CMPE 272 – Enterprise Software Platforms

Accident Chronicles Analysis with Big Data

Presented by

Vishnu Vardhan Reddy Ireddy 016816176

Vani Vineela Aremanda 016131284

Under the guidance of

Prof. Vidya Charan Bhaskar

Table of Contents

1 Introduction

1.1 Objective

Car accidents in the United States represent a major issue, impacting millions of individuals both socially and economically. A thorough analysis of accident patterns can yield valuable insights, which are instrumental in developing preventive strategies and shaping effective policies. By understanding the underlying factors of these accidents, we can identify trends and conditions that contribute to their occurrence, enabling us to propose targeted solutions to reduce their frequency and severity.

1.2 Literature Review

We delved into key publications that significantly influenced our analytical approach and methodology. A pivotal work in our research was "Data Quality Challenges in Road Accident Data Analysis" by Patel and Morris (2020). This insightful study sheds light on the inherent complexities and intricacies involved in road accident data analysis, particularly emphasizing the criticality of maintaining high data quality for accurate analytics. The insights gleaned from Patel and Morris's work were instrumental in guiding our decision to employ MapReduce for our analysis.

Further enriching our research was the resourceful book "Hadoop in Action," which serves as a testament to Hadoop's capabilities in swiftly and efficiently processing substantial datasets. This book not only underscored Hadoop's adeptness in handling large volumes of data but also its prowess in achieving rapid data processing, enabling us to extract timely and relevant insights.

These scholarly works collectively underscore the emerging trend of applying advanced Big Data technologies in the realm of traffic data analysis. They highlight the importance of not just the sheer volume of data but also the speed and accuracy with which it is processed, to derive meaningful and actionable insights. These insights are crucial in our endeavor to understand and mitigate the factors contributing to road accidents, thereby contributing to safer and more efficient traffic systems.

The dataset employed in this project encompasses all 50 states of the USA, covering a period from March 2019 to March 2023. It compiles data from a variety of sources, including the US and state departments of transportation, law enforcement agencies, traffic cameras, and sensors embedded in the road network.

This dataset “A Countrywide Traffic Accident Dataset (2016 - 2023)” is extensive, containing approximately 3.5 million records of car accidents, thus providing a detailed and expansive overview of the accident scenario across the country. This comprehensive dataset is crucial for analyzing and understanding the patterns and common characteristics of these accidents, helping to identify key areas for improvement in road safety and traffic management.

2 System Design and Implementation Details

2.1 System description

Below sections contain more details on the system description like overview, model, and a detailed description of the same.

2.1.1 System Overview

In our system, Hadoop forms the backbone that supports the entire data analytics process for traffic patterns and accident prediction. Our system model is architected around Hadoop, a framework renowned for its proficiency in managing Big Data challenges. Hadoop's ability to distribute and process massive datasets across clusters of computers makes it an essential tool for handling data volumes that were once considered unmanageable. The dataset is stored in the Hadoop Distributed File System (HDFS), ensuring high availability and fault tolerance.

The MapReduce programming model, which simplifies data processing by decomposing large-scale tasks into smaller, manageable segments, allowing for concurrent processing and analysis. MapReduce stands out for its ability to efficiently process and aggregate complex datasets, offering an effective solution to the challenges of data quality and complexity highlighted in the study. Together, they enable robust, scalable storage and efficient processing of large datasets within the Hadoop ecosystem. Figure 1 shows Hadoop's core components: HDFS, which manages data storage across the distributed cluster, and MapReduce, which processes the data stored in HDFS through parallel computation.

