Accident severity and Classification

[1]U.Anjali[2]Mr.E.Ramesh Reddy

[1]CSE, NEC, Gudur
2Associate Professor,CSE Department, NEC, Gudur

Abstract: *Traffic accidents have become severe risks as they are one of the causes of enormous deaths worldwide. Reducing the number of incidents is critical to saving lives and achieving sustainable cities and communities. Machine learning and data analysis techniques interpret the reasons for car accidents and propose solutions to minimize them. However, this needs to take the benefits of big data solutions as the size and velocity of traffic accident data are increasingly large and rapid. This paper explores road car accident data patterns and proposes a predictive model by investigating meaningful data features, such as accident severity, the number of casualties, and the number of vehicles. Therefore, a pre-processing model is designed to convert raw data using missing and meaningless feature removal, data attribute generalization, and outlier removal using interquartile. Four classification methods, including decision trees, random forest, multinomial logistic regression, and naïve Bayes, are used and evaluated to study the performance of road accident prediction. The results address acceptable levels of accuracy for car accident prediction except for naïve Bayes. The findings are discussed through a data-driven approach to understand the factors influencing road car accidents and highlight the key ones to propose accident prevention solutions. Finally, some strategies are provided to achieve healthy and community-friendly cities.*

Keywords: functional testing; non functional testing; deployment; insert

1.INTRODUCTION

According to the death statistics released by the World Health Organization, the number of traffic accidents occurring annually in the world is alarming. The traffic accidents killed 1.2 million people each year and 50 million people were injured. Approximate 3,300 people were killed and 137,000 people were injured every day. Direct economic losses of 43 billion dollar, the frequent occurrence of traffic accidents directly threaten human life and property safety. Road accident prediction is one of the most important research area in traffic safety. The occurrence of road traffic accidents is mainly affected by geometric characteristics of road, traffic flow, characteristics of drivers and environment of road. Many studies have been conducted to predict accident frequencies and analyze the characteristics of traffic accidents, including studies on hazardous location/hot spot identification, accident injury-severities analysis, and accident duration analysis.

Some studies focus on mechanism of accidents. Other factors include weather and light conditions of the road. Lee et al developed a probabilistic model relating significant crash precursors to changes in crash potential. Abdel built a previous crash prediction model with the matched case-control logistic regression technique. No specific approach available for the traffic police to predict which area is accident prone at a specific time. The traffic accident prediction play an important role in the integrated planning and management of traffic, the reason which with much randomness about the traffic accident include some nonlinear elements, such as people, car, road, climate and so on. The traditional way of linear analyses can not reveal the really situation since the noise pollution and amount of data are too little, cause the result of prediction can not satisfactory. Because of the traditional BP network have some defects, such as local minimum, too many iterations, training too slow and so on. The traditional Back propagation network has defects. It has a 7.8% lower accuracy than

the proposed model

FUNCTIONAL TESTING

There are four main types of functional testing.

1) Unit Testing

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution.

Unit testing is important because we can find more defects at the unit test level.

For example, there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

a) White Box Testing

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

b) Gorilla Testing

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

For example, the tester is testing the pet insurance company's website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let's say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

2) Integration Testing

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems. **For example,** a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

a) Gray box testing

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

3) System Testing

System testing is types of testing where tester evaluates the whole system against the specified requirements.

a) End to End Testing

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.

b) Black Box Testing

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

c) Smoke Testing

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists. The testing team will ensure that the build is stable, and a detailed level of testing will be carried orther.

For example, tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

d) Sanity Testing

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

For example, a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

e) Happy path Testing

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

f) Monkey Testing

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

The objective of Monkey Testing is to check if an application or system gets crashed by providing random input

values/data. Monkey Testing is performed randomly, no test cases are scripted, and it is not necessary to be aware of the full functionality of the system.

4) Acceptance Testing

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

a) Alpha Testing

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

For example, the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

b) Beta Testing

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment** before releasing the product to the market for the actual end-users.

Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

c) Operational acceptance testing (OAT)

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

The focus of the OAT is on the following points:

- Testing of backup and restore.
- Installing, uninstalling, upgrading software.

- The recovery process in case of natural disaster.
- User management.
- Maintenance of the software.

Non-Functional Testing

There are four main types of functional testing.

1) Security Testing

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.

It also checks how software behaves for any hacker's attack & malicious programs and how software is maintained for data security after such a hacker attack.

a) Penetration Testing

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.

Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

Notes: Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

2) Performance Testing

Performance testing is testing of an application's stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner, etc. are good tools available in the market.

IMPLEMENTATIONS

Data Preprocessing Module: This module focuses on preparing and refining the input data for the machine learning algorithms. It involves tasks such as data cleaning, handling missing values, normalization, and transforming the raw data from the Hadoop log dataset into a format suitable for effective learning by the algorithms.

Machine Learning Algorithm Implementation Module: This module encompasses the implementation of machine learning algorithms, including Decision Tree, Random Forest, and Naive Bayes. Each algorithm is configured and trained on the preprocessed data to learn and recognize patterns indicative of Hadoop log network attacks.

Dataset Integration Module: The system integrates multiple labeled datasets beyond the dataset to ensure a more comprehensive understanding of diverse attack patterns. This module facilitates the combination of various datasets, promoting a broader and more adaptable detection capability.

Dynamic Model Adaptation Module: To address the dynamic nature of threats, this module introduces dynamic and adaptive machine learning models. These models continuously evolve and learn from ongoing network behaviors, allowing the system to adapt to emerging threats in real-time and enhance its ability to detect novel attack patterns.

Scalability and Optimization Module: This module focuses on enhancing the scalability and efficiency of the system. It includes strategies for accommodating large-scale IoT networks, optimizing resource utilization, and incorporating lightweight algorithms suitable for deployment on resource-constrained devices. The goal is to ensure the system performs effectively across diverse network environments while efficiently managing computational resources

.SOFTWARE DESCRIPTION:

What is Python programming language?

Python is a **high-level, general-purpose, interpreted** programming language.

1) High-level

Python is a high-level programming language that makes it easy to learn. Python doesn't require you to understand the details of the computer in order to develop programs efficiently.

2) General-purpose

Python is a general-purpose language. It means that you can use Python in various domains including:

- Web applications
- Big data applications
- Testing
- Automation
- Data science, machine learning, and AI
- Desktop software
- Mobile apps

The targeted language like SQL which can be used for querying data from relational databases.

3) Interpreted

Python is an interpreted language. To develop a Python program, you write Python code into a file called source code.

To execute the source code, you need to convert it to the machine language that the computer can understand. And the Python **interpreter** turns the source code, line by line, once at a time, into the machine code when the Python program executes.

Compiled languages like Java and C# use a **compiler** that compiles the whole source code before the program executes.

Why Python

Python increases your productivity. Python allows you to solve complex problems in less time and fewer lines of code. It's quick to make a prototype in Python.

Python becomes a solution in many areas across industries, from web applications to data science and machine learning.

Python is quite easy to learn in comparison with other programming languages. Python syntax is clear and beautiful.

Python has a large ecosystem that includes lots of libraries and frameworks.

Python is cross-platform. Python programs can run on Windows, Linux, and macOS.

Python has a huge community. Whenever you get stuck, you can get help from an active community.

Python developers are in high demand.

History of Python

- Python was created by Guido Van Rossum.
- The design began in the late 1980s and was first released in February 1991.

