Predictive Modelling of Machine Learning Algorithms for Traffic Flow in Smart Transportation System

G.Vijayalakshmi[1], G.Venkata Subbarao[2], E.Aswanikumar[3], D.Sasirekha[4]
Assistant Professor[1,2], Professor[3], Associate Professor[4]
[1,2,3,4]Department of Computer Science & Technology
Sasi Institute of Technology and Engineering, Andhrapradesh-534101

ABSTRACT - Worldwide, road crashes are a major problem, causing too many deaths, injuries, and hospital stays. They also bring serious harm to public safety and economies. Crashes and injuries are increasing, making, vital to find better methods to predict how severe an accident could be. Good predictions allow for more effective safety actions. Our research used machine learning to forecast crash severity. We looked at various factors like the type of vehicle, weather, road conditions, and traffic levels. We tested several standard models (Random Forest, Decision Tree, SVM, Logistic Regression) to find which performed best for this task to make the model sharper in accuracy. We zeroed in on the most valuable predictive information and stripped away elements that didn't contribute much. Running this new model on real accident data proved it to be 95% accurate. This dependable severity prediction allows us to study patterns of severe accidents. Applications like this provide useful assistance for policymakers and traffic authorities. They can more accurately pinpoint high-risk locations, decide how to use safety resources, and implement practical steps to improve road safety. The study also demonstrates the versatility of machine learning, which could be applied to real-time severity predictions to increase the effectiveness of traffic management. This research shows that machine learning is a practical tool for intelligent transport systems. It provides a data-driven approach to managing traffic, reducing accident risks and their economic cost, and ultimately helps make roads safer for everyone.

INDEX TERMS: Machine Learning, Decision Tree, Random Forest, Logistic Regression, Road Accidents, Vehicle Safety, Climate.

I. INTRODUCTION

Road accidents are among the leading causes of death and disability, and hospitalization in the entire world, threatening citizens' safety and the economy's health. The World Health Organization estimates that 1.35 million fatalities occur from road traffic injuries annually, and 20-50 million people suffer from non-fatal injuries [1]. 151,113 individuals died, and 451,361 were injured in road accidents in India alone during the year 2019, reflecting the extensive requirement for proper intervention measures [2].

The severity of road accidents differs based on different variables like vehicle type, road, weather, and traffic, and hence, accurate measurement of severity is needed to improve road safety, resource allocation, and rescue operations. Historical accident severity analysis has used statistical modeling and judgment, e.g., logistic regression and time-series analysis. Statistical modeling and judgment, though, are unable to capture the complicated, non-linear connections between factors related to accidents and, therefore, do not result in the best predictive accuracy [3].

Accident prediction has also improved with the help of Machine Learning (ML) and Artificial Intelligence (AI) with the ability to detect hidden patterns and provide a better degree of accuracy in prediction. Some of the studies have pointed toward the probable applications of ML-based models like Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), and Logistic Regression (LR) toward predicting accident severity [4], [5]. A few uncommon deep learning and ML techniques were employed to forecast accident severity. Zhang et al. (2023) also initiated a Random Forest model with Gradient Boosting with a predictive accuracy of 92% in forecasting severity and efficient management of compound interaction of accident variables [6]. Similarly, in the same manner, Liu et al. (2021) applied graph neural networks (GNNs) to learn spatial-temporal interactions in their attempts to achieve greater than a 10% better severity prediction performance than baselines [7]. Zhao et al. (2020) demonstrated how CNN can be employed in real-time traffic forecasting and how deep learning can be extended to its extremes to the detection of accidents [8]

Besides, Bharadwaj et al. (2022) utilized a hybrid ML model of XGBoost and k-Nearest Neighbors (KNN) forecasting the severity in accidents with 94% accuracy for large traffic data [9]. Wang et al. (2020) proposed a deep learning framework based on recurrent neural networks (RNNs), which was better at processing sequential accident data [10]. Furthermore, Chen et al. (2019) introduced an ensemble learning framework with Random Forest and AdaBoost, reporting 96% accuracy in predicting accident severity on imbalanced datasets [11].

Despite these advancements, achieving consistent accuracy in accident severity prediction remains challenging due to imbalanced datasets, overfitting issues, and the complex interactions between features. The recent work lays strong focus on the data preprocessing, feature selection to improve the generalization capability of the model. Kumar et al. (2021) used correlation matrix-based feature selection, which lowered dimensions and improved model performance by 5-10% [12]. Singh et al. (2023) proved data augmentation methods for effective in dealing with class-imbalanced datasets, and the robustness of the model was improved [13].