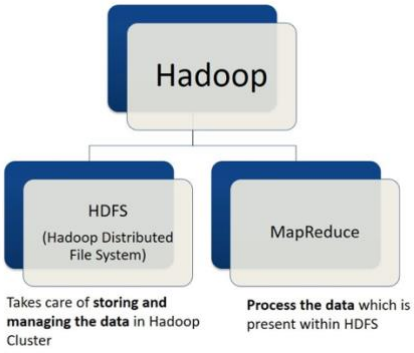

Figure 1

2.1.2 System Model

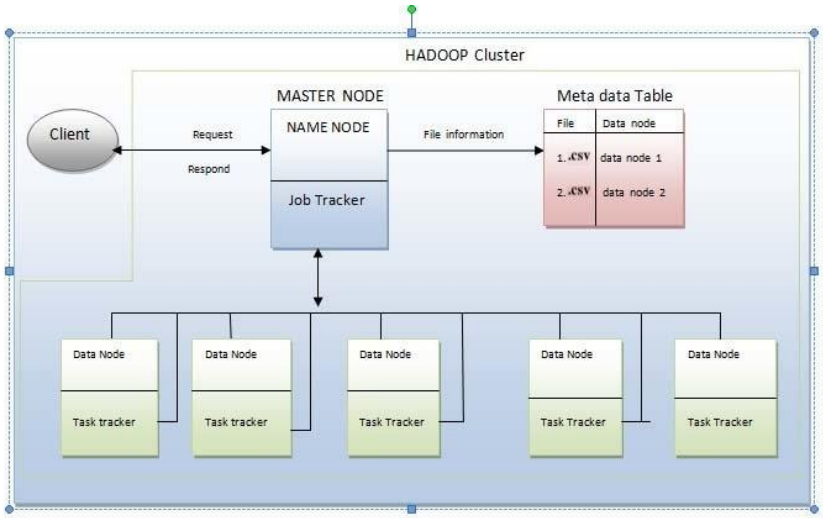

Figure 2

The system model in the Figure 2, showcases a Hadoop Cluster, a framework for processing large data sets across a distributed network of computers. The Client initiates data processing requests to the Master Node, which consists of the NameNode and the JobTracker. The NameNode handles the file system metadata, orchestrating where data is stored across the cluster without storing the data itself. The JobTracker manages the distribution and scheduling of processing tasks to the Data Nodes. Each Data Node contains a Task Tracker that executes tasks and communicates with the JobTracker. A MetaData Table indexes the data, detailing which files are stored on which Data Nodes, ensuring efficient data management.

2.1.3 Detailed System Description

The workflow begins with DriverClass.java, which sets up the job configuration and initiates the MapReduce process. From there, MapperClass.java takes over, processing records and emitting key-value pairs. It handles the transformation of input data into a format suitable for analysis. Following the mapping phase, the ReducerClass.java performs the aggregation of these intermediate key-value pairs. It is responsible for summarizing the data, which is a crucial step in extracting meaningful patterns and insights.

Optionally, AdditionalClass.java can be incorporated before the reduce step for pre-processing tasks, such as data cleansing or transformation, to improve the quality of input for the reduce phase. Finally, the output from the ReducerClass.java is written back to HDFS, where the results are stored and made available for further analysis or reporting.

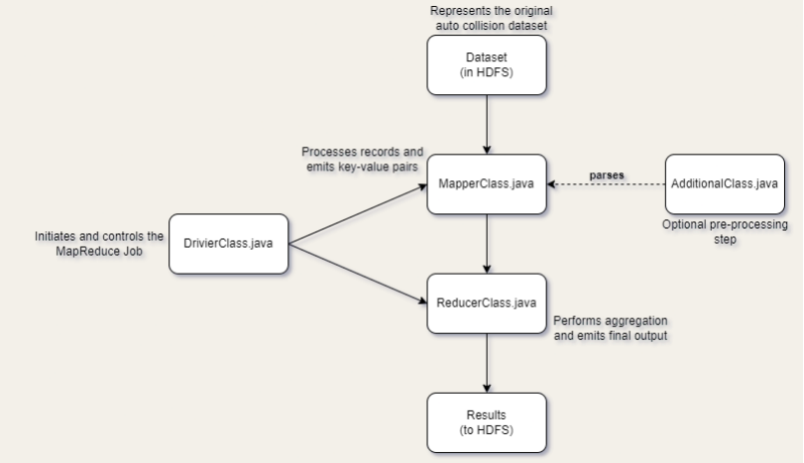

Figure 3

One of the critical advantages of using Hadoop in our model is the reduced processing delay, enabling faster insights from the data. This efficiency is particularly beneficial in real-time applications, such as accident prediction models, where timely data processing can lead to more immediate and actionable results. As shown in the Figure 3, our system showcases a streamlined and efficient process for handling and analyzing traffic data.

