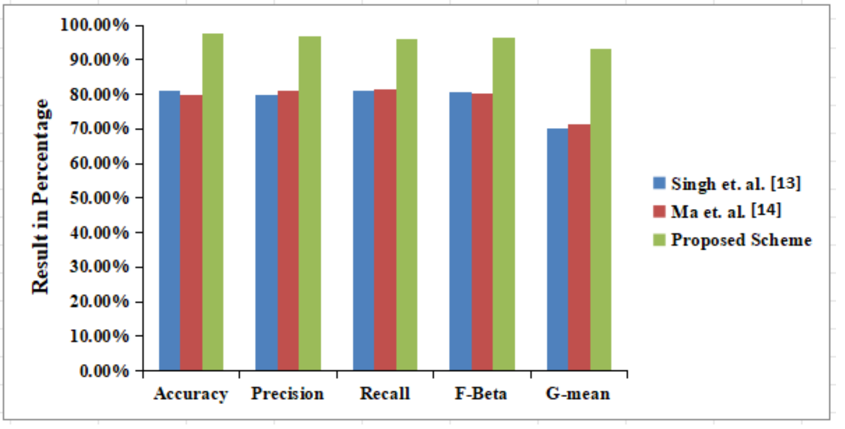


Figure 7. Comparative Performance Metrics of Different Studies

to the fact that the model has not utilized all the relevant features to get the output during classification. Moreover, it takes more time during training sessions. To be precise, the classification time taken by LSTM alone is 38.15 milliseconds (ms) and LSTM-RF is 23.35 ms; whereas the proposed LSTM-GBRT takes only 14.23 ms. In addition to that, due to the use of imbalanced time-series data, neither LSTM with RF nor LSTM alone has produced a better F-Beta score as well as G-mean compared to LSTM-GBRT.

6. Conclusion

The present study introduces a hybrid deep learning technique combining LSTM and GBRT models for developing an automated system for car accident detection and its severity analysis.

In this work, LSTM has been chosen as it performs well with time-variant data i.e. datasets having temporal characteristics. GBRT has been on the dataset after LSTM, to improve the model's performance as GBRT uses ensemble learning. So, the application of GBRT minimizes the residual errors during training. So, the use of the hybrid model LSTM-GBRT improves the accuracy during classification. The model analyzes the severity of the accidents depending on the time-series data continuously collected from the accelerometer and gyroscope sensors of the smartphone owned by the car's passenger. Multiple datasets have been prepared by collecting the sensory data via the mobile (application) placed inside a car. The performance of the model has been evaluated through extensive experiments on the datasets in terms of several performance metrics such as Accuracy, Precision, Recall, F-Beta, and G-mean. The results are compared with other existing models and the improvement proves that the proposed model outperforms other competing learning models.

There exist quite a few scopes of advancement of this proposed model where more attributes like, car health, road health, etc. could be taken into account in computing accident severity. While preparing the datasets, the current study fell short of working with real-time car accident data (members of c_2 and c_3) due to the unavailability of industry-level car crash reports from automobile car crash labs. Car accident data was prepared in a simulated environment with utmost care. In the future, the current study might be researched with industry-level car crash reports to investigate for any improvements. The proposed model can find its applications in road transport, road accident policy claims, urban transport management, etc.

References

[1] Road Traffic Injuries, World Health Organization, `https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries`, *accessed on Jan 02, 2024.*

[2] List of countries by traffic-related death rate, Wikipedia, `$https://en.wikipedia.org/wiki/List_of_countries_by_traffic-related_death_rate$`, *accessed on May 02, 2023.*

[3] Road Accidents in India 2021 Report, Ministry of Road Transport and Highways, Transport Research Wing, Govt. of India; `$https://morth.nic.in/sites/default/files/RA\_2021\_Compressed.pdf$`. *accessed on May 30, 2023.*

[4] India Records 1,55,622 Road Accident Deaths in 2022 Report; `$https://www.carandbike.com/news/india-records-1-55-622-road-accident-deaths-in-2022-3204563$`. *accessed on May 30, 2023.*

[5] Traffic Accidents Report Chapter 1A; `https://ncrb.gov.in/sites/default/files/adsi2020\_Chapter-1A-Traffic-Accidents.pdf`. *accessed May 30, 2023.*

[6] Delay causes most deaths: Neurosurgeons, Times of India Report, `$https://timesofindia.indiatimes.com/city/ranchi/delay-causes-most-deaths-neurosurgeons/articleshow/12818312.cms$`. *accessed May 30, 2023*

[7] Aloul, F. et. al.(2015). iBump: Smartphone application to detect car accidents. Computers & Electrical Engineering Journal, vol.43, pp. 66-75. doi: `10.1016/j.compeleceng.2015.03.003`

