SMARTCOP – AN AUTOMATED PLATFORM TO MITIGATE THE IMPACT OF ROAD ACCIDENTS

2020 - 052

Dissanayake D.M.K.P.

(IT16138896)

BSc (Hons) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2020

SMARTCOP – AN AUTOMATED PLATFORM TO MITIGATE THE IMPACT OF ROAD ACCIDENTS

2020 – 052

Dissanayake D.M.K.P.

(IT16138896)

Dissertation Submitted in Partial Fulfillment of the Requirements for the Bachelor of Science (Hons) in Information Technology Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …… Date: …….

[Dissanayake D.M.K.P.]

The above candidate has carried out research for the bachelor's degree dissertation under my supervision.

Signature of the supervisor: …… Date: …….

[Dr. Windhya Rankothge]

Abstract

In recent years, there is a significant tendency of increasing road accidents both locally and globally causing many deaths, injuries, and fatalities. Especially, they directly or indirectly initiate tremendous problems socially and economically in every country. The irresponsibility of the public and the improper decisions of police officers majorly cause for road accidents. Even though several statistical solutions have been implemented to predict road accidents, still the results have not been satisfied by the department of police being the responsible party of the public. Consequently, an automated platform is an essential requirement to accurately predict road accidents and monitor the results. Hence, the SmartCop automated platform has been implemented to mitigate the impact of road accidents. As, the first major component of the system, road accident prediction has been developed including four subcomponents such as predict the accident's severity, reason, frequency, and vicinity. In order to develop four prediction models, both supervised and unsupervised learning algorithms are utilized. Thus, the implemented system aids with the police to observe prediction results via an automated user-friendly, and efficient web-based application. The results depict that the prediction productively enhances the accuracy of the system in order to mitigate the impact of road accidents.

Keywords: road accidents, supervised learning, unsupervised learning, web-based application

ACKNOWLEDGEMENT

I would like to offer my heartiest and sincere gratitude to my supervisor Dr. Windhya Rankothge for providing perpetual guidance, advices, and motivation with her immense knowledge and patience throughout the research. Also, I would like to express my earnest gratitude to my co-supervisor Ms. Narmada Gamage for her continuous support and guidance. I extend my appreciation to the Sri Lanka Institute of Information Technology (SLIIT) for granting an opportunity for undergraduates to evolve with a valuable research project and my evaluation panel who have provided new directions pointing out mistakes. Especially the support and the supervision given by the team of academic experts in the Comprehensive Design and Analysis Project (CDAP) module with Dr. Janaka Wijekoon in the head chair is treasurable to make the research a success.

A special thanks offer to Mr. Laksiri Geethal who is the former Senior Superintendent of Police (SSP) in Mirihana Police Station, Sri Lanka, and the staff at Mirihana Police Station. Also, I would like to thank Mr. Amila Kothalawala, who is the Solution Architect at Cybergate Services (Pvt) Ltd for supporting us to contact Mr. Laksiri Geethal by providing further directions.

Finally. I would like to thank my family and friends for their constant support encouragement and valuable advices. Also, a special thanks offer to all the people who have helped even from a word to make the SmartCop research project a successful one.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

Abbreviation	Description
API	Application Programming Interface
GUI	Graphical User Interface
IDE	Integrated Development Environment
KNN	K-Nearest Neighbors
ML	Machine Learning
MVC	Model-View-Controller
PC	Personal Computer
PCA	Principal Component Analysis
RFC	Random Forest Classifier
SL	Supervised Learning
UI	User Interface
USL	Unsupervised Learning
WHO	World Health Organization

LIST OF APPENDICES

1. INTRODUCTION

1.1 Background Literature

Road accidents have become one of the crucial obstacles that the world has encountered recently leading 1.35 million deaths approximately per each year globally according to the statistics on the World Health Organization (WHO) [1]. It further describes that 93% of fatalities from road accidents are reported from low- and middle-income countries regardless of having almost 60% of the world's vehicles. However, the consequences of road accidents not only lead to deaths and fatalities but also health issues of the victims which influence the social well-being [2]. In addition, rather than the direct participants of road accidents, their families suffer a lot economically and psychologically. Because severe fatalities occurred to the direct participants of road accidents, their occupations can be lost which may result in financial hardships and malfunctioning of the whole family. Besides, road accidents indirectly effect on country's economy. Accordingly, the respective governments of each country need to reserve a large number of their budgets to amend the consequences of road accidents and prevent them [3].

In addition, police also require a proper solution or a system to get assistance to reduce road accidents and their consequences as the protectors of the public. Also, it is difficult to identify specific areas and frequency of road accidents effectively and efficiently from their currently used manual process. However, many researchers have discovered various solutions using conventional and automated methods. Among them, road accident prediction process has become one of the emerging techniques to reduce road accidents all over the world. Many technical experts and academic scholars have involved in road accident prediction. Whether the technique is too technical or conventional, the researches have evolved in various road accident predictions. Even though several solutions have been found, still there is an immersing demand for accurate road accident prediction.

Why is the automated road accident prediction being more accurate and reliable than conventional methods?

Road accident prediction with conventional manual techniques contains many drawbacks. Especially, it is time consuming process when it comes to large datasets. When the dataset is getting larger, the required resources are getting increased. Also, the accuracy and reliability cannot be assured thoroughly since humans can make mistakes. Therefore, it is wise to engage with automated road accident prediction solutions that can handle large datasets or big data. Because they can operate and analyze big data efficiently using automated frameworks like the Hadoop framework resolving the difficulty of data imbalance [4].

In addition, it is hard to predict road accidents using multiple algorithms and compare the accuracies of them with traditional prediction procedures. Hence, automated predictions can be performed utilizing more than one algorithm [5]. In addition, the most effective algorithm for the prediction can be selected after comparing accuracies of them.

Nevertheless, automated road accidents can be helpful rather than manual statistics to be used by the traffic departments and the police in a more systematic way. Because they can predict patterns of accidents automatically that would happen in the future [6] and the government can grant precautionary measures to ensure road safety.

What are the related works conducted by various authors to predict road accidents?

Since the number of road accidents reported around the world is increasing, many types of researches have been conducted in order to mitigate the impact of them. Especially, every country which has a more distinguished tendency for many road accidents strives

to attain a solution for this prominent problem since human lives are the foremost wealth of any country.

According to the authors of [7], An Intelligent Road Crashes Avoidance (IRCA) system which adopts the Artificial Neural Network (ANN) and Decision Tree (DT) algorithms are proposed to predict car crash risk levels for 941 districts of the UK. However, the crash risk levels are classified into five levels of driving-automation that ranging from basic driver assistance in level 1 to full automation in level 5.

In addition, the research [8] has analyzed traffic accidents more deeply to determine the intensity of accidents by implementing four advanced SL algorithms of Machine Learning (ML). Those approaches are Decision Tree, K-Nearest Neighbors (KNN), Naïve Bayes, and Adaptive Boosting (AdaBoost). Also, the mentioned research has measured the accuracy of the proposed algorithms in order to gain the best results. The outcomes of this related work are the severity of accidents mentioned as fatal, grievous, and simple injury and monitoring the collision. However, some systems are technology biased in order to be embedded in vehicles.

In addition, the research [9] has analyzed and predicted the black spots, accident-prone zones, and road conditions using exploratory visualization techniques and regression analysis. It has limited their predictions for road accidents on state highways and ordinary district roads leaving the estimation of the severity into future work. In addition, the authors have considered the type of accident and the type of spot using the R tool for their estimations. Also, they have pointed out the data of road accidents with their frequencies and analyzed with correlation analysis and exploratory visualization approaches.

In addition, some road accident predictions are supported by the sensor technology also according to the authors of [10]. In there, a thorough analysis of the collision data is used to identify the leading causes of accidents, the accident-prone locations as well as to predict the conditions for collision occurrence. However, this paper implements a novel framework to predict collisions with the help of parameters like weather, geospatial information, and social events data that can be obtained by the sensor technology.

In addition, some work has predicted the number of road accidents and casualties [11] using genetic algorithm optimization of neural networks. This solution weights and approaches on the fundamental study of the traffic-related data. However, the prediction outcomes depict that the precision is larger than Backpropagation (BP) neural network. Further, the forecasting results speculate that the total number of deaths will be 139 thousand in 2010 and 167 thousand in 2020 in China because of road accidents.

The authors of [12] have implemented a new Geographic Information System (GIS) application to display and analyze road traffic crash data. Using the Critical Analysis and Reporting Environment (CARE) software, the outcomes have been displayed. Basically, this solution has mapped vehicle crashes. In addition, the project has conducted spatial analysis and "hot spot" identification to determine the route with the least crashes. However, this project has presented the solution by Obtaining statistical reports or analytics only.

Furthermore, as mentioned in the research [13], the authors have predicted black spots of accidents utilizing three features such as accident rate, accident frequency, and accident severity. Also, they have evaluated their predictions for the part of a pathway like highways. Here, only a section of the highway of 20.5km in the western province, Sri Lanka, was subdivided into approximately 200m segments and considered for this case study.

Nevertheless, as described in the research [14], it has predicted road accident patterns and fatalities occurred on pedestrians using data mining techniques. In addition, it dispenses perception into pedestrian accidents by discovering related patterns and their periodic underlying characteristics to obtain defensive measures and to assign resources for recognized problems. In order to obtain results, Decision Tree algorithms are utilized to a dataset of fatal accidents eventuated during the year 2010 in Great Britain. Also, the authors have used K-folds Cross-Validation methods to compute the unbiased estimate of the four prediction models comparing performances.

Likewise, many systems are existing to predict road accidents in the recent pool of literature. As described in the above, many researches have been conducted in order to mitigate the impact of road accidents using various technologies.

1.2 Research Gap

As the literature reviews, there exist various solutions currently to predict road accidents. As the research [4] depicts, most of the automated road accident predictions have been able to handle big data while maintaining accuracy. Hence, most of the automated predictions have been conducted based on large datasets. However, all the solutions that are mentioned in the literature have considered one perspective of road accident prediction. The authors of [8] have predicted the severity of accidents. Even though they have conducted the study very deeply and accurately, only one side of the accidents is covered and the predictions as the accident's vicinity cannot be identified with that. Also, there are some studies that only concern the predicted number of accidents or the casualties in the future. In addition, the authors of the research [14] have considered the fatalities occurred on pedestrians mainly.

Therefore, there is a high demand for a completed road accident prediction system that can cover most of the accident's aspects and predict the key factors of road accidents as accident's vicinity, frequency, severity, and reason via a single system. As the protectors of the public, the police also have a requirement of obtaining road accident predictions covering key factors that support their current work. Also, they need an automated system rather than manual statistics.

In order to fill that gap described before, the SmartCop automated road accident prediction component has been developed. It has the predicted road accident's severity, reasons, frequency, and vicinity. However, the prediction outcomes are processed and presented via the SmartCop web application including multiple automated features in which the target users are the Department of Police.

However, Table 1 depicts a comparison among the SmartCop road accident prediction component and the related research works which are existing in the literature. In addition, the below list shows the selected papers for the comparison.

- Paper [7] - Intelligent Road Crashes Avoidance System
- Paper [8] - Road Accident Analysis and Prediction of Accident Severity by Using Machine Learning in Bangladesh
- Paper [9] - Prediction of the cause of accident and accident prone location on roads using data mining techniques
- Paper [10] - A Framework for Collision Prediction Using Historical Accident Information and Real-time Sensor Data: A Case Study for the City of Ottawa
- Paper [11] - Macro Prediction Model of Road Traffic Accident Based on Neural Network and Genetic Algorithm
- Paper [13] - Development of Traffic Accident Prediction Models Using Traffic and Road Characteristics : A Case Study from Sri Lanka
- Paper [14] - Classifier prediction evaluation in modeling road traffic accident data
- SmartCop – The SmartCop road accident prediction component

Table 1: Comparison among SmartCop road accident prediction and existing works

Features	Paper [7]	Paper [8]	Paper [9]	Paper [10]	Paper [11]	Paper [13]	Paper [14]	Smart Cop
Predict the road accident's vicinity						✓		✓
Predict the road accident's frequency								✓
Predict road accidents on state highways			✓			✓		
Road accident prediction with sensor technology				✓				
Analyze fatalities occurred on pedestrians							✓	
Predict the severity of road accidents		✓						✓
Predict car crash risk levels	✓							
Speculate the reason for the accident								✓
Predict the number of casualties					✓			
Deliver the Results via user-friendly web-based application								✓

1.3 Research Problem

Recently, there is a significant increase in road accidents all around the world. Unfortunate deaths, injuries, and fatalities are major outcomes of road accidents influencing society as well as the economy. Therefore, passengers, drivers, and police officers suffer a lot. According to WHO data published in 2020 [1], speed, drunk driving, seat belt misuse, lack of awareness, and road infrastructure issues are identified as some of the main causes of road accidents. Also, most of the fatalities due to road accidents are reported from low and middle-income countries. Since Sri Lanka is also a developing country, there exists an enormous problem within the country because of road accidents.