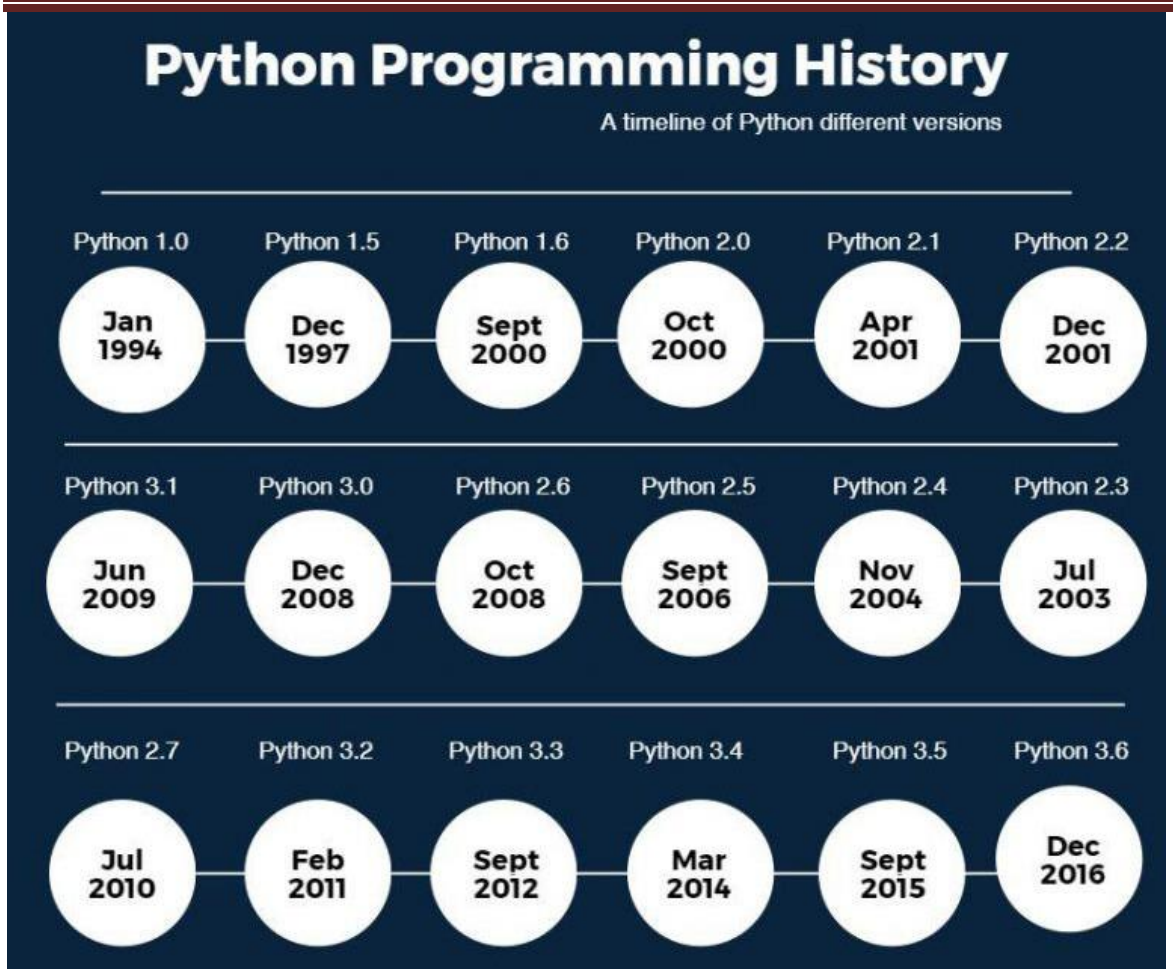
Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late 70s. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

Python Version History

Implementation started - December 1989

Internal releases – 1990



Install Python on Windows

First, download the latest version of Python from the download page.

Second, double-click the installer file to launch the setup wizard.

In the setup window, you need to check the **Add Python 3.8 to PATH** and click Install Now to begin the installation.

DEPLOYMENT DIAGRAM:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

The deployment diagram maps the software architecture created in design to the physical system architecture that executes it. In distributed systems, it models the distribution of the software across the physical nodes.

The software systems are manifested using various artifacts, and then they are mapped to the execution environment that is going to execute the software such as nodes. Many nodes are involved in the deployment diagram; hence, the relation between them is represented using communication paths.

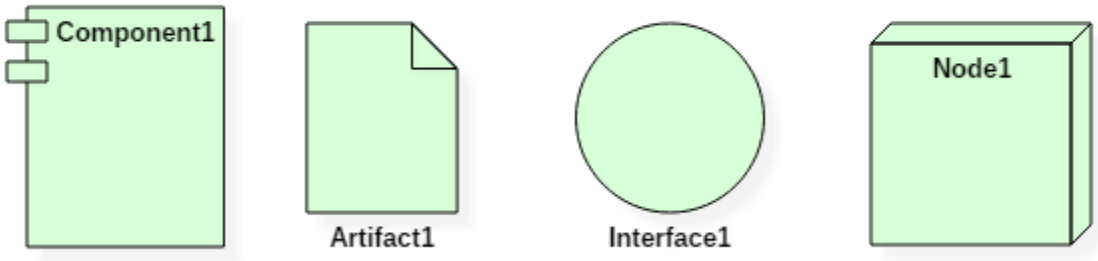
There are two forms of a deployment diagram.

- Descriptor form
  - It contains nodes, the relationship between nodes and artifacts.
- Instance form
  - It contains node instance, the relationship between node instances and artifact instance.
  - An underlined name represents node instances.

Purpose of a deployment diagram

Deployment diagrams are used with the sole purpose of describing how software is deployed into the hardware system. It visualizes how software interacts with the hardware to execute the complete functionality. It is used to describe software to hardware interaction and vice versa.

