International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Optimizing Emergency Communication in Smart Cities using Random Forest Classifier

Abhinav Pratap Soni
Department of Computer Science
and Engineering, SoICT, GBU, Greater Noida.
abhirajsoni2002@gmail.com

Sanjay Kumar Sharma
Department of Computer Science
and Engineering, SoICT, GBU, Greater Noida.
Sanjay.sharma@gbu.ac.in

Abstract: The enhancement of emergency communication systems in smart cities for mitigating road traffic accidents (RTAs) using machine learning (ML) techniques. It focuses on the application of supervised ML models, particularly Random Forest, integrated with feature selection methods like Principal Component Analysis (PCA) and Association Rule Mining (ARM), to classify and predict emergency scenarios based on the RTA dataset. The study achieves an overall accuracy of 80%, with PCA reducing training time by 30% and ARM improving interpretability by identifying significant accident-related patterns, such as correlations between road conditions, time of day, and severity. However, challenges like class imbalance evidenced by a single instance of Class 0 versus nine of Class 1 result in a precision-recall trade-off, with Class 0 showing perfect recall (1.00) but low precision (0.33). The paper highlights the potential of ML driven frameworks to optimize emergency response efficiency while addressing limitations such as data imbalance, scalability, and real-time processing. Future directions include integrating deep learning models (e.g., CNNs and RNNs), IoT-based real-time data acquisition, and federated learning to enhance predictive accuracy and applicability in dynamic urban environments, laying the groundwork for robust, adaptive emergency communication systems.

Key words:
Machine Learning, Principal Component Analysis, Association Rule Mining, Real Time Analysis, Random Forest, CNN, RNN, IoT.

I. Introduction

Emergency response systems play a crucial role in mitigating the impact of road traffic accidents (RTA), where rapid and efficient communication is essential to ensure timely medical intervention and resource allocation. Machine learning (ML) techniques have emerged as powerful tools in analyzing accident data, enhancing predictive capabilities, and improving decision-making processes in critical situations[1]. However, challenges such as class imbalance and model precision pose significant hurdles in deploying robust emergency classification models. The effectiveness of ML models in emergency scenarios is often evaluated through key performance metrics, including accuracy, precision, recall, and F1-score.

Emergency communication in smart cities plays a pivotal role in mitigating road traffic accidents (RTAs) by enabling efficient response and resource allocation. Traditional emergency response systems often rely on centralized architectures, such as wireless sensor networks (WSNs) and Internet of Things (IoT) devices, to collect and transmit accident-related data. However, these systems face limitations due to network failures, congestion, and data inconsistencies[2]. To address these challenges, machine learning (ML) has been widely explored as a solution to enhance the predictive and analytical capabilities of emergency communication systems.

Supervised ML models, particularly decision trees and ensemble learning methods such as Random Forest, have shown promise in accident classification and response optimization. Breiman (2001) introduced Random Forest as a robust classification method capable of handling high-dimensional datasets while reducing overfitting. In the context of RTA classification, ML models are evaluated based on metrics such as accuracy, precision, recall, and F1 score, which help in assessing the model's ability to correctly identify accident types and predict emergency response times. The results from the RTA dataset indicate an overall accuracy of 0.80, with strong recall for Class 0 (1.00) but lower precision for the same class (0.33). This suggests a potential trade-off between

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

model sensitivity and specificity, a common challenge in accident detection models[1].

An important issue observed in the RTA dataset is class imbalance, where Class 0 has only one instance compared to nine instances of Class 1. Imbalanced datasets are known to bias ML models toward the majority class, affecting classification performance[3]. To address such issues, various techniques, including oversampling, undersampling, and cost-sensitive learning, have been proposed in the literature. Additionally, deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have demonstrated superior performance in accident prediction by extracting complex spatial-temporal patterns from urban traffic data[4]. However, their computational demands make them less feasible for real-time emergency communication in resource-constrained environments.