As the protectors of the public, the Department of Police has a major requirement of reducing road accidents and their critical consequences. According to the police reports in Sri Lanka, the severity of the accidents is classified as fatal, serious, and slight. In addition, the details as reasons for road accidents, time slots, type of vehicles, and driver's condition of accidents are recorded in police reports. Furthermore, there are several statistical analyses which are conducted manually by relevant police stations, responsible authorities, and university students, but they were not able to produce accurate results as expected. Even though some automated and accurate road accident predictions have been conducted, most of them considered only a single type of prediction such as severity prediction or area clustering only. But there is a major requirement of a road accident prediction system with a complete set of predictions as accident's vicinity, frequency, reason, and severity. Since road accidents are a set of unexpected and undesirable occurrences, it is really complicated to find a pattern of road accidents using conventional primitive methods. Therefore, there is a tremendous requirement for an automated platform to mitigate the impact of road accidents from both public and the police perspectives. In addition, the department of police in Sri Lanka also requires an accurate system to monitor road accident prediction results as the responsible government agents of the public.

1.4 Research Objectives

1.4.1 Main objective

To implement a road accident prediction model more accurately including four forecasts of road crashes based on road accident historical data using Supervised Learning (SL) and Unsupervised Learning (USL) algorithms and deliver the outcomes of the predictions via a web-based application providing more systematic features that can be accessed by the police to obtain the predicted outcomes and their analysis.

1.4.2 Specific objectives

- To predict the severity of road accidents as slight, serious, and fatal
- To cluster the vicinities of road accidents in the suggested police region
- To predict the frequency (time slots) of road accidents and display accident peak times
- To speculate the reasons for road accidents
- To implement a separate function as road accident prediction component to display predicted information in SmartCop web-based application enhancing the best user experience
- To increase the user-friendliness of the SmartCop road accident prediction component including search options and enabling statistical visualizations

2. METHODOLOGY

2.1 Methodology

SmartCop is an automated platform that has been implemented to mitigate the impact of road accidents and the road accident prediction component is the first component of SmartCop research project. Since road accidents impact a lot in the society both economically and socially, police have a major requirement to diminish the consequences of the road accidents by obtaining more accurate predictions of them as the protectors of public. As a solution to their major requirement and to reduce the wastage of country's capital to settle the outcomes of road accidents because of imperfect manual accident analysis, the road accident prediction component has been developed as a main component of the SmartCop automated platform.

However, this component has been implemented providing predictions and classifications regarding road accidents in four aspects. They are accident's severity prediction, reasons classification, frequency prediction, and specific vicinities clustering which cover the overall prediction process. However, these predictions support the police to mitigate the impact of road accidents covering the main objective of this research component. The results of each prediction are distributed via a web-based application called SmartCop. Hence, the target customer of this SmartCop web application is the Department of Police where each police station has its own authentication to access the application. Also, the police officers who are authorized to access the application can view the details of these predictions and statistics. The Figure 1 shows the high-level system diagram for the overall road accident prediction process. As the diagram describes, the first step is data collection where the system consumes the historical data related to road accidents. Then data preprocessing is performed to obtain cleaned and normalized data. After that, the most suitable features are selected using feature importance technique and manual processes such as referring to related works. Then the road accident prediction function gets started such as accident's severity prediction, reasons prediction, frequency

prediction and vicinity clustering. The predicted results and other required information are stored in a database and fetched by the SmartCop Graphical User Interface (GUI) of the front-end application with their statistical analysis and other functionalities.

In order to obtain the most accurate results, some certain extremities in models are considered thoroughly.

- Accuracy
- Performance
- Efficiency
- Effectiveness
- Security

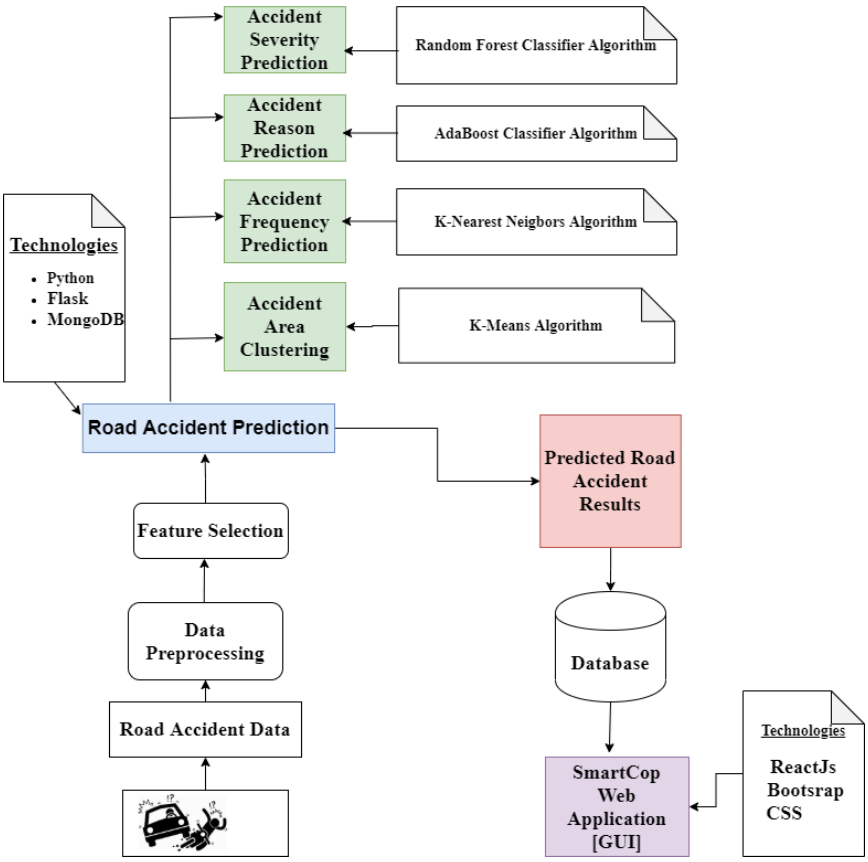

Figure 1: Road accident prediction system diagram

2.1.1 Feasibility study

Since the SmartCop web application has been developed as a solution to problems gained when analyzing accident hotspots and identifying accident frequencies based on police officers' experiences and manual statistical analysis, it is a huge requirement accomplishment from police department side.

Hence, the development and maintenance of the application is identified as feasible after considering the resources, development, time, and cost for the whole project. Though the hosting of the application costs some amount, the initial features of the application are not charged but there will be additional charges for new modifications and extra features. Hence, the budget is considerably acceptable. At the initial phase, the system will be delivered to the Department of the Police targeting the police stations. However, the system implementation deadline is also feasible according to the project plan.

In addition, the dataset and other basic information resources are easy to find and data collection is really feasible and the required development skills are manageable. Also, the tools and technologies used for the application development are freely available at the initial phase.

- PyCharm
- MongoDB Atlas
- Visual Studio Code

2.1.2 Functions

- Register to the application: Sign Up after providing the requested details and create a user profile
- Sign In process
- Manipulate the created user profile

- Input area in a relevant police region and view predicted number of accidents with their frequencies
- View Accident hotspots
- View predicted severities and reasons for accidents
- View accident peak time
- Obtain summary reports of each prediction

2.1.3 Application literacy

The main language used for SmartCop web application and the road accident prediction function is English. But Sinhala and Tamil languages will be available in future versions. Since the application is web-based, there should be a proper internet connection to access the application. Hence, the internet connection is a basic requirement to fetch data from severs and access relevant Application Programming Interface (API) to maintain the functionality of the application. Since police officers access the application, they should have a proper knowledge of their region and environment.

2.1.4 Required resources

Software requirements

1. **PyCharm**

PyCharm is an Integrated Development Environment (IDE) utilized in computer programming especially for the Python programming language which is developed by JetBrains. It is more popular because of its productive and smart assistance to develop software with almost all needed tools [15]. However, it facilitates a graphical debugger, code analysis, integrated unit tester, integration facility with version control systems, and assist Anaconda Navigator too. However, PyCharm has been used for the backend development in the SmartCop project.

2. Visual Studio Code (VS Code)

Visual Studio Code is developed by Microsoft and available for Windows, Linux, and macOS. It collaborates the lucidity of a source code editor with robust developer tools as IntelliSense code completion and debugging [16]. Basically, it is an editor and handle edit-build-debug cycle with less time in the current environment providing much time to implement the developer's ideas. However, VS Code has been used for frontend development of the project.

3. Python

Python is an object-oriented, high-level and general-purpose programming language [17]. This can be used for a wide variety of applications. It consolidates modules and packages, exceptions, dynamic data types, and classes. The Python interpreter and the substantial standard library are accessible in source or binary and freely distributed. However, Python 3.8 version has been used for the backend implementation of this project.

4. Flask

Flask is known as a micro web framework which is written in Python. It does not need certain tools and libraries. Also, it functions without a database abstraction part, form validation, and other components like pre-existing third-party libraries which provide basic and common functions. Also, flask extensions are developed for object-relational mappers, many authentication technologies, form validation, and common framework related tools [18]. Therefore, Flask is used as the server-side Python framework in the SmartCop road accident prediction component.

5. **MongoDB Atlas**

MongoDB Atlas is a cloud databased which is fully managed and developed by MongoDB. It manages the complexities of deployments [19]. As the database utilized in the project is MongoDB, the main storage unit is MongoDB Atlas Cloud.

6. **ReactJS**

ReactJS is a JavaScript library which is developed by the Facebook as an open-source project. React is not a framework and it is used to implement the View layer in Model-View-Controller (MVC) architectural pattern [20]. Therefore, it has been utilized as the main frontend programming language in the SmartCop web application also.

7. **HTML and CSS**

HTML and CSS technologies are known as frontend technologies. They have been utilized to create documents in the World Wide Web relevant to the SmartCop web application.

8. **Bootstrap4**

Frontend UI design is supported by the Bootstrap4 which is the latest version. It is the most demanded HTML, CSS, and JavaScript framework to implement efficient and responsive websites. It is free to be utilized.

9. **Anaconda Navigator and Jupyter Notebook**

Anaconda Navigator is a desktop or GUI launched by the Anaconda distribution that facilitates to launch software solutions [21]. Also, instead of using commands in a

command line, it flexibly handles the conda packages, channels, and environments. However, the navigator is available for both Windows, macOS, and Linux. In addition, Jupyter Notebook is a software application which can be navigated via the Anaconda Navigator and it has been utilized to implement and test road accident prediction models in SmartCop project.

Hardware requirements

- A server machine with high processing power for backend
- Any Personal Computer (PC) or laptop with proper internet connection

Memory requirements

Since the SmartCop web application needs ML and it has large amount of data, some higher memory is required for backend but that much memory is not required for the frontend.

Back-end (Python)

- RAM - 8 GB minimum, 16 GB or higher is recommended
- Data Storage – Maximum of 2GB

Front-end (ReactJS)

- RAM – 1GB - 2GB
- Data Storage – 500 MB or higher

2.2 Commercialization Aspects of the Product

Commercialization is the process of introducing novel products or services to the market by the definition itself [22]. Therefore, when considering the road accident prediction component that is the first component of the SmartCop platform, it states the commercialization aspect which promotes the quality production, distribution, and customer support. Especially, road accident prediction component is highly innovative because of its prediction depth. Usually, a system may perform a single type of road accident prediction as severity prediction only or the accident's hotspots prediction only but the SmartCop platform provides four accident predictions through a one system such as accident's severity prediction, reason classification, frequency prediction, and accident's vicinities clustering. Hence, the user can observe the number of accidents in a relevant area with accident frequencies along with their severities and reasons. Also, the user can identify accident hotspots in a relevant region. More importantly, the end users of this SmartCop web application along with the road accident prediction component is the police, therefore the system is handed over to the protectors who have the authority to take accidents mitigation steps. Such that, this component is highly innovative which improves the commercial value.

As a result, it reduces the damages for common infrastructures, vehicles, and other physical elements which supports the country's economy enormously. Especially, this platform saves human lives which is a major concern from the society point of view. Indirectly, it protects the people with strength to work for the country and enhance the economy. Moreover, the system provides facilities such as

- Less time consuming than the manual analysis since all four predictions are conducted as automated solutions
- All four predictions are conducted in a single platform
- Ready-made datasets with more accurate information
- Less technical knowledge is required for end-users to utilize features
- Predictions are not limited to one police region and it can be extended Island wide

As per the commercialization aspect, the SmartCop system can be served as set of packages with premium features. In this initial package release, SmartCop road accident prediction component provides functions as view predicted number of accidents with their frequencies relevant to the searched area in a relevant police region, view accident hotspots, and view predicted severities and reasons for accidents. Thereafter, functionalities like identification of accident peak times, and more statistical features relevant to the predictions can be offered as features of the premium package. However, SmartCop web application contains a main dashboard for the road accident prediction component which displays current news and information which is essential for the police. Therefore, some advertisements regarding road accidents mitigation tricks, economical help from accident mitigation, and other relevant details can be displayed along with the premium package also. In future, new, and extended features can be introduced along with premium package to be used on payment basis. Also, it will add a value to the initial SmartCop platform.