3 Problem Addressed

The gap in the literature and previous studies for the application of MapReduce with Hadoop in traffic analysis and accident prediction presented us with a unique opportunity. We capitalized on this by innovating within the traffic data analytics field, leveraging the distributed computing capabilities of Hadoop to analyze complex traffic datasets with improved efficiency and scale.

3.1 Hourly Analysis of Total Number of Accidents

Our comprehensive analysis scrutinizes accident data on an hourly basis to reveal critical insights into the temporal dynamics of road traffic accidents. By dissecting the total number of accidents occurring at each hour of the day, we aim to identify high-risk time slots when accidents are most prevalent. This granular temporal analysis is pivotal for both road users and policymakers, as it can lead to targeted traffic monitoring and control measures during identified peak accident hours, as well as to raise awareness among drivers to exercise heightened caution during these times.

3.2 Weekday Distribution of Accidents

The weekday distribution of accidents is analyzed to discern any significant variances in accident rates from Monday through Sunday. This aspect of the analysis is geared towards understanding how the rhythm of the workweek, including rush hours, versus the relaxation of the weekend, influences the incidence of road accidents. Insights from this distribution can be instrumental in deploying traffic management resources more effectively and in designing weekly specific road safety campaigns that address the unique characteristics of traffic flow and driver behavior on different days.

3.3 Proximity to Landmarks or Facilities

The correlation between accident rates and the proximity to key landmarks or facilities is a focal point of our analysis. This seeks to unravel how the vicinity to high-traffic areas such as educational institutions, healthcare facilities, commercial complexes, and entertainment venues may affect the frequency of road mishaps. Such an analysis could be invaluable for urban planners and local authorities in implementing strategic design and signage, improving road infrastructure, and planning for better traffic management around these hotspots to minimize accidents.

3.4 Accidents based on Geographical State

A geographical analysis is conducted to map out the accident rates across the various states. This enables us to pinpoint regional trends and identify areas with disproportionately high incidences of accidents. Factors such as state-specific driving laws, road maintenance quality, local weather patterns, and the effectiveness of traffic law enforcement can be evaluated to understand their impact on road safety. This state-wise breakdown can aid in crafting localized interventions and enhancing state-specific road safety regulations.

3.5 Monthly Patterns of Accidents

By mapping the monthly patterns of accidents, we aim to elucidate any seasonal or periodic trends that might influence accident occurrences. This part of the analysis can reveal how factors like seasonal weather conditions, holiday travel, or annual cultural events might contribute to fluctuations in accident rates. Recognizing these patterns allows for the anticipation of potential accident surges and the strategic allocation of emergency response resources, as well as the implementation of safety campaigns tailored to these cyclical changes.

4 Methodology Used

4.1 Tech Stack

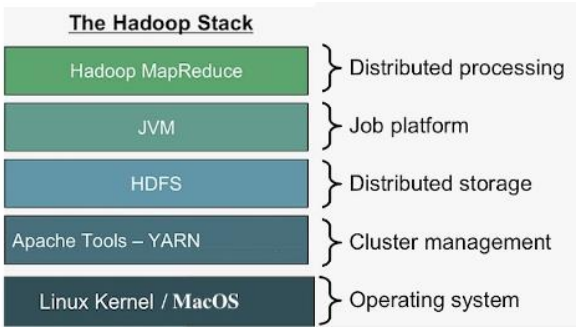

Figure 4

Hadoop MapReduce, the top layer, enables distributed data processing across clusters for scalable computation. The Java Virtual Machine (JVM) is the job execution platform, ensuring that applications run smoothly on any underlying system. The Hadoop Distributed File System (HDFS) provides distributed storage, allowing for fast data access across nodes. Cluster management is facilitated by Apache tools such as YARN, which allocates resources and schedules jobs. Underpinning the entire stack is the operating system, Linux or MacOS, serving as the interface for Hadoop components with the physical hardware. This stack is the backbone of our project, supporting our big data analytics capabilities. Figure 4 illustrates our technology stack, which is integral to the project.