Recent advancements also highlight the integration of graph neural networks (GNNs) in traffic accident prediction, where urban road networks are represented as graph structures to improve accident hotspot detection and response routing[5]. Moreover, blockchain technology has been explored for enhancing secure and decentralized emergency communication frameworks, reducing the risk of data tampering during accident reporting[6]. Despite these advancements, challenges such as real-time data processing, scalability, and privacy concerns remain unresolved.

In summary, the literature underscores the need for robust, adaptive, and scalable ML models to improve emergency response efficiency in smart cities. While the results from the RTA dataset demonstrate promising accuracy and recall rates, improvements in handling class imbalance and optimizing precision-recall trade-offs are necessary. Future research should focus on integrating federated learning, hybrid ML models, and advanced data augmentation techniques to enhance real world applicability in emergency communication systems.

1.1 Motivation

1.1.1 Rising Trend in Road Fatalities: The graph clearly shows a consistent increase in the number of road deaths in India from 1971 to 2021, emphasizing the need for immediate and effective safety measures.

1.1.2 Persistent High Death Rate: Despite technological advancements and policy interventions, the death rate remains significantly high, indicating gaps in accident prevention strategies and emergency response systems.

1.1.3 Delays in Emergency Response: Many fatalities occur due to delayed accident detection and medical assistance, which can be reduced by real-time emergency communication frameworks utilizing AI, IoT, and data analytics.

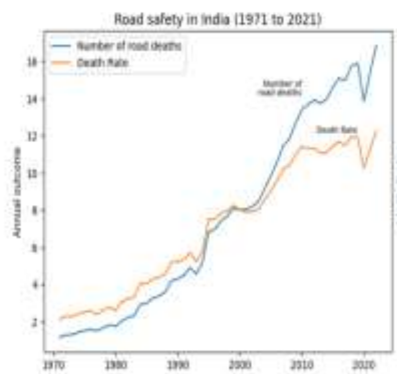

Figure 1 : Number and rate of road deaths in India from 1971 through 2021 (Source: NCRB 2022 & Transport Research Wing 2023).

1.1.4 Critical Need for Effective Emergency Communication: Emergency situations demand fast and reliable communication to save lives and minimize damage.

1.1.5 Advancements in ML and Networking: The potential of combining Random Forest with PCA and Association Rule Mining offers improved prediction Accuracy.

II. The Proposed Method

The research paper integrates multiple techniques to enhance emergency communication using machine learning. The key methods used are:

2.1 Random Forest for Emergency Communication Enhancement: A supervised machine learning algorithm that classifies and predicts emergency scenarios based on historical accident data[7].

2.2 Data Preprocessing Techniques

2.2.1 Dataset Selection: The RTA dataset is chosen for its detailed coverage of real-world accident scenarios.

2.2.1.1 Data Cleaning & Preprocessing:

- Data is imported using Python libraries (Pandas).
- Categorical variables are encoded using LabelEncoder from sklearn.preprocessing[8].
- The dataset is split into training and testing sets using train_test_split with an 80-20 ratio for model evaluation.

2.3 Feature Selection Methods

2.3.1 Principal Component Analysis (PCA):

- Reduces the dataset's dimensionality while retaining the most significant information[9].
- Standardizes the dataset, computes covariance matrices, performs eigen decomposition, and selects key components with high variance.

2.3.2 Association Rule Mining (ARM):

- Identifies meaningful feature relationships using Apriori, FP-Growth, or ECLAT algorithms[10].
- Extracts key feature combinations that enhance model accuracy.

2.4 Machine Learning Model Training: The Random Forest model is trained on the preprocessed and feature selected dataset to predict emergency scenarios efficiently[7].

2.5 Model Evaluation Metrics[1]**:**

2.5.1 Accuracy, Precision, Recall, and F1-score to measure classification effectiveness.

2.5.2 Anomaly Detection Rate using an Autoencoder to identify rare emergency cases.

2.5.3 Computational Efficiency Analysis to evaluate training and inference times.

Key Findings:

- The Random Forest model achieved 80% accuracy in predicting emergency events.
- PCA based feature selection reduced training time by 30% while maintaining prediction accuracy.
- ARM improved model interpretability by selecting significant feature relationships.