2.3 Testing & Implementation

2.3.1 Testing

Testing phase is a critical task in the software development like cycle. If the testing process contains mistakes, it could cause for system failure even. Hence, testing is an immense requirement in software development since it reveals bugs or defects prior to the software delivery guarantying its quality [23]. Also, perfectly tested application confirms the reliability, quality, and high-performance of a software application. In this implementation, there are several testing phases such as design testing, unit testing, module testing, component and integration testing, and user acceptance testing. Since road accident prediction component is a sub part of the SmartCop software application, system testing is conducted after combining all sub parts of the application.

- Design testing
    - Design testing has been conducted during the design phase of road accident prediction component. It is tested whether the required functional requirements are included in the component design. Also, the required UI designs are tested whether they fulfill the best user experience. In addition, the flow of UIs, and user controls are tested during the design testing phase.

- Unit testing
    - Unit testing is conducted after implementing a piece of code unit or a module in the component and performed individually. Main goal of this testing is to ensure whether the implemented piece of software unit is working properly. If there are bugs, they were fixed at the same phase.

- Module testing
    - In module testing, certain subcomponents of the road accident prediction component are tested. In addition, classes procedures, functionalities, and connection controls are tested in here. Main modules include the severity prediction, reason classification, frequency prediction, and accident vicinity clustering.

- Component and integration testing
    - Whole road accident prediction component is tested after integrating all subcomponents including four prediction functionalities mainly such as road accident's severity prediction, reason prediction, frequency prediction, and vicinity clustering. If there are bugs found, they were fixed

in the same testing phase. Especially, the integrated component's functionality is tested to ensure whether it is working as expected.

- User acceptance testing
    - In this testing phase, the whole component is tested whether it fulfills the business value of the product. Main goal is to satisfy the customer requirements and their acceptance level towards the working component.

Some test cases which are performed have been described below. Here, test cases implementation is divided into two sections.

- Frontend test cases (Described using Table 2 - 5)
- Backend test cases (Described using Table 6)

Frontend testing

Table 2: Test Case 001

Test case ID	001
Test case scenario	Test for user sign in with valid credentials
Test steps	Enter valid email Enter valid password Click on submit button
Test data	Username = officer001@gmail.com Password = officer@123
Expected results	Sign-in to the application successfully
Actual results	Sign-in to the application successfully
Pass/ fail status	Pass

Table 3: Test Case 002

Test case ID	002
Test case scenario	Test for user sign in with invalid credentials
Test steps	Enter invalid email Enter invalid password Click on submit button
Test data	Username = officer001@gmail.com Password = officer@111
Expected results	Prompts the error message "Invalid Credentials"
Actual results	Prompts the error message "Invalid Credentials"
Pass/ fail status	Fail

Table 4: Test Case 003

Test case ID	003
Test case scenario	User searches for a specific area to observe number of accidents in the area
Test steps	User navigates to home page Search for an area in the search bar Click on search icon
Test data	Region 202
Expected results	Displays number of predicted accidents in the searched area
Actual results	Displays number of predicted accidents in the searched area
Pass/ fail status	Pass

Table 5: Test Case 004

Test case ID	004
Test case scenario	User searches for a non-existing area to observe number of accidents
Test steps	User navigates to home page Search for non-existing area in the search bar Click on search icon
Test data	Region 2
Expected results	Error message “Please select an existing area”
Actual results	Error message “Please select an existing area”
Pass/ fail status	Fail

Backend testing

In the backend, four main predictions were conducted mainly, and test results are saved in MongoDB database and fetched to the frontend via APIs.

Table 6: Test Case 005

Test case ID	005
Test case scenario	User checks for accident frequency prediction details
Test steps	User navigates to home page Search for an existing area Click on frequency prediction button

Test data	Region 202
Expected results	Display accident frequencies relevant to the 4 time zones
Actual results	Display accident frequencies relevant to the 4 time zones
Pass/ fail status	Pass

2.3.2 Implementation

The road accident prediction component has been implemented as the first component of the SmartCop web application. Therefore, the backend implementation is conducted using the Python programming language and Flask as the server-side Python framework. Also, the MongoDB has been utilized as the database to store predicted outcomes, authentication details, log history, and main functional data. However, the frontend has been implemented using ReactJS and Bootsrap4. In addition, HTML and CSS technologies have been used to create documents in the World Wide Web. Since the target user of the SmartCop web application is the police, the UIs are generated according to the best user experience with user friendly features. Also, four models of four predictions are implemented using ML via the Python programming language. In addition, Jupyter Notebook has been utilized for model generation process in the initial stage also. Moreover, RESTful APIs have been implemented for the communication between server and the application.

Model implementation

Data collection

The dataset which has been utilized for the prediction is available online [24]. Therefore, the collected dataset contains data regarding accidents and vehicle details which encountered the relevant accidents from year 2012 to 2017. The number of records in the dataset is 830,719. Also, major reasons for road accidents, their categorization details, and data relevant to road accidents peak times are collected from the Mirihana Police Station in Sri Lanka.

Data preprocessing

Since the original dataset contained many zero and null values, data preprocessing has been accomplished. According to that process, it was checked the existence of all null and zero values as in the code snippet displayed in Figure 2 and null values in each column for more confirmation as mentioned in Figure 3.

```
#Checking for sum of all null values
sum_of_nulls = sum(acc_data.isnull().sum())
print(sum_of_nulls)
```

Figure 2: Checking for sum of all null values

```
#Checking for missing values in each column
count_of_nulls_each_column = acc_data.isnull().sum()
print(count_of_nulls_each_column)
```

Figure 3: Checking for null values in each column

After clarifying null and zero values in the dataset, they were eliminated, and the dataset was cleaned as mentioned in Figure 4.

```
#remove columns with many null values
accident_df2=acc_data[acc_data.columns[acc_data.isnull().mean() < 0.8]]
#remove rows with null values
accident_df3 = accident_df2.dropna(how='any',axis=0)
print(accident_df3.shape)
print(accident_df3.head())
```

Figure 4: Eliminate null and zero values from dataset

Since the dataset contains many categorical variables, it needs to be normalized in order to conduct classifications using ML. As displayed in Figure 5, the dataset has been normalized using Label Encoding via Pandas Python libraries.

```
#label encoding
lbl_enc = LabelEncoder()
tf_acc_data = accident_df3.copy()
for i in accident_df3.columns:
    tf_acc_data[i]=lbl_enc.fit_transform(accident_df3[i])
```

Figure 5: Normalize data using label encoding

Feature selection

Feature selection immensely impacts on the performance of any machine learning model and reduces overfitting, training time, and improves the accuracy[25]. Since the original dataset contained 52 features as displayed in Figure 6 and Figure 7, a feature selection process has been conducted before each prediction type.

```
Age_Band_of_Driver
Age_of_Vehicle
Driver_Home_Area_Type
Driver_IMD_Decile
Engine_Capacity_.CC.
Hit_Object_in_Carriageway
Hit_Object_off_Carriageway
Journey_Purpose_of_Driver
Junction_Location
make
model
Propulsion_Code
Sex_of_Driver
Skidding_and_Overturning
Towing_and_Articulation
Vehicle_Leaving_Carriageway
Vehicle_Manoeuvre
Vehicle_Reference
Vehicle_Type
Was_Vehicle_Left_Hand_Drive
X1st_Point_of_Impact
Year_x
1st_Road_Class
1st_Road_Number
2nd_Road_Class
2nd_Road_Number
Accident_Severity
Carriageway_Hazards
Date
```

Figure 6: Original features set 1

```
Day_of_Week
Did_Police_Officer_Attend_Scene_of_Accident
Junction_Control
Junction_Detail
Latitude
Light_Conditions
Local_Authority_(District)
Local_Authority_(Highway)
Location_Easting_OSGR
Location_Northing_OSGR
Longitude
LSOA_of_Accident_Location
Number_of_Casualties
Number_of_Vehicles
Police_Force
Road_Surface_Conditions
Road_Type
Special_Conditions_at_Site
Speed_limit
Time
Urban_or_Rural_Area
Weather_Conditions
Year_y
```

Figure 7: Original features set 2

In order to conduct road accident severity prediction, 21 features are selected manually as of Figure 8 after referring related works [8, 26] which have predicted the severity of the accidents. After that, feature importance has been conducted to obtain the best matching features set and maximize the accuracy of the severity prediction as possible which has selected six features such as date, age of vehicle, make, time, longitude, and latitude as the most important features. The output of feature selection results is shown as a numerical format in Figure 9 and graphical format in Figure 10.

```
Latitude
Longitude
Urban_or_Rural_Area
1st_Road_Class
Speed_limit
Road_Type
Road_Surface_Conditions
Weather_Conditions
Light_Conditions
Date
Age_Band_of_Driver
Age_of_Vehicle
Junction_Detail
Junction_Location
X1st_Point_of_Impact
make
Vehicle_Type
Vehicle_Manoeuvre
Did_Police_Officer_Attend_Scene_of_Accident
Time
reasons
```

Figure 8: Features set used for severity prediction

```
Variable: Date                 Importance: 0.12
Variable: Age_of_Vehicle       Importance: 0.12
Variable: make                 Importance: 0.12
Variable: Time                 Importance: 0.12
Variable: Latitude             Importance: 0.08
Variable: Longitude            Importance: 0.08
Variable: Age_Band_of_Driver   Importance: 0.05
Variable: reasons              Importance: 0.05
Variable: Vehicle_Manoeuvre    Importance: 0.04
Variable: 1st_Road_Class       Importance: 0.03
Variable: X1st_Point_of_Impact Importance: 0.03
Variable: Vehicle_Type         Importance: 0.03
Variable: Weather_Conditions   Importance: 0.02
Variable: Junction_Detail      Importance: 0.02
Variable: Junction_Location    Importance: 0.02
Variable: Did_Police_Officer_Attend_Scene_of_Accident Importance: 0.02
Variable: Urban_or_Rural_Area  Importance: 0.01
Variable: Speed_limit          Importance: 0.01
Variable: Road_Type            Importance: 0.01
Variable: Road_Surface_Conditions Importance: 0.01
Variable: Light_Conditions     Importance: 0.01
```

Figure 9: Feature importance for severity prediction in numeric format

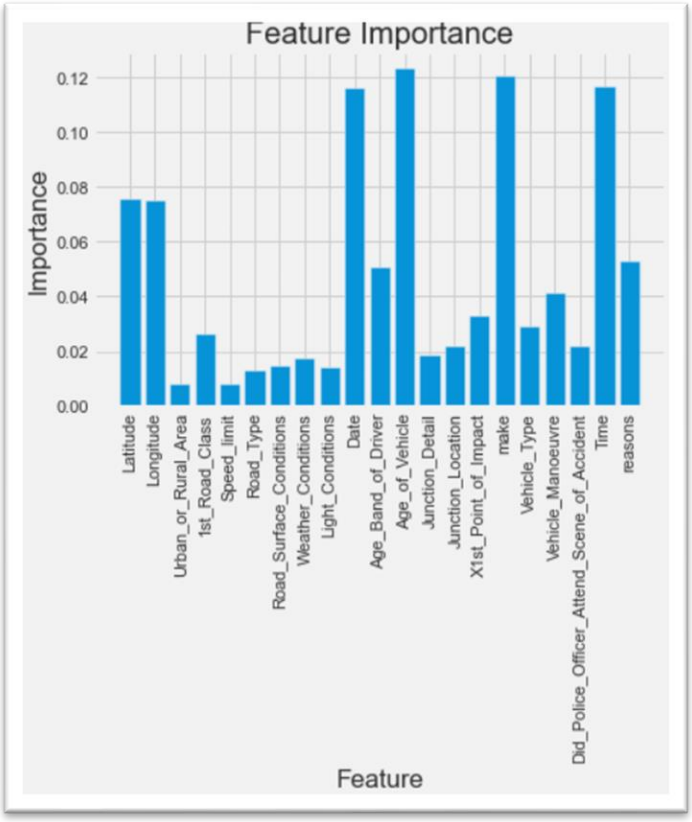

Figure 10: Feature importance for severity prediction in graphical format

Then reason, and frequency predictions are implemented by selecting features using the feature importance method also. Therefore, using six features such as latitude, longitude, date, age of vehicle, make, and time, the accident's reason prediction has been conducted. Also, the accident's frequency prediction is performed using 22 features as shown in Figure 11.

```
Latitude
Longitude
Urban_or_Rural_Area
1st_Road_Class
Speed_limit
Road_Type
Road_Surface_Conditions
Weather_Conditions
Light_Conditions
Date
Age_Band_of_Driver
Age_of_Vehicle
Junction_Detail
Junction_Location
X1st_Point_of_Impact
make
Vehicle_Type
Vehicle_Manoeuvre
Did_Police_Officer_Attend_Scene_of_Accident
Time
reasons
Accident_Severity
```

Figure 11: Selected features for frequency prediction

However, three features have been selected for the accident's vicinity clustering using Principal Component Analysis (PCA) as described in Figure 12. In addition, PCA considers all features of the dataset, project them to a lower-dimension, and reduce them to a smaller number of dominant principal ones [27]. Accordingly, the selected features are longitude, latitude, and urban or rural area which have described most of the variance in the dataset.