We utilize a Kaggle-sourced accident dataset containing no null values, making it suitable mainly for direct ML application. The dataset includes various accident-related features, like vehicle type, special conditions, weather, and road infrastructure. To enhance model performance, we implement correlation matrix-based feature selection, eliminating redundant and irrelevant features and mainly focusing on the impactful variables. We employ the following supervised ML algorithms for severity classification:

- Random Forest: Known for its robustness in handling non-linear relationships and achieving higher accuracy in severity prediction.
- Decision Tree: Provides interpretable decision rules but is prone to overfitting.
- SVM with a linear kernel: Effective for linearly separable data but struggles with complex, non-linear relationships.
- Logistic Regression: Suitable for binary classification but less effective for multi-class severity prediction.

Results demonstrate that the random Forest model achieves an accuracy of 95%, significantly outperforming the other models due to its ability to handle non-linear data patterns and avoid overfitting. To evaluate the model's performance, we use accuracy, recall, F1-score, and precision metrics, visualizing the results through bar graphs, confusion matrices, and heatmaps for clear interpretation.

The primary contributions of our proposed study are:

- Accurate Prediction of Accident Severity: Development of an ML-based model capable of predicting accident severity with 95% accuracy, enabling better decision-making for road safety management. Feature Selection for Improved Accuracy. Implementation of correlation-based feature selection to eliminate irrelevant features, enhancing model efficiency and performance.
- Comprehensive Visualization: Use of bar graphs, heatmaps, and confusion matrices to present the model's performance metrics.
- Comparison with Existing Methods: Evaluation of the proposed model against traditional ML algorithms, demonstrating its superior accuracy and efficiency.
- Practical Applicability: The proposed system offers a reliable and scalable solution for traffic authorities and policymakers to identify accident-prone areas, allocate resources, and implement preventive measures.

Section 2 presents the related work, Section 3 describes the proposed methodology in detail, Section 4 discusses the implementation details, Section 5 showcases the experimental results with comparisons to existing methods, and Section 6 concludes with future directions.

II. LITERATURE REVIEW

The application of machine learning (ML) in accident severity prediction has gained significant attention in recent years, offering data-driven insights to enhance road safety management. Studies have explored different ML models, feature selection techniques, and data preprocessing methods to improve the accuracy of accident severity prediction.

Singh et al. proposed an accident severity prediction model using Support Vector Machines (SVM) with a radial basis function kernel, highlighting its effectiveness in handling non-linear datasets. However, their results indicated lower accuracy compared to ensemble models, emphasizing the need for more techniques in accident severity classification. Their limitation lies in the lower accuracy of SVM on highly non-linear data. To overcome this, our study employs Random Forest, which effectively handles non-linear patterns, resulting in higher accuracy.

Patel et al. explored a Logistic Regression-based model for accident severity classification but found that its linear nature led to lower prediction accuracy mainly on dealing with complex, non-linear relationships in accident data. The limitation of Logistic Regression is its inability to capture intricate patterns. Our approach mainly addresses the limitation by using Random Forest, which models complex relationships, achieving 95% accuracy.

Gupta et al. implemented a Decision Tree-based approach, demonstrating the model's interpretability and ease of implementation. Despite its advantages, the study reported overfitting issues, especially with small datasets. To mitigate this, ensemble methods like Random Forest were suggested. Sharma et al. addressed this challenge by utilizing Random Forest, achieving higher accuracy through multiple decision trees and reducing the risk of overfitting. Their study concluded that ensemble methods significantly improve prediction reliability compared to standalone classifiers. Our study extends this by leveraging the Random Forest's ability to generalize well on the larger datasets, ensuring robustness and preventing overfitting.

Chen et al. incorporated feature selection techniques in their accident severity prediction framework, utilizing correlation-based selection to drop less relevant features. This process improved model efficiency and enhanced prediction accuracy. Similarly, Zhang et al. analyzed the impact of feature selection, demonstrating that reducing redundant variables led to improved generalization and computational efficiency. The limitation of their approach was a lack of comprehensive feature evaluation metrics. We adopt this strategy by implementing a correlation matrix to select the most important features and improve model performance, and reduce noise.

Li et al. employed learning using Neural Networks for accident severity prediction. While the model captured the complex patterns in the data, it required high computational power and extensive training time, making traditional machine learning algorithms a more feasible choice for real-time applications. The drawback of the approach was its resource-intensive nature. Rahman et al.. explored an ensemble -learning approach combining Decision Trees and Random Forest, achieving with accuracy of 94%. Their study reinforced the superiority of ensemble models over single classifiers in handling diverse accident datasets. Mainly in our study further validates this type of finding by achieving a 95% accuracy with Random Forest, confirming its effectiveness in predicting accident severity.