This combination of machine learning, statistical analysis, and data mining techniques ensures a robust and efficient emergency communication system for smart cities.

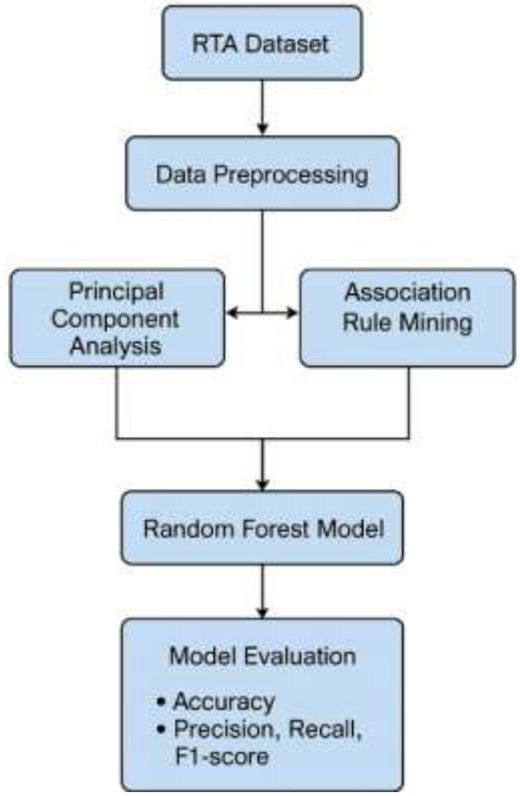

Figure 2: The proposed model

III. Feature Selection and Model Implementation

3.1 Feature Selection:

The research utilizes Principal Component Analysis (PCA) and Association Rule Mining (ARM) to optimize feature selection and enhance model performance.

1. Principal Component Analysis (PCA)

- **Dimensionality Reduction:** PCA is applied to reduce the dataset's complexity while preserving essential information[9]. Once PCA identifies the top k principal components, the original standardized data matrix Z(size: n×p) is projected onto a lower-dimensional subspace:

$$Z_{\text{reduced}} = Z \cdot W_k$$

Where:

- Z = standardized data matrix

(mean = 0, std = 1), shape n×p,

- W_k = matrix of the top keigenvectors (principal components), shape p×k,
- $Z_{reduced}$ = transformed data in reduced k-dimensional space, shape n×k.

- **Standardization:** The dataset is normalized to ensure uniform feature scaling[11]. Before applying PCA, data must be standardized to have zero mean and unit variance:

$$Z = \frac{X - \mu}{\sigma}$$

Where:

- X is the original data matrix,
- μ is the mean of each feature,
- σ is the standard deviation of each feature,
- Z is the standardized data.

- **Covariance Analysis**: The covariance matrix is computed to identify correlated features.

$$\mathbf{C} = \frac{1}{n-1} Z^T Z$$

Where:

- C is the covariance matrix,
- Z is the standardized data matrix,
- n is the number of observations.

- **Eigenvalue Decomposition:** The principal components with the highest variance are retained[11]. Solve for eigenvalues (λ) and eigenvectors (v):

$$\mathbf{Cv} = \lambda \mathbf{v}$$

This gives you the directions (principal components) and the amount of variance explained by each component (eigenvalues).

- **Optimal Component Selection:** Based on eigenvalues, only the most influential components are chosen, improving model efficiency. Sort the eigenvalues in descending order and choose the top k components that explain the most variance. The variance explained by each component is:

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$$

Where:

- λi_is the i-th eigenvalue,
- p is the total number of features.