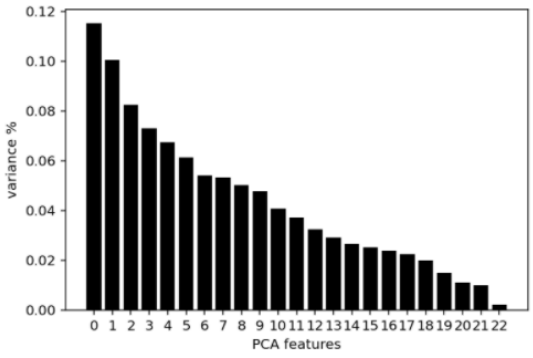

Figure 12: PCA for vicinity prediction

Model generation, training, and testing

Basically, four models have been generated using ML algorithms in order to cover the whole road accident prediction process. First, the model for road accident severity prediction has been implemented with the use of a preprocessed dataset and the selected attributes as mentioned in the feature selection section. The model was generated using the Random Forest Classifier (RFC) algorithm which is a SL algorithm. Literally, RFC is a collection of decision trees where RFC merges the output of isolated decision trees to generate the final output of the RFC [28]. Here, RFC is selected because it can be easily parallelized, and it leads to less overfitting because of containing many decision trees. Nevertheless, Random Forest is used for both classification and regression and here it was used as a classification algorithm to predict the severity of the accidents. In order to conduct the prediction, the dataset is divided into 80% of training data and 20% of testing data. After the training process, the test dataset has been utilized to predict the accident severities using the RFC algorithm. However, the model has classified the severity as slight, serious, and fatal. The model implementation is displayed in Figure 13. Since the AdaBoost classifier algorithm has shown more accuracy slightly than the RFC, the model has been trained and tested again using the AdaBoost classifier algorithm also in order to compare accuracies and produce the best outcome.

Next, reasons for road accidents are predicted with the preprocessed data and the selected features from the feature importance technique. The prediction has been conducted with the AdaBoost classifier algorithm which is a SL boosting algorithm developed on top of the decision tree family. It is applied for the reason prediction because it boosts the performance of the ML algorithm and it is better to be utilized with weak learners [29]. However, the dataset has been divided into 80% training data and 20% testing data to perform the accident's reason prediction which is classified into 7 main reasons such as careless driving, road rules violation, driving after taking drinks, speed, overtake, turning, and others. The model implementation is displayed in Figure 14.

```
#Eliminate 'Accident_Severity' column: the output feature
X = tf_acc_data.drop(['Accident_Severity'], axis=1)
#Assign 'accident_data' column to Y varible
Y = tf_acc_data['Accident_Severity']
#list the selected input features
feature_list = list(X.columns)

#Divide the dataset by 80:20 as training and testing, random_state=42
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state = 42)

#Display the measured training shape/length of training and testing datasets
print('The shape of Training Features: ', X_train.shape)
print('The shape of Training Lables: ', Y_train.shape)
print('The shape of Testing Features: ', X_test.shape)
print('The shape of Testing Lables: ', Y_train.shape)

#Implementation of Random Forest Classifier algorithm
rfc = RandomForestClassifier(max_depth=6) #n_estimators is number of trees in the forest
rfc.fit(X_train, Y_train)

#Accident severity predicitions: test data
severity_pred_results = rfc.predict(X_test)
print('severity prediction results')
print(severity_pred_results)

#Measure the accuracy
accuracy=sm.accuracy_score(Y_test, severity_pred_results)
print('Measured accuracy level of test data:',str(accuracy))
```

Figure 13: Severity prediction model implementation

```
#Eliminate 'reasons' column: the output feature
X = tf_acc_data.drop(['reasons'], axis=1)
#Assign 'reasons' column to Y varible
Y = tf_acc_data['reasons']
feature_list = list(X.columns)

#Divide the dataset by 80:20 as training and testing, random_state=42
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state = 42)

#Display the measured training shape/length of training and testing datasets
print('The shape of Training Features: ', X_train.shape)
print('The shape of Training Lables: ', Y_train.shape)
print('The shape of Testing Features: ', X_test.shape)
print('The shape of Testing Lables: ', Y_train.shape)

#Implementation of AdaBoost Classification Algorithm which is an expansion of decision tree

X_train, Y_train = make_classification(n_samples=1000, n_features=21,n_informative=2, n_redundant=0,random_state=0, shuffle=False)

adb_clf = AdaBoostClassifier(n_estimators=100, random_state=0)
adb_clf.fit(X_train, Y_train)

#Accident reason predicitions: test data
acc_reason_pred = adb_clf.predict(X_test)


#Measure the accuracy
accuracy=sm.accuracy_score(Y_test, acc_reason_pred)
print('Measured accuracy level of test data:',str(accuracy))
```

Figure 14: Reason prediction model implementation

After that, the accident's frequency prediction has been conducted with the KNN algorithm where road accidents are classified into 4 time zones of the day. Such as time zone 1 (12.00 am – 6.00 am), time zone 2 (6.00 am – 12.00 pm), time zone 3 (12.00 pm – 6.00 pm), and time zone 4 (6.00 pm – 12.00 pm). Here, the preprocessed dataset is utilized, and the most suitable attributes are selected from the feature importance process. However, the dataset was divided into 60% training data and 40% testing data to accomplish the accident's frequency prediction with the KNN algorithm. Since the K value or the number of neighbors in the KNN algorithm extremely impacts the performance of the model [30], 3 is selected as the K value for this prediction by obtaining F1 score or the accuracy rate against distinct values of K as described in Figure 15.

```
knn_scores_arr = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train, Y_train)
    knn_scores_arr.append(knn_classifier.score(X_test, Y_test))
```

Figure 15: Selecting the most suitable K value

KNN is utilized for the frequency prediction because the time zone classification should be conducted based on existing patterns. Also, it is a SL and non-parametric algorithm which manipulates the similarity of features to predict the cluster of new data point [30]. However, the model implementation process is displayed in Figure 16.

```
#Eliminate 'time_zone' column: the output feature
X = tf_acc_data_for_fp.drop(['time_zone'], axis=1)
#Assign 'time_zone' column to Y varible
Y = tf_acc_data_for_fp['time_zone']
feature_list = list(X.columns)

#train set = 60% and test set = 40%
X_train, X_test,Y_train, Y_test = train_test_split(X, Y, test_size=0.4)

#Apply KNN classifier
knn_classifier_for_fp = KNeighborsClassifier(n_neighbors=3, metric='minkowski', p=2)
knn_classifier_for_fp.fit(X_train, Y_train)

#Fit the test set
Y_pred= knn_classifier_for_fp.predict(X_test)

#Measure the accuracy
print(metrics.accuracy_score(Y_test, Y_pred))
print(metrics.f1_score(Y_test, Y_pred, average='weighted'))
```

Figure 16: Frequency prediction model implementation

In order to perform the frequency prediction, the time zones should be generated. Hence, the time attribute of the dataset has been considered and divided each data point of the dataset into one of the four time zones as the code that is shown in Figure 17.

.

```
#Read dataset
acc_data_for_fp = pd.read_csv('./dataset/preprocessed_acc_data.csv')

time_array1 = []
time_array2 = []
time_zone_array = []

for time_value1 in acc_data_for_fp['Time']:
    time_value_first_two_chars = time_value1[:2]
    time_array1.append(time_value_first_two_chars)

for time_value2 in time_array1:
    first_char_value = time_value2[0]
    second_char_value = time_value2[1]
    if second_char_value == ":" :
        time_array2.append(first_char_value)
    else:
        time_array2.append(time_value2)

for time_value3 in time_array2:
    int_time_value = int(time_value3)
    if (int_time_value >= 0) & (int_time_value < 7) :
        time_zone_array.append(1)
    elif (int_time_value >= 7) & (int_time_value < 13) :
        time_zone_array.append(2)
    elif (int_time_value >= 13) & (int_time_value < 19) :
        time_zone_array.append(3)
    elif (int_time_value >= 19) & (int_time_value < 24) :
        time_zone_array.append(4)
```

Figure 17: Time zones categorization

As the final prediction, road accident vicinity clustering has been conducted using the K-means algorithm. Since road accidents area prediction has been conducted without identifying the output labels, USL should be selected. Hence, the K-means algorithm has been chosen since it is known as the simplest USL algorithm which performs an uncomplicated process to classify a dataset with respect to a given number of clusters assuming K as the number of clusters [31]. Basically, K-means clustering involves with market segmentation, customer profiling, computer vision, geo-statistics, and astronomy. Since the road accidents vicinity clustering is relevant to the geo-statistics stream with no identified output labels, the K-means algorithm has been selected to implement the model. In addition, the preprocessed dataset with 22 attributes is taken for the clustering process. The K value that has been chosen for the clustering is 5. However, the K value selection method is the elbow technique that is displayed in Figure 18 and the elbow distribution with the best possible K value is displayed in Figure 19.

```
kms = range(1, 10)
inertias = []
for k in kms:
    # k clusters: model
    model = KMeans(n_clusters=k)

    # Fit model to samples
    pca = model.fit(PCA_components.iloc[:,:3])

    # Append the inertia to inertias lists
    inertias.append(model.inertia_)
```

Figure 18: Method of finding best K value

```
plt.plot(ks, inertias, '-o', color='black')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```

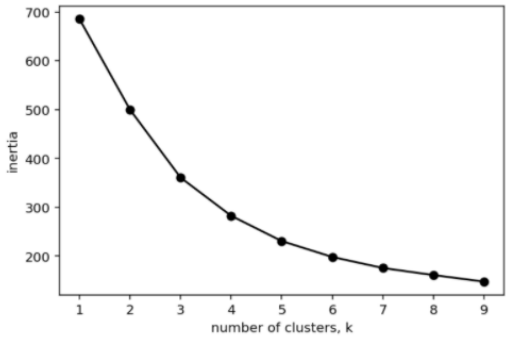

Figure 19: Elbow distribution for K-means

After that, the accident areas which have more probability for having road accidents are clustered using the K-means algorithm as described in Figure 20.

```
#state number of clusters
k_means = KMeans(n_clusters=5)
#Run the clustering algorithm
model = k_means.fit(X)
model
#Generate cluster predictions
y_pred = k_means.predict(X)
```

```
labels = k_means.labels_
metrics.calinski_harabasz_score(X, labels)
```

```
45344.68229521609
```

Figure 20: K-means model implementation

In addition, the below Figure 21 shows some of the required libraries for the prediction. Out of them. Pandas library and some scikit-learn libraries are utilized the most.

```
import pandas as pd
#feature scaling
from sklearn import metrics
#sklearn.metrics has metrics of various score functions, performance, distance computations and pairwise metrics
import sklearn.metrics as sm
#normalize labels because there are both numerical and categorical features. Use LabelEncorder
from sklearn.preprocessing import LabelEncoder
#split dataset
from sklearn.model_selection import train_test_split
```

Figure 21: Some of the required libraries for predictions

Quantitative evaluation

For each prediction in the road accident prediction component, a classification report has been generated. It includes the precision, recall, and F1 score for the relevant prediction.

The precision is a percentage of the precise classifications per class. Hence, True Positive means the actual positives that the model comprehends whereas False Positive is a result in which the model incorrectly predicts the positive class.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Also, the recall is a classification that elaborates how effectively true classes are detected. Therefore, True Positive means the actual positives that the model captures whereas False Negative means the negative class where the model wrongfully predicted.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

However, the F1 score is also generated for each prediction type that is a function of Precision and Recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recalls}$$

Functional implementation

Architectural style

The SmartCop web application has been implemented according to the MVC architectural style maintaining its quality. According to the architecture, the software application is divided into three sections such as the Model, View, and Controller. Model is the component where the data-related work is handled. Also, the View is the section where the user accesses the application or generally it is the frontend. Nevertheless, the Controller handles the business logic of the application acting as an interface between the Model and the View.

Sign-Up process

Sign Up form is implemented using ReactJS and it requires fields,

- First Name
- Last Name
- NIC
- Contact number
- Police ID
- Region
- Police station
- Email
- Password

After the relevant user enters the required data, form will be submitted after the validation process. If there are validation issues, the user would be advised to insert valid data. If the inserted data is correct, user profile would be automatically created. Therefore, the authorized police officer can Sign-In to the SmartCop web application.

Sign-In process

After creating the user profile in the Sign-Up process, user can Sign-In to the SmartCop web application. In order to proceed, user needs to enter the registered email address as the username and the password. When user clicks on the submit button, the system validates them considering the data in database. If there is a credential miss match, the system would prompt an error message otherwise the user can successfully log into the system and access the SmartCop main dashboard.

Database connection

MongoDB database has been used as the datastore of the SmartCop web application. Therefore, all data relevant to the application is stored inside the MongoDB Atlas Cloud. After creating a cluster inside the MongoDB Atlas Cloud as shown in Figure 22, the connection has been established in the application itself as shown in the Figure 23.

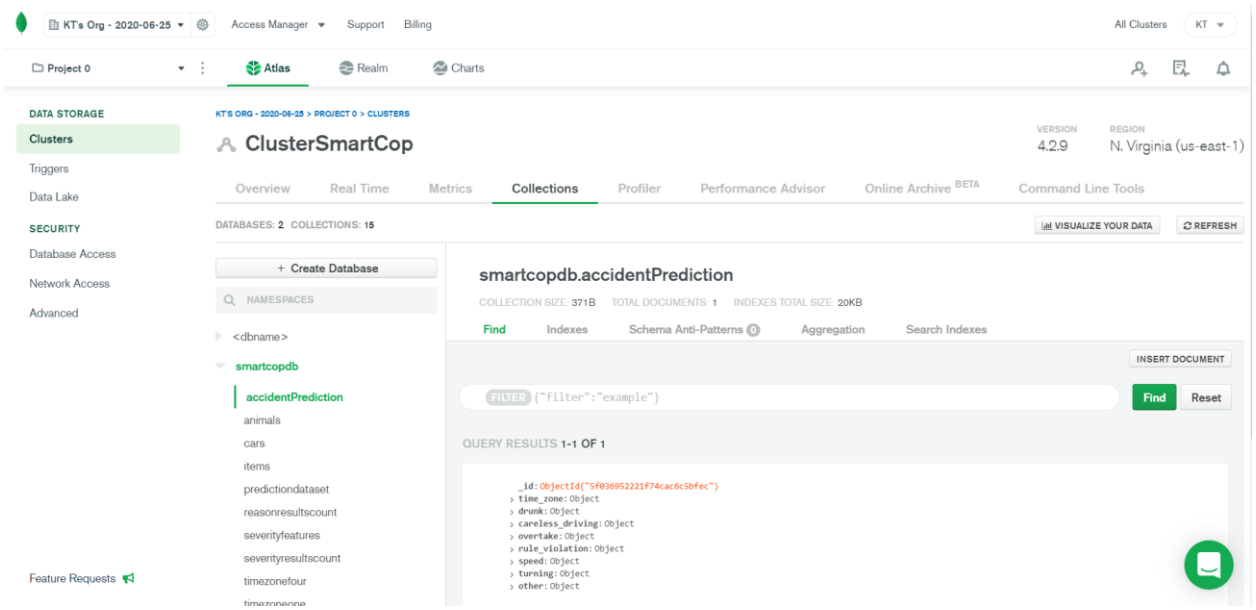

Figure 22: MongoDB Atlas Cloud SmartCop cluster