Deployment Diagram Symbol and notations



Deployment Diagram Notations

: 2.CONCLUSION

paper has covered the need to reduce traffic accidents since the number of accidents produced is still relatively high and causes significant economic expenditure. To find insights into where and how these accidents could be reduced, the project aimed to generate predictive models to understand the causes of the accidents. We determined the independent determinants from the extensive literature review (see Section 2), including weather, light, road surface conditions, and others. We first had to drop unnecessary columns that were not needed for the prediction model from the dataset we used. Outlier data were dropped to make the data more robust to uncertainties and errors. Furthermore, we performed some pre-processing to convert some of these variables into more useful formats, such as latitude and longitude, into geographical regions for more general solutions. The output variables to predict are the number of vehicles, the number of casualties, and accident severity. They were treated as discrete labels due to the limited values and concentration around that small range. Since the dataset was large in volume, big data analysis is important to handle large datasets of various types and generate them quickly. The dataset used in this research is quite large, and therefore, big data analysis helps generate faster results. Additionally, big data analysis allows us to extract useful information from the datasets through classification to generate predictive models. By using the analyzed results, decision-makers can easily understand accident patterns, driver behavior, time of day, road and weather conditions causing traffic congestion and other key factors contributing to accidents, such as fatalities and serious injuries, thus improving traffic safety control strategies. They can also use predictive models to adopt new policies in road safety and accident prevention. Future work may include determining the determinant

factors from this dataset before developing predictive models using methods such as correlation matrix to find the significant dependent variables for the dataset.

3.REFERENCES

.World Health Organization. Road Traffic Injuries. Available online: https://www.who.int/news-room/fact-sheets/detail/roadtraffic-injuries (accessed on 20 June 2021).

2. International Transport Forum. Road Safety Annual Report 2015. Road Saf. Annu. Rep. 2015, 486. [CrossRef]

3. Wang, Y.; Zhang, W. Analysis of Roadway and Environmental Factors Affecting Traffic Crash Severities. Transp. Res. Procedia 2017, 25, 2119–2125. [CrossRef]

4. Wang, J.; Lu, H.; Sun, Z.; Wang, T.; Wang, K. Investigating the Impact of Various Risk Factors on Victims of Traffic Accidents. Sustainability 2020, 12, 3934. [CrossRef]

5. Potoglou, D.; Carlucci, F.; Cirà, A.; Restaino, M. Factors associated with urban non-fatal road-accident severity. Int. J. Inj. Control. Saf. Promot. 2017, 25, 303–310. [CrossRef] [PubMed

6. Alomari, E.; Mehmood, R.; Katib, I. Road Traffic Event Detection Using Twitter Data, Machine Learning, and Apache Spark. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 1888–1895. [CrossRef]

7. Asor, J.R.; Catedrilla, G.M.B.; Estrada, J.E. A Study on the Road Accidents Using Data Investigation and Visualization in Los Baños, Laguna, Philippines. In Proceedings of the 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 6–7 March 2018; pp. 96–101. [CrossRef]

8. Aziz, K.; Zaidouni, D.; Bellafkih, M. Real-Time Data Analysis Using Spark and Hadoop. In Proceedings of the 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Morocco, 26–27 April 2018; pp. 1–6. [CrossRef]

9. Fisher-Hickey, D. 1.6 million UK Traffic Accidents. 2021. Available online: https://www.kaggle.com/daveianhickey/2000-16- traffic-flow-england-scotland-wales (accessed on 20 June 2021).

10. Li, Y.; Xing, L.; Wang, W.; Wang, H.; Dong, C.; Liu, S. Evaluating impacts of different longitudinal driver assistance systems on reducing multi-vehicle rear-end crashes during small-scale inclement weather. Accid. Anal. Prev. 2017, 107, 63–76. [CrossRef] 11. Nofal, F.; Saeed, A. Seasonal variation and weather effects on road traffic accidents in Riyadh City. Public Health 1997, 111, 51–55. [CrossRef] 12. Hammad, H.M.; Ashraf, M.; Abbas, F.; Bakhat, H.F.; Qaisrani, S.A.; Mubeen, M.; Fahad, S.; Awais, M. Environmental factors affecting the frequency of road traffic accidents: A case study of sub-urban area of Pakistan. Environ. Sci. Pollut. Res. 2019, 26, 11674–11685. [CrossRef]