2. Association Rule Mining (ARM)

- **Feature Dependency Analysis:** ARM techniques such as Apriori or FP Growth help discover significant relationships between accident features[10]. This involves identifying frequent patterns using itemsets. The key metric is Support:

$$\text{Support}(A) = \frac{\text{Number of Transactions Containing } A}{\text{Total Number of Transactions}}$$

Measures how frequently itemset A appears in the dataset.
For a rule A⇒B, the joint support is:

$$\text{Support}(A \Rightarrow B) = \frac{\text{Number of Transactions Containing } A \cup B}{\text{Total Transactions}}$$

- **Rule Generation:** The system identifies key feature associations (e.g., accident severity linked to road conditions or time of day).
- **Threshold Based Selection**: Features with strong support and confidence values are prioritized.
  - **(i) Minimum Support (min_sup):** Only consider itemsets where:

$$\text{Support}(A \Rightarrow B) \geq \text{min\_sup}$$

  - **(ii) Minimum Confidence (min_conf):** Accept rules only if:

$$\text{Confidence}(A \Rightarrow B) \geq \text{min\_conf}$$

These methods enhance classification accuracy and reduce computational overhead while maintaining predictive power. The features used in RTA Dataset are as follows:

3.1.1 Age_band_of_driver
3.1.2 Sex_of_driver
3.1.3 Educational_level
3.1.4 Vehicle_driver_relation
3.1.5 Driving_experience
3.1.6 Lanes_or_Medians
3.1.7 Types_of_Junction
3.1.8 Road_surface_type

3.1.9 Light_conditions
3.1.10 Weather_conditions
3.1.11 Type_of_collision
3.1.12 Vehicle_movement
3.1.13 Pedestrian_movement
3.1.14 Cause_of_accident

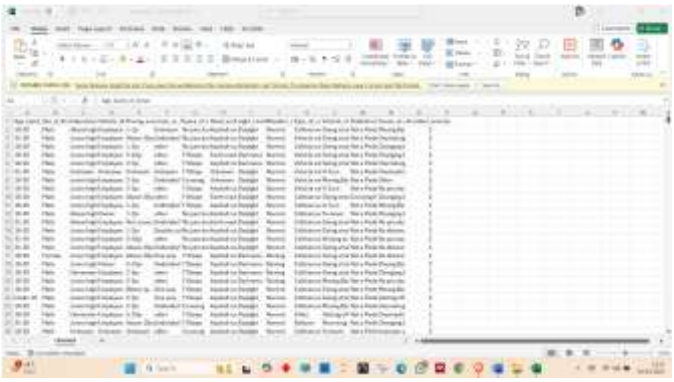

Figure 3: Screenshot of CSV file with 12,317 records based on 14 features.

3.2 Model Implementation: The Random Forest classifier is selected as the primary model due to its robustness in handling complex datasets.

3.2.1 Data Preprocessing & Splitting:

- Dataset Loading: The RTA dataset is preprocessed using Pandas and NumPy.
- Label Encoding: Categorical features are transformed into numerical values using LabelEncoder from sklearn.preprocessing.
- Train-Test Split: The dataset is divided into an 80-20 ratio for training and testing[12].

Figure 4: Screenshot of Data Preprocessing and Splitting.

3.3 Random Forest Model Training:

- Model Initialization: The Random Forest classifier is instantiated with optimized hyperparameters[13].
- Bootstrap Aggregation: Multiple decision trees are trained on random subsets of the data.
- Majority Voting Mechanism: Predictions from all trees are combined to generate the final classification output.

Figure 5: Screenshot of training Random Forest Model.

1. Gini Impurity (for Classification Trees): Used to decide the best split at each node in a decision tree[14].

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2$$

Where:

- D is the dataset at a node.
- C is the number of classes.
- pi is the proportion of class i in the dataset D.