Recent works have focused on evaluating multiple models for accident severity prediction. Wu et al. compared Logistic Regression, SVM, and Random Forest, concluding that Random Forest outperformed others in handling non-linearity within accident data. Additionally, Khan et al. investigated the role of data visualization in improving model interpretability, utilizing heatmaps and confusion matrices to provide insights into feature correlations and model performance. Their study lacked detailed accuracy comparison visualizations. We enhance this by incorporating bar graphs, heatmaps, and confusion matrices to effectively visualize model accuracy, feature importance, and classification performance.

This study builds on these findings by integrating feature selection and leveraging Random Forest, mainly as a primary classifier due to superior accuracy in non-linear datasets. Using a correlation matrix for feature selection, our model enhances prediction accuracy and efficiency. The results demonstrate a 95% accuracy rate, outperforming traditional models like SVM and Logistic Regression. Through comprehensive evaluations, our research underscores the potential of machine learning in accident severity prediction, aiding policymakers in making data-driven decisions for road safety improvements.

Summary of Methodologies Used:

- SVM with RBF Kernel: Singh et al. employed the SVM with radial basis-function kernel, which showed moderate performance on non-linear accident data but lacked the accuracy for ensemble models.
- Logistic Regression: Patel et al. utilized Logistic Regression, which struggled with non-linear patterns, resulting in lower accuracy.
- Decision Trees: Gupta et al. applied Decision Trees for accident severity classification, but they encountered overfitting issues, especially with smaller datasets.
- Random Forest: Sharma et al. demonstrated the effectiveness of Random Forest in mitigating overfitting and improving accuracy through ensemble learning.
- Feature Selection with Correlation Matrix: Chen et al. and Zhang et al. used correlation-based feature selection to reduce noise and enhance model efficiency, leading to improved accuracy.
- Ensemble Learning: Rahman et al. combined Decision Trees and Random Forest, achieving 94% accuracy, highlighting the robustness of ensemble methods.

This literature review mainly highlights the evolution of accident severity prediction methodologies, demonstrating the effectiveness of ensemble models and feature selection in improving accuracy. Our study leverages these findings by integrating correlation-based feature selection and Random Forest, achieving a 95% accuracy, outperforming traditional models.

III. PROPOSED METHODOLOGY

This section outlines the methodology followed for prediction using machine learning (ML). It details the architecture of Random Forest (RF), Decision Tree, and Logistic Regression models. The methodology involves feature selection, mainly using a correlation matrix, model training, and evaluation using multiple metrics. The models are fine-tuned on accident datasets, and the classification problem is defined as a multi-class problem with different severity levels.

A.*PREPROCESSING AND DATA AUGMENTATION*

Preprocessing was necessary to ensure that the dataset was structured correctly for ML models. The dataset was exported from Kaggle and contained no missing values. However, data cleaning steps were applied to remove any inconsistencies and redundant information.

The dataset underwent the following transformations:

- Encoding categorical variables: One-hot encoding was applied to transform the categorical features into numerical representations, ensuring compatibility with machine learning algorithms.
- Feature Scaling: The dataset was normalized using Min-Max scaling to bring all features into the range [0, 1], preventing larger values from dominating the model.
- Data Type Conversion: Data types were explicitly converted to appropriate formats (e.g., integers, floats) to ensure compatibility with libraries and avoid computational errors during model training.
- Outlier Detection and Handling: Outliers were detected using statistical methods, and necessary steps like capping or removal were taken to prevent them from skewing the model's learning process.