4.2 Algorithm

The MapReduce algorithm implemented in this project consists of three primary components: `MapperClass`, `ReducerClass`, and `DriverClass`, along with an optional `AdditionalClass` for data handling. This algorithm is designed to process and analyze large datasets.

MapperClass

The `MapperClass` iteratively processes each record in the dataset. For each record, it parses the data into individual attributes and extracts relevant information, such as collision severity and location. This class then emits key-value pairs, with the location as the key and the collision severity as the value.

ReducerClass

The `ReducerClass` receives the sorted and grouped key-value pairs emitted by the Mapper. For each unique key, it initializes an accumulator and iterates over the values associated with that key. The class aggregates these values, typically by counting instances of each severity level at each location. Finally, it emits the key (location) along with the aggregated result (e.g., count of severities) as its output.

DriverClass

The `DriverClass` serves as the entry point of the MapReduce job. It sets up and configures the job, including specifying the job name, input and output formats, and the classes to be used for mapping and reducing. It also defines the input and output paths in the file system. Once configured, it submits the job to the MapReduce engine for execution.

AdditionalClass (Optional)

An optional `AdditionalClass` is included to define the structure of the dataset headers and provide utilities for parsing and validating record attributes. This ensures that the data fed into the MapReduce process is correctly formatted and reliable.

4.3 Pseudo Code

Mapper Class

```
class MapperClass {
    void Map(Text[] input) {// Auto collision records as text
        for (int i = 0; i < input.length; i++) {
            Text record = input[i];
            ParsedRecord parsedRecord = parse(record); // Parse the record into its attributes
            RelevantInfo relevantInfo = extractRelevantInfo(parsedRecord); // Extract relevant
information like collision severity, location
            emit(relevantInfo.location, relevantInfo.severity); // Emit a key-value pair with location
as key and severity as value
        }
    }
}
```

Reducer Class

```
class ReducerClass {
    void Reduce(KeyValuePair[] input) {// Key-value pairs sorted and grouped by key
        for (int i = 0; i < input.length; i++) {
            Key currentKey = input[i].key;
            int counter = 0; // Initialize a counter or accumulator
            for (int j = 0; j < input[i].values.length; j++) {
                Value val = input[i]. values[j];
                counter += aggregate(val); // Aggregate values (e.g., count severity instances)
            }
            emit(currentKey, counter); // Emit the key and the aggregated result as output
        }
    }
}
```

Driver Class

```
class DriverClass {
    public static void main(String[] args) {
        JobConfig jobConfig = new JobConfig(); // Initialize Job Configuration
        jobConfig.setJobName("JobName");
        jobConfig.setInputFormat(InputFormatClass.class);
        jobConfig.setOutputFormat(OutputFormatClass.class);
        jobConfig.setMapperClass(MapperClass.class);
        jobConfig.setReducerClass(ReducerClass.class);
        jobConfig.setInputOutputPath("inputPath", "outputPath");
        submitJob(jobConfig); // Submit the job to the MapReduce engine
    }
}
```

Additional Class (Optional)

```
class AdditionalClass {
    ParsedRecord parseRecord(Text record) {
        // Implementation for parsing a record
    }

    boolean validateRecord(Text record) {
        // Implementation for validating a record
    }
}
```