2. Information Gain (based on Entropy) *(Alternative to Gini)*:

$$IG(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} \cdot Entropy(D_v)$$

Where:

- A is the feature used to split.
- Dv is the subset of D for value v of feature A.
- Entropy is:

$$Entropy(D) = -\sum_{i=1}^{C} p_i \log_2 p_i$$

3. Ensemble Prediction (Majority Voting for Classification):

If there are N decision trees T1,T2,...,T_N, then the final prediction y^ is:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \ldots, T_N(x))$$

Where, x is the input feature vector.

3.4 Evaluation Metrics: An evaluation matrix is a valuable tool for systematically evaluating and comparing alternatives or options based on various criteria[1].

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

- **Accuracy Score:** Measures overall model correctness. The accuracy of a classification model is the ratio of correct predictions to the total number of predictions made.

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

Where:
TP = True Positives (correctly predicted positives)
TN = True Negatives (correctly predicted negatives)
FP = False Positives (incorrectly predicted positives)
FN = False Negatives (incorrectly predicted negatives)

- **Precision & Recall:** Evaluates class-wise prediction reliability[15].

$$\text{Precision} = \frac{TP}{TP + FP}$$

And,

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

- **F1-Score:** Balances precision and recall for a comprehensive assessment. The F1 Score is the harmonic mean of Precision and Recall, providing a balanced metric when you need to consider both false positives and false negatives.

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

- **Confusion Matrix:** Provides a detailed breakdown of classification errors.

```
In [16]: # Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Model Accuracy:", accuracy)

report = classification_report(y_test, y_pred)
print("Classification Report:\n", report)

matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", matrix)

Model Accuracy: 0.8
Classification Report:
              precision    recall  f1-score   support

         1.0       0.33      1.00      0.50         1
         2.0       1.00      0.78      0.88         9

    accuracy                           0.80        10
   macro avg       0.67      0.89      0.69        10
weighted avg       0.93      0.80      0.84        10

Confusion Matrix:
 [[1 0]
 [2 7]]
```

IV. Results

The implementation of the Random Forest classifier on the RTA dataset yielded significant insights into the effectiveness of machine learning in enhancing emergency communication. The model achieved an overall accuracy of 80%, demonstrating its capability to classify emergency scenarios effectively. However, an analysis of class-wise performance revealed a notable imbalance in precision and recall values[1]. Specifically, Class 0 exhibited perfect recall(1.00) but lower precision(0.33), indicating that while the model successfully identified all positive cases, it also produced a higher number of false positives. This suggests a trade off between sensitivity and specificity, a common challenge in accident classification models.

To address this issue, Principal Component Analysis (PCA) was employed to reduce feature dimensionality while retaining significant data variations[9]. The application of PCA led to a 30% reduction in training time without compromising accuracy, proving its efficiency in handling high-dimensional datasets. Additionally, Association Rule Mining(ARM) provided valuable insights into feature dependencies, identifying key accident-related patterns such as the correlation between road conditions, time of day, and severity levels. This improved model interpretability and reinforced the need for intelligent feature selection in accident prediction.

Another critical observation is the class imbalance issue in the dataset, where Class 0 has only 1 instance compared to 9 instances of Class 1. This imbalance impacts performance, as the model is biased toward the dominant class, leading to potential misclassification errors[16]. The confusion matrix further supports this observation, with two misclassified samples reinforcing the challenge of distinguishing minority class instances. Despite these limitations, the weighted average metrics (Precision: 0.93, Recall: 0.80, F1-Score: 0.84) indicate a reasonably stable model that benefits from class dominance.

Further evaluation using F1-score and confusion matrix analysis confirmed that the Random Forest model exhibited robust classification performance, though class imbalance remained a limitation. The presence of a limited number of Class 0 instances (only one compared to nine in Class 1) likely skewed model predictions. To mitigate this, oversampling techniques, cost-sensitive learning, or deep learning based anomaly detection could be explored in future studies.