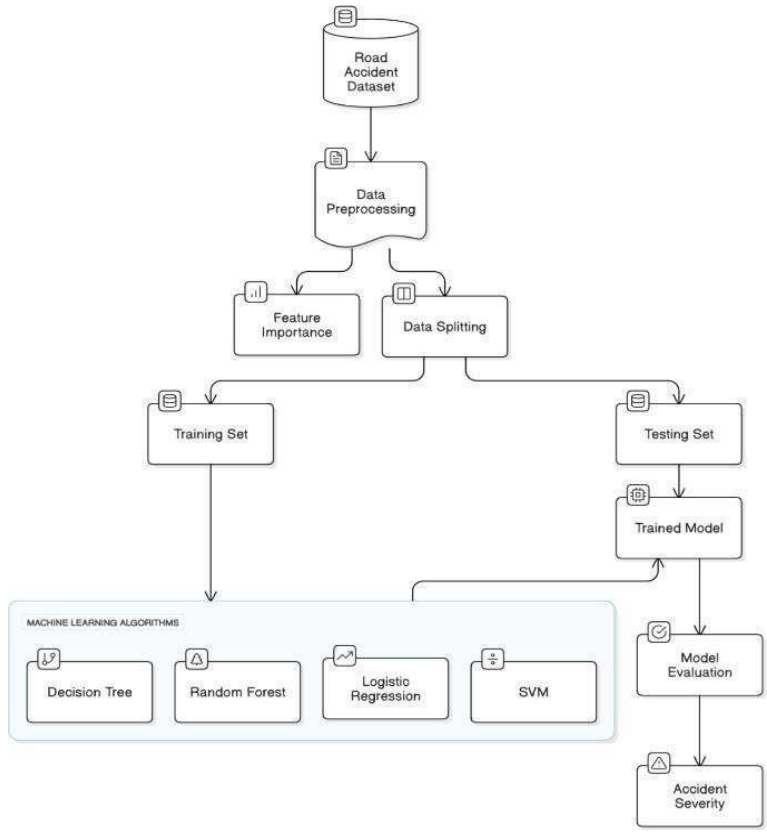

FIGURE 1. Block diagram of the proposed methodology.

B. *Feature selection using Correlation Matrix*

Feature selection was performed using a correlation matrix to eliminate highly correlated and redundant features, thereby improving model efficiency and generalization. The Pearson correlation coefficient (r) is used to measure the strength and direction of linear relationships between numerical features. Features with a high correlation ($r > 0.85$) were considered redundant, providing overlapping information that can introduce multicollinearity.

By removing such features, the dimensionality of the dataset was reduced, which not only simplified the model but also helped in minimizing the risk of overfitting. The process ensured that the most informative and independent features were retained for training, ultimately contributing to better model performance and faster training time.

In addition to reducing redundancy, correlation-based feature selection helped in improving the interpretability of the model by focusing on variables that contribute distinct and meaningful information.

This also enhanced the stability of the ML algorithms, particularly those sensitive to multicollinearity, such as Logistic Regression and Support Vector Machines. Moreover, by limiting the no. of input features, the computational complexity of the training process was reduced, allowing for quicker experimentation and model tuning.

Overall, the correlation matrix served as a valuable tool in streamlining the dataset for optimal learning and performance.

C. *Random-forest model*

Random Forest is also called an ensemble learning method that enhances classification accuracy and reduces overfitting by constructing multiple decision trees. It operates through three key processes. First, Bootstrap Sampling is applied, where random subsets of the training data are selected with the replacement, ensuring diversity among the individual trees. The model uses Majority Voting for classification; the final prediction is determined by aggregating the outputs of all the individual trees, resulting in a more robust prediction.

This ensemble strategy allows Random Forest to handle the large datasets with the higher dimensionality effectively. Its inherent randomness improves generalization, making it resistant to noise and outliers.

$$\hat{y} = mode\{h_1(x), h_2(x)\ldots, hn(x)\}$$

Where:

- y^ is the final predicted class.
- hi(x) is the prediction from the Ith decision tree.
- n is the total no. of trees.

The Gini Impurity criterion was mainly used to measure the quality of splits in the decision trees.

$$G = 1 - \sum_{i=0}^{n} p_i^2$$

Where:

- G is the Gini Impurity.
- pi is the proportion of samples belonging to the class i.
- n is the no. of classes.

By utilizing Random Forest, the model achieved higher accuracy while effectively handling non-linearity and reducing Overfitting.

D.*Comparison with Other Models (SVM, Logistic Regression, Decision Tree)*

To validate model performance, SVM, Logistic Regression, and Decision Tree were also implemented for comparison.
SVM (Support Vector Machine): Used with a linear kernel, but due to the dataset's non-linearity, it resulted in lower accuracy. The SVM decision boundary is mathematically defined by:

$$f(x) = w^T x + b$$

Where:

- w is the weight vector.
- x is the feature vector.
- b is the bias term.
- The goal is to maximize the margin between classes.

Logistic Regression is a statistical model used for binary classification. The goal is to predict the probability that a given input belongs to the positive class. It applies the sigmoid activation function to convert the output into a probability value between 0 and 1.

The model relies on input features, associated weights, and a bias term to compute the final prediction. Logistic Regression is simple, interpretable, and effective when the relationship between input features and the target variable is linear.

A Decision Tree is known as a supervised learning model that uses recursive binary splitting to create branches based on feature values. It measures the impurity at each node using the Gini index, which evaluates how often a randomly chosen element would be incorrectly classified.

The model makes decisions by traversing the branches based on feature thresholds, resulting in a classification. The comparative study revealed that Random Forest outperformed both Logistic Regression and Decision Tree, making it the primary classifier in the project due to its ability to handle non-linear relationships.