```
app.config['MONGO_URI']='mongodb://ktprojects:kt2020@clustersmartcop-shard-00-00-nhjxy.mongodb.net:27017,' \
                        'clustersmartcop-shard-00-01-nhjxy.mongodb.net:27017,' \
                        'clustersmartcop-shard-00-02-nhjxy.mongodb.net:27017/smartcopdb?ssl=true&replicaSet=' \
                        'ClusterSmartCop-shard-0&authSource=admin&retryWrites=true&w=majority'

mongo=PyMongo(app)
```

Figure 23: Database connection

API generation

Since the SmartCop web application is built up on MVC architecture, APIs according to RESTful architecture are used for routing. Especially, Create, Read, Update, and Delete (CRUD) operations are handled via the fully functional APIs. As an example, the Figure 24 shows one of the POST operations conducted on severity prediction. Also, APIs are used to handle server connections on backend too.

```
#add severity prediction results to db
@app.route('/add_prediction_results',methods=['POST'])
def add_prediction_results():
    if sv_rs_count and request.method == 'POST':
        id = mongo.db.severityresultscount.insert(sv_rs_count)
        resp = jsonify("severity results added successfully")
        resp.status_code = 200
        return resp

    else:
        return not_found()
```

Figure 24: Severity prediction POST method usage

Functionality in interfaces

The below figures show the relevant interfaces for road accident prediction component in the SmartCop web application. Main dashboard is displayed by the Figure 25 and the dashboard for the road accident prediction function is presented in Figure 26. In addition, accident severity prediction details interface can be visible via Figure 27 and Figure 28. Also, the reason prediction statistical interface is displayed in Figure 29.

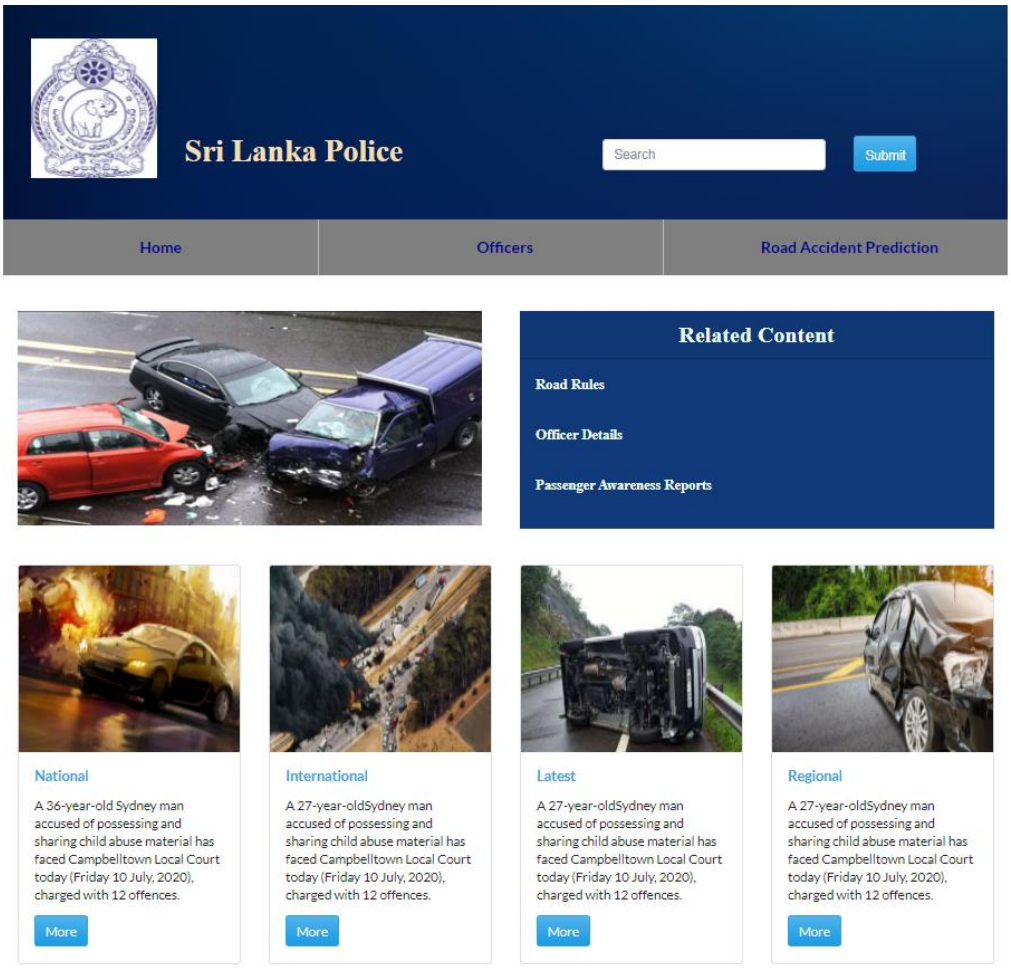

Figure 25: SmartCop web application main dashboard

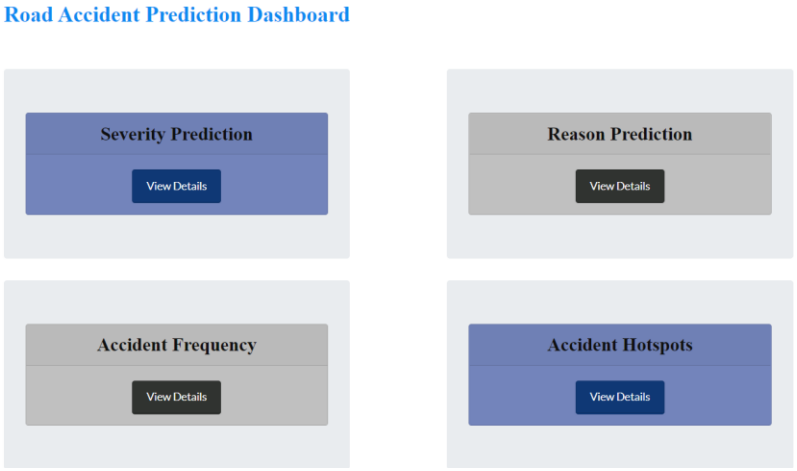

Figure 26: Road accident prediction main dashboard

Figure 27: Road accident severity prediction statistical interface

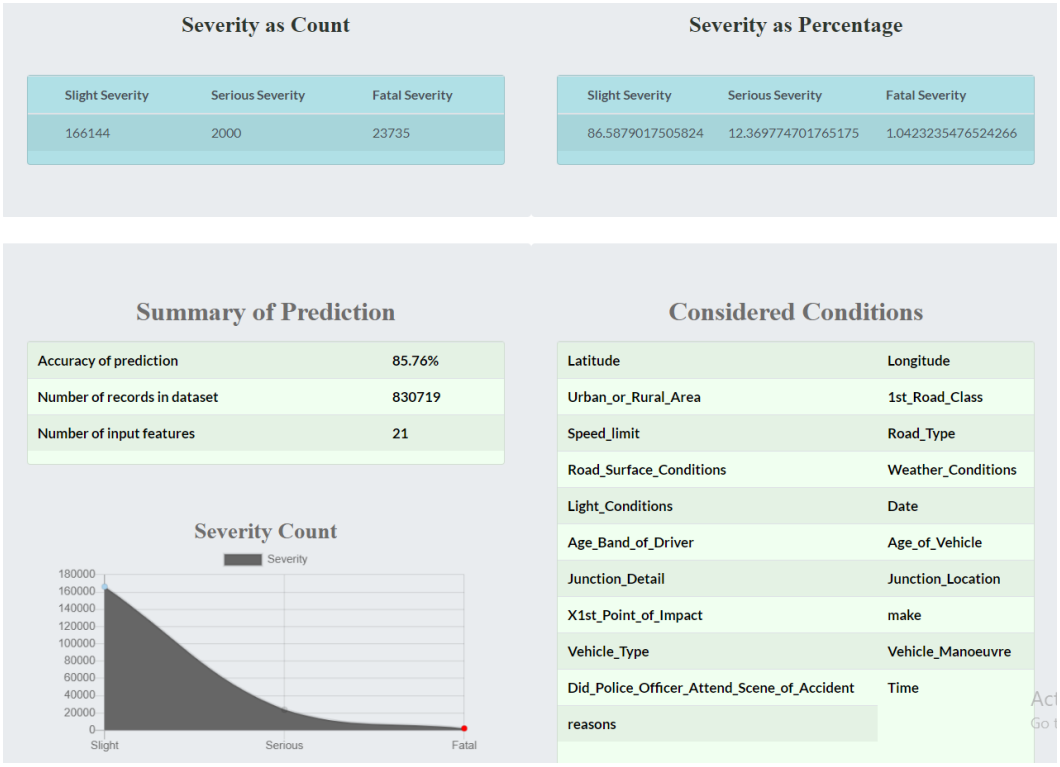

Figure 28: Road accident severity prediction analytical interface

Figure 29: Road accident reason prediction statistical interface

In addition, more detailed code of the whole road accident prediction component is attached in the Appendix A section. It includes the supportive functionalities for the prediction process and the component creation.

3.RESULTS & DISCUSSION

3.1 Results

This section elaborates the results and observations of the road accident prediction component in the SmartCop research project. The results have been obtained according to the prediction process mentioned in the methodology. Hence, the observed results can be categorized according to the four predictions such as road accident severity prediction, reasons classification, frequency prediction, and vicinity clustering. The accuracy and effectiveness of the prediction process depended on these outcomes. However, the outcomes are predominantly based on the dataset and the utilized ML algorithms.

3.1.1 Road accident's severity prediction

Road accident's severity is predicted and classified into three types of severities such as slight, serious, and fatal using RFC. Accidents with slight severities are not too dangerous and caused for imperceptible injuries for the passengers and damages for the vehicles. Also, they do not impact critically on the infrastructure. However, the fatal accidents cause for deaths and severe damages on passengers, vehicles, and road infrastructure also. Moreover, the serious type of accidents originates impacts on vehicles and public being in the middle state of both slight and fatal type of accidents.

According to the severity prediction results of the RFC algorithm, 82.7% of accidents are identified as slight accidents where most of the accidents do not dangerously effect on the passengers, vehicles, and surroundings. Also, 11.8% of accidents belong to the serious category and the fatal accidents are recorded very rarely as 5.4% as a percentage. The results distribution is presented in Figure 30.

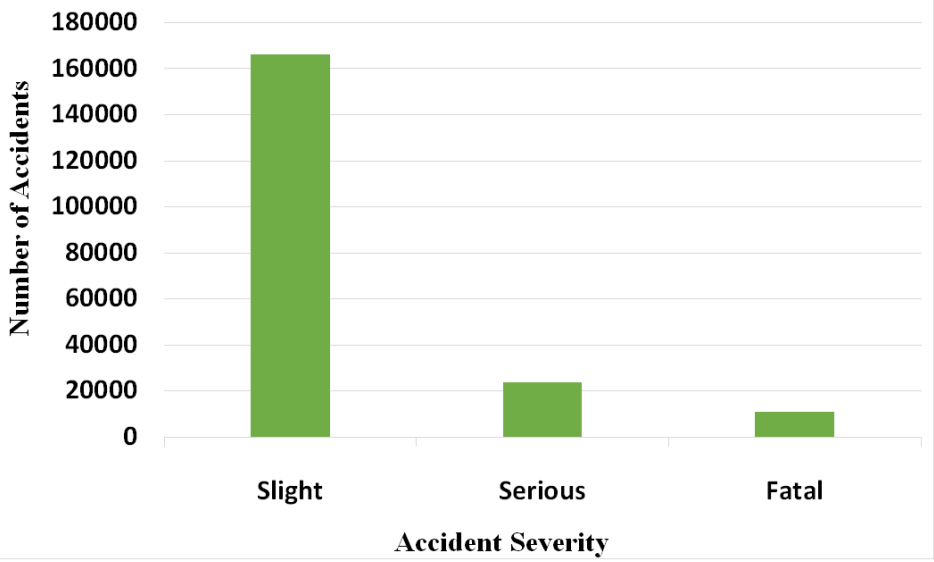

Figure 30: Severity prediction results

However, the accuracy of the road accident's severity prediction with RFC is 85.76% as a percentage where it has been given 85.64% of accuracy after the prediction has been conducted with AdaBoost Classifier. Since the accuracy of the RFC is greater than the AdaBoost and the RFC performs a parallel ensembling than the AdaBoost, the severity prediction results have been gained via the RFC algorithm and the relevant accuracies are displayed as a classification report in the Figure 31.

Road accident Severity Classification Report :

	precision	recall	f1-score	support
Slight	0.01	0.95	0.02	22
Serious	0.04	0.57	0.07	1490
Fatal	1.00	0.86	0.92	164632
accuracy			0.86	166144
macro avg	0.35	0.79	0.34	166144
weighted avg	0.99	0.86	0.92	166144

Figure 31: Severity prediction classification report

Moreover, the confusion matrix distributed in Figure 32 depicts the correct severity predictions in the top left and bottom right corners whereas the lower left and upper right corners show the predictions missed by the model.

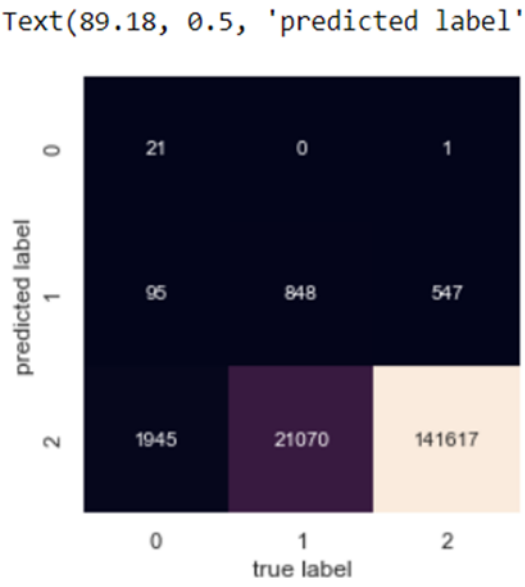

Figure 32: Confusion matrix for accident's severity prediction

3.1.2 Road accident's reason prediction

Reasons for the road accidents are predicted with the AdaBoost classifier algorithm. According to the root cause of the accidents, they are classified into seven categories. Hence, the root causes consist careless driving, drunk driving, high speed, overtake, rules violation, turning and any other factors. Figure 33 displays the graphical distribution of predicted reasons as categories along with the number of accidents recorded for each category.

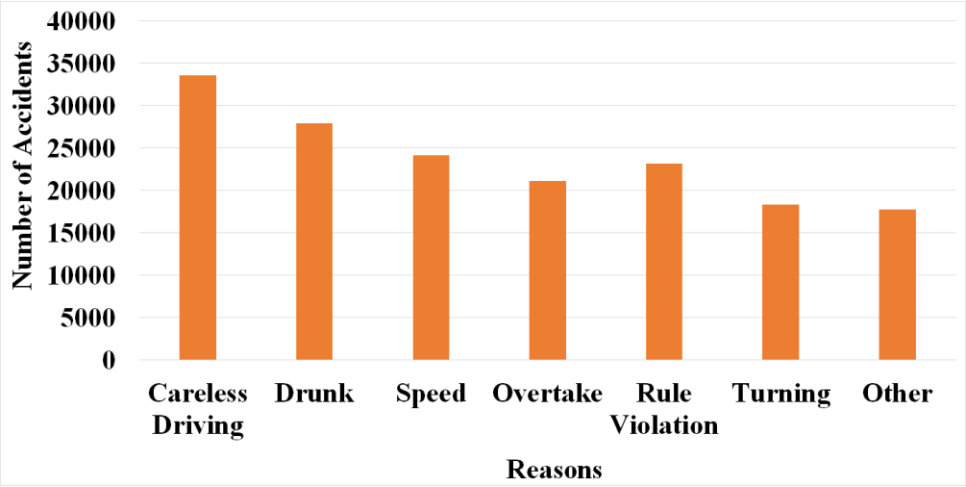

Figure 33: Reason prediction results

As the statistics describe, the highest number of accidents happens because of careless driving and it is 20.20% as a percentage. Also, the slighter number of accidents are recorded because of the other reasons which do not belong to careless driving, drunk, speed, overtake, rule violation, and turning root causes. It is 10.69% as a percentage. However, the drunk driving is the second highest root cause, which has evolved in 16.80% of accidents.

Moreover, the accuracy given for the reason prediction with 21 input features is 24.12% but the model has generated 48.12% of accuracy when input features are reduced to six features. In addition, these six features have been selected by performing the feature

importance technique and the selected features are latitude, longitude, date, age of vehicle, make, and time. Therefore, the performance of the prediction has been increased by using the most important features.

3.1.3 Road accident's frequency prediction

Third, accident's frequency has been classified in to four time zones of the day using KNN algorithm. Such as,

- Time zone 1 (12.00 am to 6.00 am)
- Time zone 2 (6.00 am to 12.00 pm)
- Time zone 3 (12.00 pm to 6.00 pm)
- Time zone 4 (6.00 pm to 12.00 pm)

However, the Figure 34 elaborates the predicted road accidents' frequencies for each time zone. Hence the police officers can get an idea of the accident peak time zone of the day by observing these results.

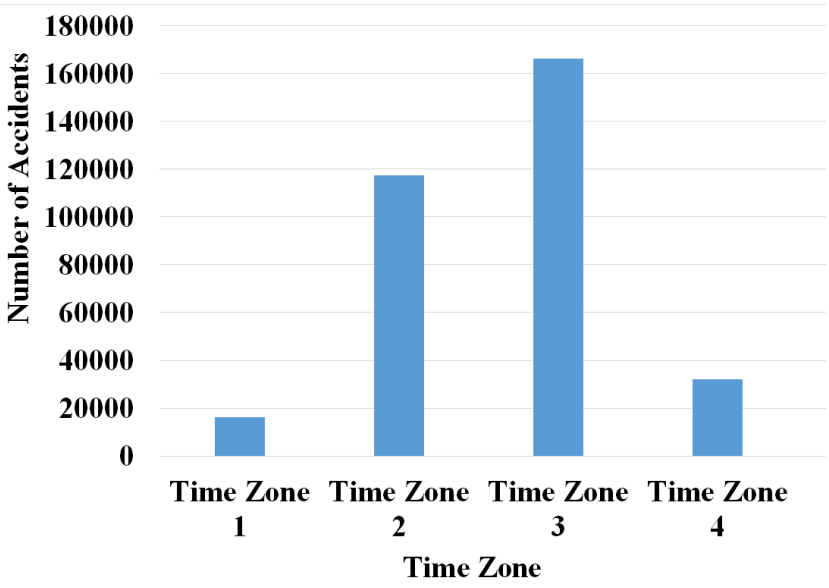

Figure 34: Frequency prediction results

As the results depict, the highest accident frequency is recorded in time zone 3 with 50.07% of percentage and time zone 1 shows the least number of accidents with 4.90% of percentage. Therefore, time zone 3 (12.00 pm to 6.00 pm) can be identified as the highest accident peak time. Also, it is logical to conclude that many accidents are occurred during the daytime since time zone 2 belongs to the 6.00 am to 12.00 pm and time zone 3 belongs to 12.00 pm to 6.00 pm time durations. Nevertheless, accuracy of the frequency prediction with KNN algorithm is 63.15% as a percentage. Hence, the generated classification report relevant to the prediction is displayed in the Figure 35.

Road accident frequency Classification Report :

	precision	recall	f1-score	support
time_zone1	0.37	0.48	0.42	16341
time_zone2	0.66	0.62	0.64	116253
time_zone3	0.73	0.65	0.69	167481
time_zone4	0.40	0.64	0.49	32213
accuracy			0.63	332288
macro avg	0.54	0.60	0.56	332288
weighted avg	0.66	0.63	0.64	332288

Figure 35: Frequency prediction classification report

Also, the generated confusion matrix is displayed in Figure 36. It further elaborates the correct and incorrect predictions of the model.

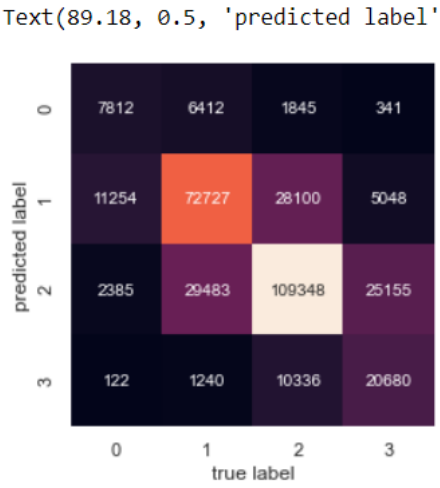

Figure 36: Confusion matrix for frequency prediction

3.1.4 Road accident's vicinity prediction

Finally, road accident's vicinity has been predicted with the K-means clustering algorithm. In addition, four main clusters have been identified as accident hotspots in the considered region. Therefore, the police can clearly observe the vicinities with high potential for road accidents. Figure 37 depicts the identified clusters' visualization.

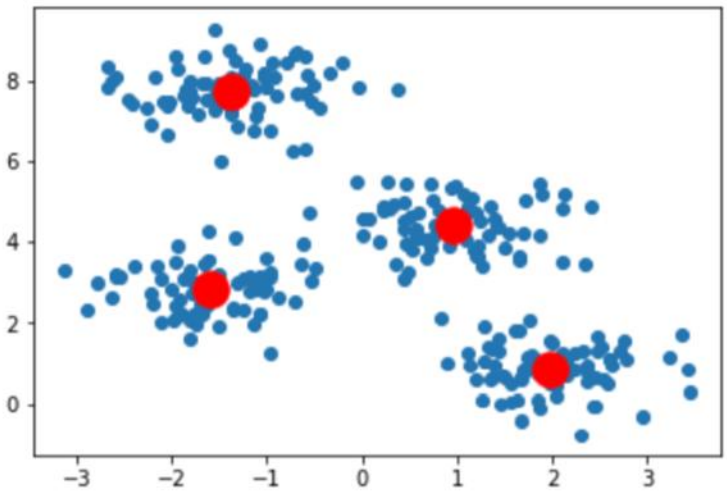

Figure 37: Accident hotspots

3.2 Research Findings

The SmartCop research project facilitates an automated platform in which the main intention is to mitigate the impact of road accidents. Therefore, the road accident prediction component generates and delivers four predictions via a single system such as predict accident's severity, reason, frequency, and vicinity. The idea behind this prediction concept is innovative since these four predictions were not delivered from a single system before as described in the literature review of this report. Basically, researches consider only one road accident's prediction category at a time. Therefore, the road accident prediction component owns a weighted prediction depth also. In addition, this component

does not focus only on conducting four predictions and provides a statistical report as conventional methods, but it analyzes the outcomes of four predictions and facilitates the functionalities of the SmartCop web application.

In addition, it is found that the model training in several times is important to increase the performance of the model. Also, it is mandatory of choosing a real dataset for predictions to generate a model with a high performance and reliability. Moreover, the wise use of input features for a prediction matters to increase the accuracy of a model. Therefore, the most appropriate feature selection method should be chosen for the prediction. In this road accident prediction component, accident's severity, reason, and frequency predictions have been utilized the feature importance and researching on related works methods to select the best features set. Also, the accident's vicinity clustering has been used the PCA technique to select the most applicable components. Nevertheless, data preprocessing is important to obtain the most accurate prediction outcomes since it cleans and normalizes the dataset. These facts support to generate an effective and efficient predictions.

3.3 Discussion

The completed research project, SmartCop is an automated platform that has been implemented to mitigate the impact of road accidents. Accordingly, this platform contains four main components such as predict road accidents, recommend and schedule police officers, enhance road accidents prevention awareness, and enhance road accidents response awareness. In addition, the SmartCop platform is divided into two sub platforms as web-based application which consists the road accident prediction, and police officers' recommendation and scheduling components. And the other sub platform is a game-based learning mobile application which contains the road accidents prevention awareness game, and the road accidents response awareness game. Out of them, this research component is the road accident prediction that is the first main function of the SmartCop web application.