5 Results and Analysis

Hourly Analysis

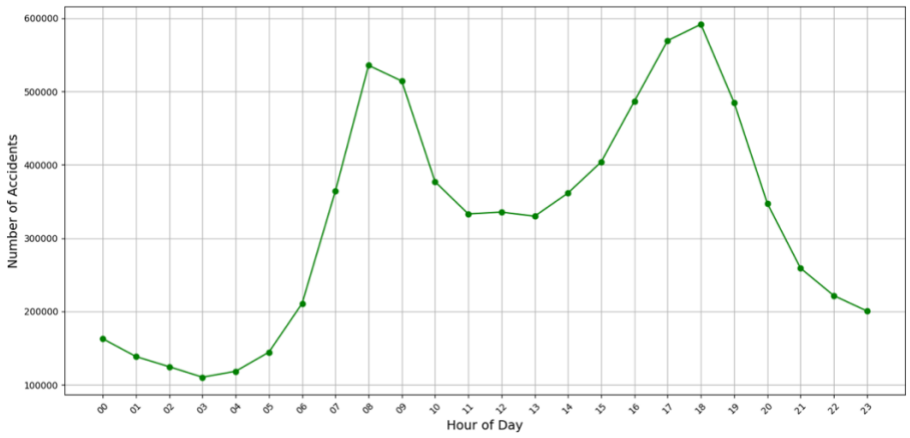

Figure 5: This line graph shows the number of accidents plotted against the hour of the day. The x-axis ranges from 0 to 23, representing a 24-hour time cycle, while the y-axis shows the number of accidents, ranging from approximately 100,000 to 600,000.

In conclusion from our analysis, it is identified that a distinct pattern with two prominent peaks, one in the late morning and another in the evening. There is a noticeable dip in the early hours of the day and mid-afternoon. This pattern may suggest that accidents are more frequent during typical rush hour periods, potentially due to increased traffic volume, and less frequent during the early morning hours when there is likely less traffic on the roads.

Weekly Distribution

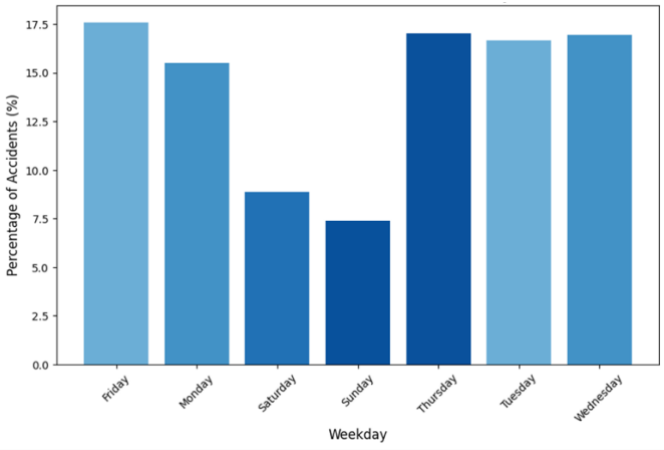

Figure 6: The bar graph depicts the percentage of accidents that occur on each day of the week. The x axis lists the days from Friday to Wednesday, and the y-axis represents the percentage of accidents, ranging from 0% to slightly over 17.5%

In conclusion from our analysis, it is comprehended that the highest percentage of accidents occur on Friday, followed by a notable decrease on Saturday and Sunday. The percentage rises again from Monday through Wednesday. This data could suggest that accidents are more likely at the end of the traditional workweek, with a decrease during the weekend, potentially due to less commuter traffic, and then increasing again as a new workweek begins.

Proximity to Landmarks or Facilities

In conclusion from our analysis, it is noticed that smaller percentages for accidents near other facilities such as roundabouts, stations, stops, traffic calming measures, crossings, and amenity bumps. This analysis suggests that the majority of accidents occur at or near traffic signals, junctions, and give ways, which are typically areas with higher vehicular movement and complex driving maneuvers, potentially contributing to a greater likelihood of accidents.