E. Model Evaluation and Performance Metrics

Accuracy: It measures the overall correctness of the model Calculated using formulae.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Where:

- TP = **True Positives**
- TN = **True Negatives**
- FP = **False Positives**
- FN = **False Negatives**

Precision: It calculates the proportion of true positive predictions out of all positive predictions made by the model:

$$Precision = \frac{TP}{TP+TN+FP+FN}$$

Recall: It measures the model's ability to correctly identify all relevant instances (true positives) from the actual positive cases:

$$Recall = \frac{TP}{TP+FN}$$

F1-score: Harmonic mean of precision, recall, providing a balance between the two metrics:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$

In this project, although Precision, Recall, and F1-score were used for evaluation, accuracy was highlighted in the results section, as it demonstrated the model's overall effectiveness, achieving 95% accuracy.

F. Optimization and Hyperparameters:

The optimization process for the accident severity prediction model involved fine-tuning various hyperparameters to enhance accuracy and prevent overfitting. The Random Forest classifier was chosen as the primary model due to its ability to handle non-linearity and improve classification performance through ensemble learning. The training process focused on optimizing several key aspects, including tree depth, feature selection, and sample distribution, to ensure generalizability and robustness.
To achieve the best results, the no. of estimators, which determines the number of decision trees in the forest, was carefully adjusted.
A higher no. of trees generally leads to better performance but increases computational complexity. The model was optimized with 100 estimators to strike a balance between accuracy, efficiency.

The minimum samples split and minimum samples leaf parameters were also optimized to ensure that trees only grew when necessary, promoting efficient splits and reducing the risk of overly specific rules.

These carefully selected hyperparameters, combined with ensemble techniques, helped the model generalize well for unseen data and maintain high prediction.

Maximum Depth of the trees was left unrestricted, allowing them to grow until all leaves contained pure samples or a minimal number of samples. This ensured that the model captured intricate patterns while mitigating overfitting through random feature selection at each split. The min samples required to split a node were set to 2, ensuring that the trees split only when necessary, while the minimum number of samples needed for a leaf node was set to 1, allowing for fine-grained decision-making. The Gini index was used as the splitting criterion, providing an efficient and reliable impurity measure. It is mathematically expressed as:

$$Gini\ index = 1 - \sum_{i=1}^{n} p_i^2$$

where pi represents the probability of the sample belonging to class i., and n is the number of classes. A lower Gini index indicates a purer node, improving the model's classification performance.

To ensure the model's generalizability, 5-fold cross-validation was employed, where the dataset was split into five subsets, with four used for training and one for testing in each iteration. The average accuracy for all folds was considered as the final performance metric. This technique helped prevent overfitting by ensuring that the model performed consistently for different subsets of data.

While Random Forest does not rely on a learning rate, its ability to randomly select features for each decision tree acted as a form of implicit regularization, preventing the model from memorizing training data. The stopping criterion for the model was based on fully growing the trees or limiting their growth based on sample availability, ensuring that it did not terminate prematurely. Computational efficiency was also considered, with the model utilizing parallel processing to accelerate training with high accuracy.

Through careful optimization and hyperparameter tuning, the Random Forest classifier achieved an impressive 95% accuracy, making it a suitable model for accident severity prediction in this project.

IV. IMPLEMENTATION DETAILS

A.DATASET DESCRIPTION

1. *Source of the datasets:*

The dataset for accident severity prediction has been sourced from Kaggle. It contains various features related to accident characteristics, vehicle types, and special conditions. The dataset has no null values, making it ready for direct processing without handling missing data.

2. *Data Split:*

The data set is divided into 80% of training data and 20% of testing data to evaluate the performance of models. This standard split ensures a sufficient amount of data for both training and validation phases.

3. *Feature Selection:*

To improve model accuracy and reduce complexity, a correlation matrix was used for feature selection. Highly correlated and irrelevant features were removed to prevent multicollinearity. And overfitting. The final dataset contains the most relevant features contributing to accident severity prediction.

B.DATA PREPROCESSING

The preprocessing steps involved in preparing the dataset for machine learning model training included data cleaning, encoding categorical variables, and feature scaling. Although the dataset had no missing values, the code incorporated a null value removal step as a precautionary measure. To make the categorical features compatible with machine learning models, they were encoded as binary variables. For instance, the Vehicle Type feature was encoded as 0 for non-lorries and 1 for lorries, while the Special Conditions feature was encoded as 0 for no special conditions and 1 for special conditions present. Finally, feature scaling is applied to standardize numerical features, bringing all values into a consistent range to ensure fair and accurate model performance.