Accordingly, the road accident prediction component contains four predictions which can be also identified as subcomponents such as road accident's severity prediction, reasons classification, frequency prediction, and vicinity clustering. Also, the outcomes are analyzed and delivered to the end user via the SmartCop web application. In addition, main functionalities that this component facilitates are listed below.

- Search for predicted number of accidents with their frequencies in a relevant region
- View Accident hotspots
- View predicted severities and reasons of accidents
- View accident peak time
- Obtain summary reports of each prediction
- Statistical observations for each prediction type

Since road accidents are getting increased both locally and globally and impact a lot in the society both economically and socially, the Department of police has a major requirement to diminish the consequences of the road accidents as the protectors of public. As discussed in the literature, there are many solutions have been identified in order to reduce road accidents and their impacts. Some of them are manual processes and some of them are fully functional automated systems. But most of the related works have considered on only one prediction type mainly like severity or area prediction. Also, some statistical analysis is conventional to obtain accurate results. Therefore, the road accident prediction component has been developed to fill that identified gap and as a solution to the major requirement police including four predictions. Also, the outcomes support to reduce the wastage of country's capital to settle the consequences of road accidents and save human lives facilitating both economically and socially.

Moreover, the latest technologies as Python and ReactJs, and tools as PyCharm and VS Code have been utilized to implement this component. In order to select them, many articles, personal blogs, and advices of technical and academic expertise were involved

exceptionally. However, many challenges were faced during the development of the road accident prediction component. The major challenge was to find an actual dataset with the required features set. After a deep research of them, a proper actual dataset has been found which is available online [24]. Also, the Mirihana police station in Sri Lanka has guided a lot as the representative of the Department of Police to identify police regions, their experiences and records regarding previous road accidents, the way of classifying accident hotspots manually, and their major requirements. Therefore, a detailed actual dataset with broad insight of road accidents are obtained by them. In addition, it was really a challenge to select the most suitable algorithm for each prediction. Therefore, many related works and articles have been referred to select some applicable algorithms and the most appropriate algorithm has been chosen after comparing their accuracies mostly. The other prominent challenge was to select proper input features and that has been accomplished by using several techniques like feature importance, PCA, and referring the related works.

Nevertheless, a completed road accident prediction component has been developed successfully and delivered along with the SmarCop web application. Initially, the application can be accessible to police stations of a single police region. In future, the Island wide accessibility will be enabled facilitating multiple premium features. However, the SmartCop web application can convince to mitigate the impact of road accidents facilitating the public enormously.

4. CONCLUSION

The SmartCop automated platform has been developed to mitigate the impact of road accidents. This platform contains a web-based application and as well as a mobile application. Therefore, this report has been generated concerning the first component of the SmartCop wed-based application which is the road accident prediction component.

Recently, road accidents have become one of the tremendous problems all around the world remaining deaths, fatalities, and other impacts as social and economic deprivations. Therefore, the whole world is disturbed by road accidents. In addition, police have a major requirement of reducing road accidents and their impacts as the protectors of the public. As the literature survey depicts, there are many automated solutions that have been conducted with road accident predictions but all of them concerned only one aspect of road accidents which is not sufficient to grant effective decisions for the police officers. In order to fill that gap, the SmartCop road accident prediction component has been implemented.

However, this component contains four types of predictions. First, road accident severity prediction has been implemented using the RFC algorithm and classified the severity as slight, serious, and fatal. Second, reasons for road accidents have been predicted into seven reasons such as careless driving, drunk driving, high speed, overtake, rules violation, turning, and any other factors. The classification is conducted via the AdaBoost classifier algorithm. Third, frequency of road accidents has been predicted with the KNN algorithm and classified into four time zones as time zone 1 (12.00 am – 6.00 am), time zone 2 (6.00 am – 12.00 pm), time zone 3 (12.00 pm – 6.00 pm), and time zone 4 (6.00 pm – 12.00 pm). Finally, the road accident's vicinity has been clustered using the K-means algorithm. The outcomes of the predictions are analyzed again and displayed in the SmartCop web application with multiple statistics. In addition, there are user-friendly functionalities for police officers to obtain the required data from the application.

As per the commercialization aspect, the SmartCop web application can be served as a series of packages with premium features. All in all, the results of the SmartCop road accident prediction component ensure that the required service for the police officers and the public has been provided more effectively and efficiently.

REFERECES

[1] (2020, 23-April-2020). *Road traffic injuries*. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries#:~:text=Approximately%201.35%20million%20people%20die,of%20their%20gross%20domestic%20product.

[2] M. Másilková, "Health and social consequences of road traffic accidents," *Kontakt,* vol. 19, no. 1, pp. e43-e47, 2017/03/01/ 2017.

[3] R. Gorea, "Financial impact of road traffic accidents on the society," *International Journal of Ethics, Trauma & Victimology,* vol. 2, 07/29 2016.

[4] S.-h. Park, S.-m. Kim, and Y.-g. Ha, "Highway traffic accident prediction using VDS big data analysis," *The Journal of Supercomputing,* vol. 72, 01/20 2016.

[5] V. M. Ramachandiran, P. N. K. Babu, and R. Manikandan, "Prediction of road accidents severity using various algorithms," *International Journal of Pure and Applied Mathematics,* vol. 119, pp. 16663-16669, 01/01 2018.

[6] K. R. Sumana and D. Phaneendra, *Smart Automated Modelling using Eclat Algorithm for Traffic Accident Prediction*. 2019.

[7] A. Ashtaiwi, "Intelligent Road Crashes Avoidance System," 2019.

[8] F. Labib, A. Rifat, M. Hossain, A. Das, and F. Nawrine, "Road Accident Analysis and Prediction of Accident Severity by Using Machine Learning in Bangladesh," in *7th International Conference on Smart Computing & Communications*, 2019, pp. 1-5.

[9] G. Kaur, E. H. J. t. I. C. o. C. Kaur, Communication, and N. Technologies, "Prediction of the cause of accident and accident prone location on roads using data mining techniques," pp. 1-7, 2017.

[10] E. Reveron and A. Cretu, "A Framework for Collision Prediction Using Historical Accident Information and Real-time Sensor Data: A Case Study for the City of Ottawa," in *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2019, pp. 1-7.

[11] Q. Liyan and S. Chunfu, *Macro Prediction Model of Road Traffic Accident Based on Neural Network and Genetic Algorithm*. 2009, pp. 354-357.

[12] A. J. Graettinger, J. K. Lindly, and G. J. Mistry, "Display and analysis of crash data," (in English), Research Paper 2005.

[13] T. Sivakumar and D. Amarathung, "Development of Traffic Accident Prediction Models Using Traffic and Road Characteristics : A Case Study from Sri Lanka," 2015.

[14] R. G. Ramani, S. J. I. I. C. o. C. I. Shanthi, and C. Research, "Classifier prediction evaluation in modeling road traffic accident data," pp. 1-4, 2012.

[15] JetBrains. (1-April-2020). *PyCharm*. Available: https://www.jetbrains.com/pycharm/

[16] (1-April-2020). *Visual Studio Code*. Available: https://code.visualstudio.com/docs/editor/whyvscode

[17] (1-Jan-2020). *What is Python? Executive Summary*. Available: https://www.python.org/doc/essays/blurb/

[18] (23-March-2020). *Flask (web framework)*. Available: https://en.wikipedia.org/wiki/Flask_(web_framework)

[19] (1-March-2020). *MongoDB Atlas*. Available: https://docs.atlas.mongodb.com/

[20] "Beginners Guide to ReactJS," Accessed on: 23-April-2020Available: https://medium.com/zenofai/beginners-guide-to-reactjs-3ca07f56d526#:~:text=React%20or%20ReactJS%20is%20a%20JavaScript%20library.&text=React%20is%20an%20open%2Dsource,application%20(Model%20View%20Controller).

[21] (25-April-2020). *Anaconda Navigator*. Available: https://docs.anaconda.com/anaconda/navigator/#:~:text=Anaconda%20Navigator%20is%20a%20desktop,without%20using%20command%2Dline%20commands.&text=To%20get%20Navigator%2C%20get%20the%20Navigator%20Cheat%20Sheet%20and%20install%20Anaconda.

[22] (2020, 12-July-2020). *Commercialization*. Available: https://www.investopedia.com/terms/c/commercialization.asp#:~:text=Commercialization%20is%20the%20process%20of,the%20new%20product%20or%20service.

[23] (23-May-2020). *Why is Testing Necessary?* Available: https://www.toolsqa.com/software-testing/istqb/why-is-testing-

necessary/#:~:text=The%20testing%20is%20important%20since,and%20high%2Dperformance%20software%20operation.

[24] Thanasis. UK Road Safety: Traffic Accidents and Vehicles [Online]. Available: https://www.kaggle.com/tsiaras/uk-road-safety-accidents-and-vehicles#Accident_Information.csv

[25] R. Shaikh, "Feature Selection Techniques in Machine Learning with Python," no. 1-Jan-2020, Available: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

[26] S. Ariel, "Accident Severity Classification," no. 12-Jan-2020, Available: https://www.kaggle.com/arielyaakobi10/accident-severity-classification-shirley-ariel

[27] D. Kavyazin, "Principal Component Analysis and k-means Clustering to Visualize a High Dimensional Dataset," Accessed on: 23-June-2020Available: https://medium.com/@dmitriy.kavyazin/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2

[28] S. Yıldırım, "Decision Trees and Random Forests — Explained," vol. 2020, Available: https://towardsdatascience.com/decision-tree-and-random-forest-explained-8d20ddabc9dd

[29] Z. Little, "AdaBoost," vol. 2020, no. 23-Feb-2020, Available: https://medium.com/@xzz201920/adaboost-7f80fa39584b#:~:text=AdaBoost%20can%20be%20used%20to%20boost%20the%20performance%20of%20any,trees%20on%20binary%20classification%20problems.

[30] R. Khandelwal, "K-Nearest Neighbors(KNN)," Accessed on: 5-May-2020Available: https://medium.com/datadriveninvestor/k-nearest-neighbors-knn-7b4bd0128da7

[31] D. Madushan, "Introduction to K-means Clustering," Accessed on: 23-May-2020Available: https://medium.com/@dilekamadushan/introduction-to-k-means-clustering-7c0ebc997e00#:~:text=The%20K%2Dmeans%20clustering%20algorithm,to%20the%20most%20relevant%20group.

APPENDICES

Appendix A

Source code of the road accident's severity prediction