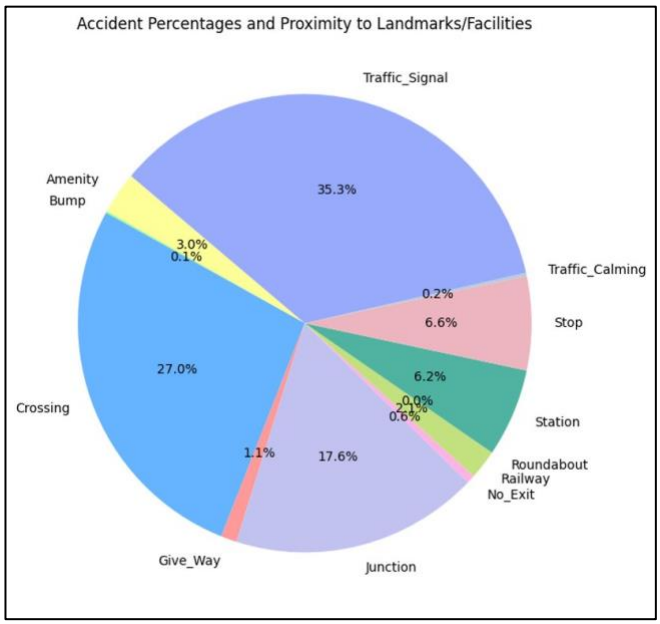

Figure 7: The pie chart illustrates the distribution of accident percentages in relation to their proximity to various landmarks or facilities. Traffic signals account for the highest percentage of accidents at 35.3%, followed by junctions with 27.0%

Accidents filtered by State.

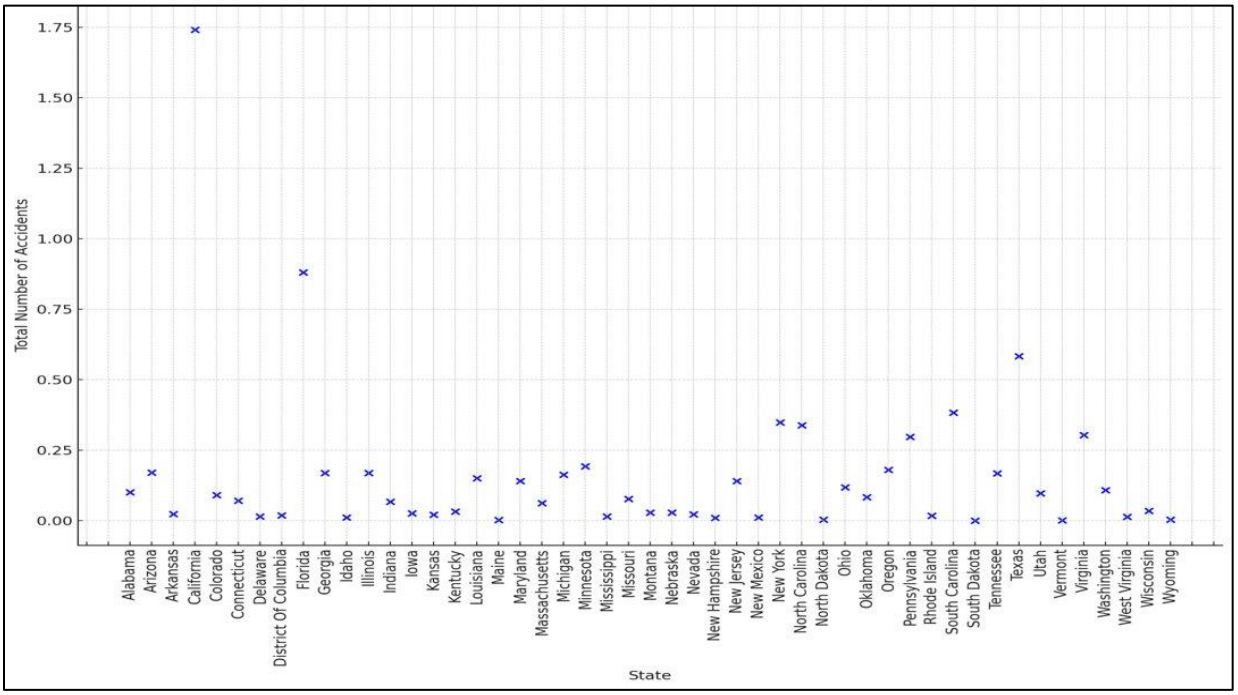

Figure 8: This scatter plot details the total number of accidents occurred by state. The x-axis lists the states in alphabetical order, while the y-axis indicates the total number of accidents, scaling up to 1.75 million.