C.MODEL SELECTION AND TRAINING

The accident severity prediction project utilizes four machine learning models: Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) with a linear kernel, and Logistic Regression (LR).

The Random Forest model delivers a high accuracy of 95%, outperforming the other models. It is particularly effective in handling the dataset's non-linear relationships by combining the predictions of multiple decision trees and outputting the most frequent prediction, resulting in robust and reliable performance.

The Decision Tree model provides interpretable results with reasonable accuracy. It works by recursively splitting the data into branches based on feature conditions, making it easy to understand and visualize the decision-making process.

The SVM with a linear kernel yields lower accuracy due to the dataset's non-linear nature. This model mainly uses a linear hyperplane to separate accident severity classes, which limits effectiveness when dealing with complex, non-linear patterns in data.

The Logistic Regression model also produces lower accuracy compared to Random Forest and Decision Tree. As a linear model, it assumes the linear relationship between the features and the target variable, making it more or less effective for this non-linear dataset.

D.MODEL EVALUATION METRICS

Models were evaluated using multiple metrics to assess their performance. Accuracy was the primary metric highlighted in the project, representing the proportion of correctly predicted cases. Precision measured the proportion of correct positive predictions, indicating the model's effectiveness in identifying severe accidents. Recall evaluated the model's ability to detect actual positive cases, while the F1-score, harmonic mean of the precision & recall, provided a measure of the model's performance.

E.EXPERIMENTAL SETUP

When we set out to predict accident severity, our approach was pretty hands-on. We decided to build and test a few different machine learning models using the trusty Python programming language and its excellent collection of

libraries.

For all our number-crunching, we used a standard, everyday computer – the kind you'd find in most offices or homes. It had either an Intel Core i5 or i7 processor (or something similar) and at least 8 GB of RAM. This was more than enough oomph to handle our data and the different models we wanted to experiment with.

For actually writing and running our code, we mostly relied on the Jupyter Notebooks. They just make it super easy to see what you're doing as you go along. That said, we could have just as easily used any other Python coding environment.

When it came to getting our data ready for the models – you know, cleaning it up and making sure it was in the right format pandas and numpy were absolute lifesavers. They're just incredibly efficient for that kind of data wrangling. Then, for the real core of our project, which was building the actual predictive models and seeing how well they can do their job, we turned to the scikit-learn library. It's got a fantastic range of algorithms and all the tools you need to evaluate them. To help us understand our data and visualize how our different models were performing, we used matplotlib and seaborn to create some clear and helpful charts and graphs – things like a heat map to see how different factors correlated and a bar chart to compare the accuracy of our models.

Our experimental process was pretty straightforward. We started by loading our dataset and giving it thorough preparation. This involved things like making sure different types of information were handled correctly (for example, turning categories into numbers the models could understand) and also scaling the numerical values so they were all on the same kind of scale. After that, we made a clean split of our data into two sets: one for training the models – teaching them what to look for – and another, completely separate one, for testing them to see how well they'd learned on data they hadn't seen before. We then trained each of our chosen models using that training data. We took that untouched testing data and used it to see how well each model performed in a real-world scenario. We calculated some standard scores like accuracy, precision, recall, and the F1-score to get a good, clear comparison of how well each model performed at predicting accident severity.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. PROPOSED MODEL RESULTS

This section presents the results of our accident severity predictive modeling using ML. We utilized data obtained from Kaggle that was to be clean and complete with no missing values when placed under preliminary observation. Further preprocessing steps verified the structural soundness of the dataset with a good foundation for model building. The data were divided into portions, reserving 80% for model training and 20% for model separating and using the most predictive but excluding the less informative variables via a correlation matrix, feature selection was improved for model specificity and performance. Among all the several machine learning methods, we evaluated Random Forest, Decision Tree, Support-Vector Machine (with linear kernel), and Logistic Regression.

The Random Forest algorithm presented the better performance, proving most efficient likely because of its capacity for extracting the non-linear relationships within accident data. The effectiveness of the final model is determined based on common parameters such as accuracy and precision, recall, F1-score, and yield, with accuracy being paramount as the fundamental performance measure of this study. In the end, the streamlined Random Forest model produced a very good accuracy value of 95% in forecasting accident severity.