```
import pandas as pd
#feature scaling
from sklearn import metrics
#sklearn.metrics has metrics of various score functions, performance,
distance computations and pairwise metrics
import sklearn.metrics as sm
#normalize labels because there are both numerical and categorical
features. Use LabelEncorder
from sklearn.preprocessing import LabelEncoder
#split dataset
from sklearn.model_selection import train_test_split
#Random Forest Classifier algorithm
from sklearn.ensemble import RandomForestClassifier
#create a dataframe
from pandas import DataFrame


#Load the complete Accident dataset
acc_data_edited = pd.read_csv('edited_accident_vehicle_data.csv',
encoding = 'latin')

#Sampling data by selecting 22 features over 52 features
acc_data = acc_data_edited[['Latitude','Longitude',
'Urban_or_Rural_Area','1st_Road_Class', 'Speed_limit',
          'Road_Type', 'Road_Surface_Conditions', 'Weather_Conditions',
'Light_Conditions', 'Date',
          'Age_Band_of_Driver', 'Age_of_Vehicle', 'Junction_Detail',
'Junction_Location', 'X1st_Point_of_Impact',
          'make', 'Vehicle_Type',
'Vehicle_Manoeuvre','Did_Police_Officer_Attend_Scene_of_Accident','Time
','reasons','Accident_Severity'
         ]]

#Count number of records
num_of_records=len(acc_data)
print(num_of_records)

#Display dataset including features
print(acc_data.shape)
print(acc_data.head())

#Checking for sum of all null values
sum_of_nulls = sum(acc_data.isnull().sum())
print(sum_of_nulls)
```

```
#Checking for missing values in each column
count_of_nulls_each_column = acc_data.isnull().sum()
print(count_of_nulls_each_column)

#label encoding
lbl_enc = LabelEncoder()
tf_acc_data = accident_df3.copy()
for i in accident_df3.columns:
    tf_acc_data[i]=lbl_enc.fit_transform(accident_df3[i])

#Eliminate 'Accident_Severity' column: the output feature
X = tf_acc_data.drop(['Accident_Severity'], axis=1)
#Assign 'accident_data' column to Y varible
Y = tf_acc_data['Accident_Severity']
#list the selected input features
feature_list = list(X.columns)