[8] Andersson, E., Berggren, Z. (2017). A Comparison Between MongoDB and MySQL Document Store Considering Performance. `$https://www.diva-portal.org/smash/get/diva2:1161166/FULLTEXT01.pdf$` *accessed on Feb 05, 2023*

[9] Antony, J., Jayaseelan, V., Olickal, J. J., Alexis, J., & Sakthivel, M. (2021). Time to reach health-care facility and hospital exit outcome among road traffic accident

victims attending a tertiary care hospital, Puducherry. Journal of education and health promotion, 10(429), pp. 1-6. doi: 10.4103/jehp.jehp_109_21

[10] Balfaqih, M. et. al.(2022). An Accident Detection and Classification System Using Internet of Things and deep Learning towards Smart City. Sustainability, 14(210). doi: 10.3390/su14010210

[11] Chang, W.J., Chen, L.B., Su, K.Y. (2019).Deepaccident: A Deep Learning-Based Internet of Vehicles System for Head-On and Single-Vehicle Accident Detection With Emergency Notification. IEEE Access, vol.7, pp. 148163-148175, doi: 10.1109/ACCESS.2019.2946468.

[12] Hochreiter S., Schmidhuber J., Long Short-Term Memory, in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi:10.1162/neco.1997.9.8.1735.

[13] Hozhabr Pour, H. et. al.(2022). A deep Learning Framework for Automated Accident Detection Based on Multimodal Sensors in Cars. Sensors, 22(10):3634. doi: 10.3390/s22103634

[14] Li, P., Abdel-Aty, M., Yuan, J. (2020). Real-time crash risk prediction on arterials based on LSTM-CNN, Accident Analysis, and Prevention Journal, vol. 135, pp.1-9. doi:10.1016/j.aap.2019.105371

[15] Ma Z, Mei G, Cuomo S.(2021). An analytic framework using deep learning for prediction of traffic accident injury severity based on contributing factors. Accident Analysis & Prevention Journal, vol. 160, pp. 106322. doi: 10.1016/j.aap.2021.106322

[16] Mannering, F. L., Shankar, V. and Bhat, C. R. (2016). Unobserved heterogeneity and the statistical analysis of highway accident data, Analytic Methods in Accident Research, vol. 11, pp. 1-16. doi: 10.1016/j.amar.2016.04.001

[17] Mannering, F. L., Bhat, C. R.(2014). Analytic methods in accident research: Methodological frontier and future directions. Analytic Methods in Accident Research, vol. 1, pp.1-22. doi: 10.1016/j.amar.2013.09.001

[18] Mishra S., Rajendran P. K., Vecchietti L. F., Har D., "Sensing Accident-Prone Features in Urban Scenes for Proactive Driving and Accident Prevention," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 9, pp. 9401-9414, Sept. 2023, doi: 10.1109/TITS.2023.3271395

[19] Prettenhofer, P., Louppe, G. (23 February 2014). Gradient Boosted Regression Trees in Scikit-Learn [Paper presentation]. PyData 2014, London, United Kingdom. https://hdl.handle.net/2268/163521.

[20] Roy, S., Kumari, A., Roy, P., Banerjee, R. (2020, September). An arduino based automatic accident detection and location communication system. In 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE) (pp. 38-43). IEEE. doi: 10.1109/ICCE50343.2020.9290701

[21] S-H Jo et. al.(2021). A Study on the Application of LSTM to Judge Bike Accidents for Inflating Wearable Airbags, Sensors, 2021, 21(19):6541. doi: 10.3390/s21196541

[22] Sarker, I.H.: Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. SN COMPUT. SCI. 2, 154 (2021). doi: 10.1007/s42979-021-00535-6

[23] Singh S., Yadav V.(2021). An Improved Particle Swarm Optimization for Prediction of Accident Severity. IJEER, vol. 9(3), pp. 42-7. doi: 10.37391/IJEER.090304

[24] Uguz, S. Buyukgokoglan, E.(2022).A Hybrid CNN-LSTM Model for Traffic Accident Frequency Forecasting During the Tourist Season, Tehnicki Vjesnik, vol.9(6), pp. 2083-2089, doi: 10.17559/TV-20220225141756.

[25] Yu, Y., Si, X., Hu, C., Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270. doi: 10.1162/neco_a_01199

[26] Zhang, C. (2021). Spatio-Temporal ConvLSTM for Crash Prediction: A unique deep learning approach for accident prediction, https://towardsdatascience.com/spatial-temporal-convlstm-for-crash-prediction-411909ed2cfa *accessed on April 23, 2023.*

[27] Zhang, Z., Yang, W., Wushour, S.(2020). Traffic Accident Prediction Based on LSTM-GBRT Model, Journal of Control Science and Engineering, vol.2020, Article ID. 4206919. doi: 10.1155/2020/4206919

[28] https://datatracker.ietf.org/doc/html/rfc7946

Received 17.12.2023, Accepted 17.06.2024