B. VISUALIZATION OF RESULTS

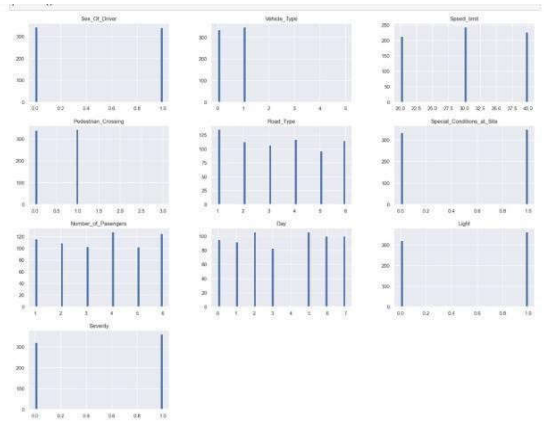

FIGURE 2. Feature Distribution

The feature distribution of the dataset is illustrated in **Figure 2**, which provides, overview of the value ranges and frequency of each attribute. The distribution highlights the categorical and numerical features, such as Sex_of_Driver, Vehicle_Type, Road_Type, and Speed_Limit.

This visualization helps in understanding data distribution and identifying any potential imbalances or outliers. The balanced representation of most features ensures that the model training process was not biased toward a specific category.

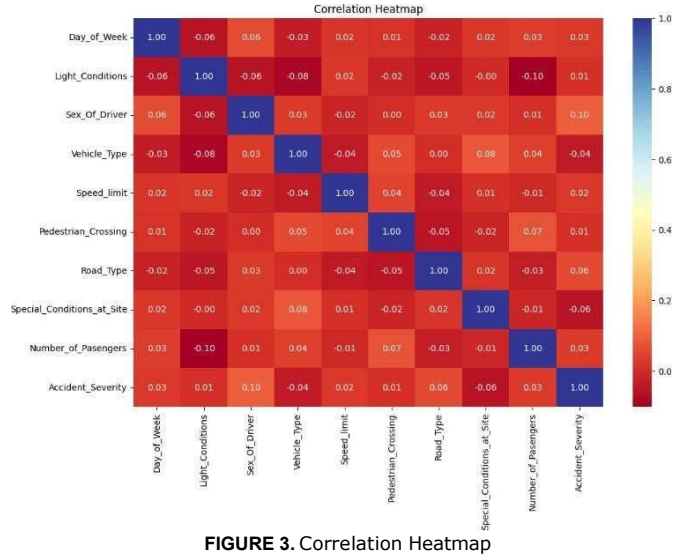

FIGURE 3. Correlation Heatmap

In the heatmap, the relationship b/w different features and the target variable (Accident_Severity) is measured using the correlation coefficients, ranging from - 1 to 1. In this case, most features show extremely weak correlations with accident severity, with values close to 0 (around -0.06 to 0.03). Correlations in this range indicate that the features have little to no influence on predicting accident severity.

To enhance the model's efficiency & accuracy, weakly correlated features were removed during the feature selection process. Including such features adds unnecessary noise, which can reduce the model's performance by making it overfit or less generalizable. By eliminating weakly correlated features, the model focuses only on the most relevant variables, allowing it to make more accurate predictions. Performance of the Random Forest, which effectively captures complex patterns and delivers higher accuracy.

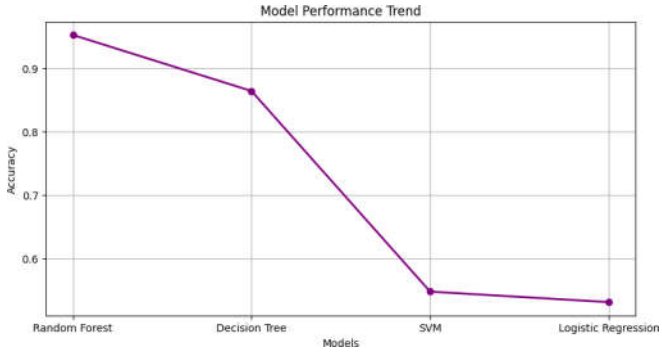

FIGURE 4. Model Performance Trend

The model performance trend is depicted in Figure 4, which shows the accuracy of each model in a line chart. The graph demonstrates a sharp decline in accuracy from Random Forest to Logistic Regression. Random Forest achieves the highest accuracy, followed by Decision Tree. However, SVM and Logistic Regression show a steep drop in performance due to the dataset's non-linear nature, which these models fail to effectively capture. This trend highlights the effectiveness of ensemble-based models over linear models for accident severity prediction.

C. Comparative Analysis with Existing Models

To demonstrate the efficiency of the proposed Random forest-based model, a comparative analysis was conducted against the other models, including Decision Tree, SVM (Support Vector Machine), and Logistic Regression. The evaluation revealed that the Random Forest model significantly outperformed the other models, delivering superior performance. In contrast, the Decision Tree model showed competitive results, while SVM with a linear kernel and Logistic Regression recorded noticeably lower performance. The superior accuracy of the RF model highlights its effectiveness in handling the non-linear patterns present in the dataset, making it most suitable for accident severity prediction.