#Divide the dataset by 80:20 as training and testing, random_state=42
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state = 42)

#Display the measured training shape/length of training and testing
datasets
print('The shape of Training Features: ', X_train.shape)
print('The shape of Training Lables: ', Y_train.shape)
print('The shape of Testing Features: ', X_test.shape)
print('The shape of Testing Lables: ', Y_train.shape)

#Implementation of Random Forest Classifier algorithm
rfc = RandomForestClassifier(max_depth=6) #n_estimators is number of
trees in the forest
rfc.fit(X_train, Y_train)

#Accident severity predicitions: test data
severity_pred_results = rfc.predict(X_test)
print('severity prediction results')
print(severity_pred_results)

#Measure the accuracy
accuracy=sm.accuracy_score(Y_test, severity_pred_results)
print('Measured accuracy level of test data:',str(accuracy))

#Classification report with respect to the predicted values

print("Road accident Severity Classification Report : \n\n",
metrics.classification_report(severity_pred_results, Y_test,
target_names = ["Slight","Serious","Fatal"]))

#convert test results in to dataframe
df_severity_test_results = DataFrame(severity_pred_results, columns=
['Accident_Severity'])
print(df_severity_test_results)
```

```
#convert train results in to dataframe
df_severity_train_results = DataFrame(rfc, columns=
['Accident_Severity'])
print(df_severity_train_results)

#convert dataframe to json
df_severity_test_results.to_json
(r'./json_data/severity_test.json',orient='columns')
df_severity_train_results.to_json
(r'./json_data/severity_train.json',orient='columns')
```

Source code of the road accident's reason prediction

```
import pandas as pd
#feature scaling
from sklearn import metrics
#sklearn.metrics has metrics of various score functions, performance,
distance computations and pairwise metrics
import sklearn.metrics as sm
#normalize labels because there are both numerical and categorical
features. Use LabelEncorder
from sklearn.datasets import make_classification
from sklearn.preprocessing import LabelEncoder
#split dataset
from sklearn.model_selection import train_test_split
#AdaBoost Classification algorithm
from sklearn.ensemble import AdaBoostClassifier
#create a dataframe
from pandas import DataFrame

#Load the complete Accident dataset
acc_data_origin = pd.read_csv('edited_accident_vehicle_data.csv',
encoding = 'latin')

#Sampling data by selecting 22 features over 52 features
acc_data_for_reason_pred = acc_data_origin[['Latitude','Longitude',
'Date','Age_of_Vehicle','make','Time','reasons']]



#select the dataset except the Accident_Index column
acc_data=acc_data_for_reason_pred.loc[:,
acc_data_for_reason_pred.columns != 'Accident_Index']

#Checking for sum of all null values
sum_of_nulls = sum(acc_data.isnull().sum())

#Checking for missing values in each column
count_of_nulls_each_column = acc_data.isnull().sum()


#Normalize data
```

```
lbl_enc = LabelEncoder()
tf_acc_data = accident_df2.copy()
for i in accident_df2.columns:
    tf_acc_data[i]=lbl_enc.fit_transform(accident_df2[i])

#Eliminate 'reasons' column: the output feature
X = tf_acc_data.drop(['reasons'], axis=1)
#Assign 'reasons' column to Y varible
Y = tf_acc_data['reasons']
feature_list = list(X.columns)

#Divide the dataset by 80:20 as training and testing, random_state=42
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state = 42)

#Display the measured training shape/length of training and testing
datasets
print('The shape of Training Features: ', X_train.shape)
print('The shape of Training Lables: ', Y_train.shape)
print('The shape of Testing Features: ', X_test.shape)
print('The shape of Testing Lables: ', Y_train.shape)

#Implementation of AdaBoost Classification Algorithm which is an
expansion of decision tree

X_train, Y_train = make_classification(n_samples=1000,
n_features=21,n_informative=2, n_redundant=0,random_state=0,
shuffle=False)

adb_clf = AdaBoostClassifier(n_estimators=100, random_state=0)
adb_clf.fit(X_train, Y_train)

#Accident reason predicitions: test data
acc_reason_pred = adb_clf.predict(X_test)


#Measure the accuracy
accuracy=sm.accuracy_score(Y_test, acc_reason_pred)
print('Measured accuracy level of test data:',str(accuracy))

# Display classification report with respect to the predictions

print("Road accident Reason Classification Report : \n\n",
      metrics.classification_report(acc_reason_pred, Y_test,
target_names=["careless driving","drunk","speed", "overtake", "rule
violating", "turning", "other"]))


#convert test results in to dataframe
df_reason_test_results = DataFrame(acc_reason_pred, columns=
['reasons'])
#convert train results in to dataframe
df_reason_train_results = DataFrame(adb_clf, columns= ['reasons'])
```

```
#convert dataframe to json
df_reason_test_results.to_json
(r'./json_data/reason_test.json',orient='columns')
df_reason_train_results.to_json
(r'./json_data/reason_train.json',orient='columns')
```

Source code of the road accident's frequency prediction

```
import numpy as np
import pandas as pd
#normalize labels because there are both numerical and categorical
features. Use LabelEncorder
from sklearn.preprocessing import LabelEncoder
#split dataset
from sklearn.model_selection import train_test_split
#Import KNN classifier
from sklearn.neighbors import KNeighborsClassifier
#Import metrics to test accuracy and scores
from sklearn import metrics
#create a dataframe
from pandas import DataFrame

#Read dataset
acc_data_for_fp =
pd.read_csv('./dataset/acc_data_for_frequency_prediction.csv')
print(acc_data_for_fp.head())

#label encoding
lbl_enc = LabelEncoder()
tf_acc_data_for_fp = acc_data_for_fp.copy()
for i in acc_data_for_fp.columns:
    tf_acc_data_for_fp[i]=lbl_enc.fit_transform(acc_data_for_fp[i])

#Eliminate 'time_zone' column: the output feature
X = tf_acc_data_for_fp.drop(['time_zone'], axis=1)
#Assign 'time_zone' column to Y varible
Y = tf_acc_data_for_fp['time_zone']
#collect and save the required features list
feature_list = list(X.columns)

#train set = 60% and test set = 40%
X_train, X_test,Y_train, Y_test = train_test_split(X, Y, test_size=0.4)

#Apply KNN classifier
knn_classifier_for_fp = KNeighborsClassifier(n_neighbors=3,
metric='minkowski', p=2)
knn_classifier_for_fp.fit(X_train, Y_train)

#Fit the test set
Y_pred= knn_classifier_for_fp.predict(X_test)
```

```
#Measure the accuracy
print(metrics.accuracy_score(Y_test, Y_pred))
print(metrics.f1_score(Y_test, Y_pred, average='weighted'))

# Display classification report with respect to the predictions

print("Road accident frequency Classification Report : \n\n",
      metrics.classification_report(Y_pred, Y_test,
target_names=["time_zone1","time_zone2","time_zone3", "time_zone4"]))


#convert test results in to dataframe
df_frequency_test_results = DataFrame(Y_pred, columns= ['time_zone'])

#convert dataframe to json
df_frequency_test_results.to_json
(r'./json_data/frequency_test.json',orient='columns')
```

Source code of the road accident's vicinity prediction

In order to include the code for making plots of relevant area prediction, the code implemented in the Jupyter Notebook is displayed in here. Also, the model has been implemented for a random sample of 100 records in here.

```
# Required Libraries
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
%config InlineBackend.figure_format='retina'
#normalize labels because there are both numerical and categorical
features. Use LabelEncorder
from sklearn.preprocessing import LabelEncoder
import numpy as np
from sklearn import metrics


accident_df_for_ap = pd.read_csv('acc_data_for_area_prediction.csv'
)
accident_df_for_ap.head()

#randomly select 100 records
smple_df_for_ap = accident_df_for_ap.loc[np.random.choice(accident_
df_for_ap.index, size=100)]
#Count number of records
```

```
num_of_records=len(smple_df_for_ap)
print(num_of_records)

#label encoding
lbl_enc = LabelEncoder()
tf_acc_data = smple_df_for_ap.copy()
for i in smple_df_for_ap.columns:
    tf_acc_data[i]=lbl_enc.fit_transform(smple_df_for_ap[i])

# Standardize the data
X_std = StandardScaler().fit_transform(tf_acc_data)

# Create PCA instance
pca = PCA(n_components=23)
principalComponents = pca.fit_transform(X_std)

# Plot the explained variances
features = range(pca.n_components_)
plt.bar(features, pca.explained_variance_ratio_, color='black')
plt.xlabel('PCA features')
plt.ylabel('variance %')
plt.xticks(features)

# Save components to a DataFrame
PCA_components = pd.DataFrame(principalComponents)

plt.scatter(PCA_components[0], PCA_components[1], alpha=.1, color='
black')
plt.xlabel('PCA 1')
plt.ylabel('PCA 2')

kms = range(1, 10)
inertias = []
for k in kms:
    # k clusters: model
    model = KMeans(n_clusters=k)

    # Fit model to samples
    pca = model.fit(PCA_components.iloc[:,:3])

    # Append the inertia to inertias lists
    inertias.append(model.inertia_)

plt.plot(ks, inertias, '-o', color='black')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()

#state number of clusters
k_means = KMeans(n_clusters=4)
```

```
#Run the clustering algorithm
model = k_means.fit(PCA_components.iloc[:,:3])
model
#Generate cluster predictions
y_pred = k_means.predict(PCA_components.iloc[:,:3])

labels = k_means.labels_
metrics.calinski_harabasz_score(PCA_components.iloc[:,:3], labels)

kmeans = KMeans(n_clusters=4, init='k-means++', max_iter=100,
n_init=10, random_state=0)
pred_y = kmeans.fit_predict(X_std)
plt.scatter(X_std[:,0], X_std[:,1])
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[
:,
1], s=100, c='red')
plt.show()
```

Source code of the supported function of frequency prediction

```
import pandas as pd

#Read dataset
acc_data_for_fp = pd.read_csv('./dataset/preprocessed_acc_data.csv')

time_array1 = []
time_array2 = []
time_zone_array = []

for time_value1 in acc_data_for_fp['Time']:
    time_value_first_two_chars = time_value1[:2]
    time_array1.append(time_value_first_two_chars)

for time_value2 in time_array1:
    first_char_value = time_value2[0]
    second_char_value = time_value2[1]
    if second_char_value == ":" :
        time_array2.append(first_char_value)
    else:
        time_array2.append(time_value2)

for time_value3 in time_array2:
    int_time_value = int(time_value3)
    if (int_time_value >= 0) & (int_time_value < 7) :
        time_zone_array.append(1)
    elif (int_time_value >= 7) & (int_time_value < 13) :
        time_zone_array.append(2)
    elif (int_time_value >= 13) & (int_time_value < 19) :
```

```
        time_zone_array.append(3)
    elif (int_time_value >= 19) & (int_time_value < 24) :
        time_zone_array.append(4)

#add time zones to dataframe
acc_data_for_fp['time_zone'] = pd.Series(time_zone_array,
index=acc_data_for_fp.index)

#Convert dataframe to csv
acc_data_for_fp.to_csv
(r'./dataset/acc_data_for_frequency_prediction.csv', index = False,
header=True)
```

Source code for data preprocessing

```
import pandas as pd
#create a dataframe
from pandas import DataFrame

#Load the complete Accident dataset
acc_data_edited = pd.read_csv('edited_accident_vehicle_data.csv',
encoding = 'latin')

#Sampling data by selecting 22 features over 52 features
acc_data = acc_data_edited[['Latitude','Longitude',
'Urban_or_Rural_Area','1st_Road_Class', 'Speed_limit',
          'Road_Type', 'Road_Surface_Conditions', 'Weather_Conditions',
'Light_Conditions', 'Date',
          'Age_Band_of_Driver', 'Age_of_Vehicle', 'Junction_Detail',
'Junction_Location', 'X1st_Point_of_Impact',
          'make', 'Vehicle_Type',
'Vehicle_Manoeuvre','Did_Police_Officer_Attend_Scene_of_Accident','Time
','reasons','Accident_Severity'
         ]]

#Count number of records
num_of_records=len(acc_data)
print(num_of_records)

#Display dataset including features
print(acc_data.shape)
print(acc_data.head())

#Checking for sum of all null values
sum_of_nulls = sum(acc_data.isnull().sum())
print(sum_of_nulls)

#Checking for missing values in each column
count_of_nulls_each_column = acc_data.isnull().sum()
print(count_of_nulls_each_column)
```

```
#remove columns with many null values
accident_df2=acc_data[acc_data.columns[acc_data.isnull().mean() < 0.8]]
#remove rows with null values
accident_df3 = accident_df2.dropna(how='any',axis=0)
print(accident_df3.shape)
print(accident_df3.head())

#Again check for missing values for each column
column_null_count = accident_df3.isnull().sum()
print(column_null_count)

#Convert preprocessed data into csv file
accident_df3.to_csv (r'./dataset/preprocessed_acc_data.csv', index =
False, header=True)

#Count number of records
num_of_records=len(accident_df3)
print(num_of_records)
```