Each point in the plot represents a state's total accident count. In conclusion from our analysis, it is apparent that there is significant variation between states. A few states have notably higher numbers of accidents, as indicated by points that are much higher on the y-axis. These outliers suggest that certain states have a higher prevalence of accidents, which could be due to a variety of factors such as population density, amount of daily traffic, or number of roadways. The majority of states appear to have a relatively lower and more consistent number of accidents. This scatter plot is a useful tool for identifying patterns and outliers in state-wise traffic accident data.

Monthly Distribution

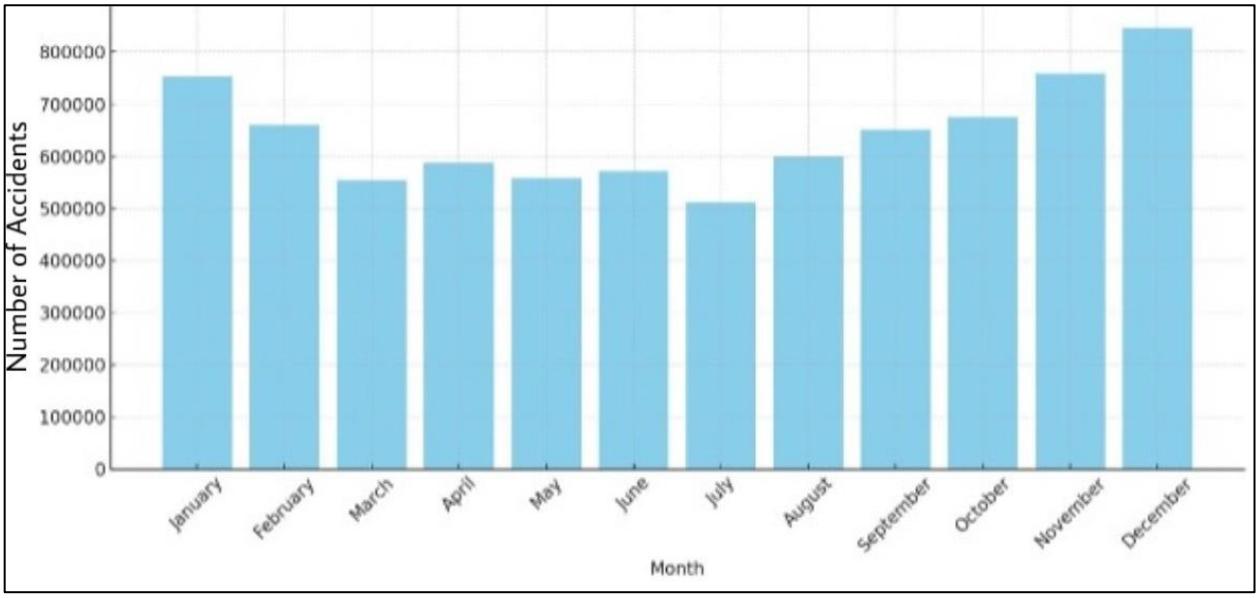

Figure 9: The bar graph represents the number of accidents that occurred during a month. The x-axis categorizes the data by month, from January to December, and the y-axis shows the number of accidents, ranging from around 400,000 to 800,000

There appears to be a seasonal trend with varying numbers of accidents throughout the year. In conclusion from our analysis, it is understood that the number of accidents rises and falls at various times of the year, with the highest peaks occurring in the months of March and October. These peaks may correlate with seasonal changes, holiday periods, or other monthly factors that can influence driving conditions or travel behavior. The lowest number of accidents appears to be in February, which could be attributed to it being the shortest month or possibly due to other factors such as weather conditions that discourage driving. Overall, the chart suggests a fluctuation in accident rates throughout the year, which could be valuable for planning safety campaigns or allocating resources for traffic management.