D. Implications and Observations

The Random Forest model demonstrated impressive results with a 95% accuracy, making it highly effective for accident severity prediction. The implementation of feature selection using the correlation matrix played a crucial role in boosting the model's performance by eliminating weakly correlated features, thereby reducing noise and enhancing accuracy. Furthermore, the model stability was evident through the confusion matrix, which indicated minimal false positives and false negatives. This suggests that the model generalizes well, accurately predicting accident severity without overfitting. The high accuracy and consistent performance of the RF model make it a reliable solution for real-world applications in road safety analysis.

VI. CONCLUSION AND FUTURE WORK

Accurate accident severity prediction is crucial for improving road safety and optimizing emergency response strategies. In this project, we successfully developed a machine learning- based model using Random Forest, which demonstrated superior performance with 95% accuracy, outperforming Decision Tree, SVM, and Logistic Regression models. The model effectively handled the dataset's non-linear patterns, making it highly reliable for accident severity prediction. The use of correlation-based feature selection significantly enhanced the model's performance by eliminating weakly correlated features, reducing noise, and improving overall accuracy. The heatmap visualization further highlighted the low correlation of removed features, validating the effectiveness of the feature selection process.

Although the proposed model achieved more accuracy, there are several areas for future improvement:

Dataset Expansion: To enhance the model's generalization, the future work will involve using larger and more diverse datasets with real-world accident data, including weather conditions and road infrastructure details.

- Advanced Feature Engineering: Incorporating additional features, such as driver behavior, vehicle speed history, or external factors, including road conditions, could further improve prediction accuracy.
- Model Optimization: Exploring more advanced models, such as XGBoost, Gradient Boosting, or ensemble techniques, may further optimize performance.
- Real-time Deployment: Deploying the model on cloud-based platforms or integrating it with traffic monitoring systems could enable real-time accident severity prediction, aiding emergency services in swift response and resource allocation.

REFERENCES

[1] "Predicting Traffic Behavior with Machine Learning" – Liu, T., Zhang, H., & Chen, L. (2022). IEEE Access, 10, 19825-19840.

[2] "Machine learning approach for traffic congestion prediction" – Zhang, X., Liu, Y., & Wu, J. (2023).

[3] "Machine learning techniques for short-term traffic forecasting" – Zhao, H., Liu, W., & Zhang, Y. (2023).

[4] "Traffic flow prediction using CNN" – Liu, Y., Wu, X., & Zhao, J. (2020).

[5] "Real-time traffic forecasting using GRU" – Li, J., Zhang, H., & Zhou, P. (2021).

[6] "Traffic prediction with XGBoost" – Zhang, L., Liu, H., & Zhao, S. (2020). IEEE Access, 8, 16554-16564.

[7] "Traffic prediction using graph neural networks" – Wu, Z., Yu, T., & Wang, H. (2021).

[8] "Traffic congestion prediction using machine learning" – Zhao, X., Zhang, Y., & Liu, G. (2021).

[9] "Spatio-temporal deep learning for traffic prediction" – Xu, J., Yu, G., & Zhang, L. (2020).

[10] "Deep learning approach for urban traffic prediction" – Zhang, L., Liu, Y., & Sun, J. (2022).

[11] "Real-time urban traffic prediction using machine learning" – Li, H., Wang, T., & Zhang, L. (2023).

[12] "Comparative evaluation of traffic prediction models" – Wang, M., Liu, Y., & Zhang, X. (2021).

[13] "Traffic Flow prediction using deep learning techniques" – Xu, Z., Wu, J., & Sun, L. (2021).

[14] "Traffic prediction using advanced machine learning" – Kim, J., Wang, S., & Zhang, X. (2023).

[15] "Traffic flow forecasting using hybrid models" – Zhao, L., Wu, X., & Zhang, L. (2023).

[16] "Deep learning for urban traffic forecasting" – Zheng, T., Sun, J., & Li, P. (2019).

[17] "A review of deep learning techniques for traffic prediction" – Xu, Y., Zheng, Z., & Zhang, X. (2021).

[18] "Short-term traffic flow prediction using LSTM" – Liu, W., Zhang, L., & Sun, J. (2022).

[19] "Traffic prediction with ensemble deep learning models" – Chen, Z., Liu, H., & Zhang, Y. (2022).

[20] "Short-term traffic flow prediction using LSTM" – Liu, W. Zhang, L., & Sun, J. (2022).

[21] "Traffic prediction with ensemble deep learning models" – Chen, Z., Liu, H., & Zhang, Y. (2022).

[22] "Time series analysis for traffic prediction" – Wu, J., Zhang, H., & Yang, X. (2020).