6 Conclusion

Our analysis of the data using Hadoop's MapReduce framework has provided valuable insights into nationwide traffic accident patterns. The project demonstrates the power of Big Data tools in extracting meaningful information from vast datasets, which can inform better traffic management strategies.

The data analyzed through the MapReduce job has provided insightful revelations on the relationship between accident occurrences and their proximity to various landmarks and facilities. The results indicate that certain areas, particularly those around traffic signals, junctions, and give ways, are more prone to accidents. Seasonal and daily patterns also emerged, with higher accident rates observed during peak traffic hours and specific months of the year. These findings underscore the complex interplay between traffic dynamics and accident rates and highlight the potential of big data analytics in enhancing our understanding of traffic safety.

7 Future Work

Future enhancements may include real-time streaming data analysis, integration with IoT devices for live traffic updates, and the use of machine learning for predictive analytics. The potential for further application of Hadoop in traffic data analysis remains vast and untapped.

The foundational work conducted in this project opens several avenues for future exploration and application. Integrating the insights from the MapReduce job with live traffic systems stands out as a promising extension. By doing so, real-time traffic management could be enhanced, potentially reducing accident rates by providing timely warnings and improving traffic flow around identified high-risk areas.

Furthermore, the application of this data in the development of autonomous vehicles is of particular interest. The nuanced understanding of accident patterns in relation to landmarks and facilities could be instrumental in training the algorithms that govern the behavior of autonomous vehicles. This could lead to safer navigation and decision-making by these systems, especially in complex urban environments.

As a part of continuous improvement, future iterations of the MapReduce job could include more granular data, such as weather conditions, driver behavior, and traffic density. The integration of machine learning models to predict potential hotspots for accidents before they occur could also be a significant step forward. Overall, the ongoing enhancement of data analytics capabilities will remain crucial in our quest to improve road safety and traffic efficiency.

8 Appendix

GitHub Repo: https://github.com/VardhanReddy2/CMPE-272-group-project

9 References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008.

[2] K. Pahlavan and P. Krishnamurthy, "Principles of Wireless Networks – A Unified Approach", Prentice Hall, 2002.

[3] M. E. Niedermeyer and A. Devarajan, "On the Robustness of Roadside Accident Data for Use in Autonomous Vehicle Control Algorithms", IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 5, pp. 1923-1934, 2019.

[4] F. R. da Silva and S. Han, "Traffic Accident Analysis Using Machine Learning Paradigms", Informatica, vol. 24, no. 1, pp. 61-80, 2000.

[5] A. Fernandes and J. P. Sousa, "Intelligent Traffic Systems for Accident Prevention and Reduction", Journal of Advanced Transportation, vol. 50, no. 8, pp. 1549-1567, 2016.

[6] T. Litman, "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning", Victoria Transport Policy Institute, 2020.

[7] H. Xia, P. Zhang, B. Li, and Z. Xu, "Real-Time Traffic Signal Control for Urban Traffic Networks Based on Big Data Analysis", IEEE Access, vol. 7, pp. 18665-18675, 2019.

[8] D. Watzenig and M. Horn, "Automated Driving: Safer and More Efficient Future Driving", Springer, 2017.

[9] S. K. Khisty and B. K. P. Horn, "Reliability Analysis of Traffic Engineering Data for Intelligent Transportation Systems", Journal of Transportation Engineering, vol. 122, no. 3, pp. 200-207, 1996.

[10] L. A. Klein, S. Toth, and B. S. Kerner, "Traffic Theory and Modeling for Advanced Traffic Management Systems", IEEE Transactions on Intelligent Transportation Systems, vol. 2, no. 3, pp. 149-158, 2001.

[11] Y. Zheng, "Trajectory Data Mining: An Overview", ACM Transactions on Intelligent Systems and Technology, vol. 6, no. 3, Article 29, 2015.

[12] C. Chen, K. Kockelman, and B. Khanal, "Locating Traffic Signal Controls using High-Resolution Traffic Accident Data", Accident Analysis & Prevention, vol. 47, pp. 66-75, 2012.