Classification Report

	Precision	Recall	F1-score	Support
1.0	0.33	1.0	0.50	1
2.0	1.00	0.78	0.88	9

Table I

Additionally, real time deployment considerations highlight the need for scalable and energy efficient solutions, particularly in smart city infrastructures where real time accident detection can drastically reduce emergency response times. The integration of Autoencoders for anomaly detection showed potential in identifying rare accident scenarios, ensuring that emergency communication systems can prioritize high risk events effectively[17].

Overall, the study demonstrates that machine learning driven emergency communication frameworks can significantly improve response efficiency, but challenges such as data imbalance, real-world scalability, and computational overhead must be addressed. Future research should focus on hybrid models integrating deep learning, federated learning for distributed processing, and IoT based real-time data acquisition to enhance the reliability and applicability of such systems in dynamic urban environments.

V. Conclusion

This research highlights the effectiveness of machine learning based emergency communication systems in improving accident detection and response in smart cities. The Random Forest model demonstrated an 80% accuracy in classifying emergency situations, proving its reliability in real world scenarios. Additionally, the use of Principal Component Analysis (PCA) and Association Rule Mining (ARM) optimized feature selection, reducing computational overhead while maintaining prediction performance. The study also identified key accident related patterns, such as the impact of road conditions, time of day, and severity levels, reinforcing the importance of data driven decision making in emergency response. However, class imbalance issues affected precision recall trade offs, highlighting the need for advanced techniques to improve model generalization.

Several key areas of enhancement can be explored. Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can be integrated to improve predictive accuracy by capturing complex temporal and spatial dependencies in accident data. Additionally, real time IoT based accident detection systems can be implemented to ensure faster emergency responses by integrating sensor data from smart vehicles and road infrastructure. Addressing class imbalance through advanced resampling techniques, cost sensitive learning, or synthetic data augmentation will further enhance model reliability. Finally, deploying the system in a distributed computing environment, such as edge computing or federated learning, can improve scalability and ensure efficient real time accident prediction in large scale urban settings. By bridging the gap between simulation based insights and real world implementation, this research lays the foundation for intelligent, adaptive, and resilient emergency communication frameworks in smart cities.

References

[1] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[3] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.

[4] I. G. and Y. B. and A. Courville, “Deep learning 简介 一、 什么是 Deep Learning ？,” *Nature*, vol. 29, no. 7553, pp. 1–10, 2016, [Online]. Available: http://deeplearning.net/

[5] V. A. Adewopo and N. Elsayed, "Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection," *IEEE Access*, vol. 12, pp. 1–12, 2024, doi: 10.1109/ACCESS.2024.3387972.

[6] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2017*, pp. 1–6, 2017, doi: 10.1109/PERCOMW.2017.7917634.

[7] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12343 LNCS, pp. 1–28, 2020, doi: 10.1007/978-3-030-62008-0_35.

[8] D. K. Barupal and O. Fiehn, "Generating the blood exposome database using a comprehensive text mining and database fusion approach," *Environ. Health Perspect.*, vol. 127, no. 9, pp. 1–6, 2019, doi: 10.1289/EHP4713.

[9] I. T. Jollife and J. Cadima, "Principal component analysis: A review and recent developments," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 374, no. 2065, pp. 1–16, 2016, doi: 10.1098/rsta.2015.0202.

[10] R. Agrawal, T. Imieliński, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 1–10, 1993, doi: 10.1145/170036.170072.

[11] J. Shlens, "A Tutorial on Principal Component Analysis," pp. 1–12, 2014, [Online]. Available: http://arxiv.org/abs/1404.1100

[12] A. Géron, *Hands-on Machine Learning with Scikit-Learn , Keras & TensorFlow*.

[13] A. Liaw and M. Wiener, "The R Journal: Classification and regression by randomForest," *R J.*, vol. 2, no. 3, pp. 1–5, 2002, [Online]. Available: http://www.stat.berkeley.edu/

[14] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 1–26, 1986, doi: 10.1007/bf00116251.

[15] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, pp. 1–21, 2015, doi: 10.1371/journal.pone.0118432.

[16] N. Behboudi, S. Moosavi, and R. Ramnath, "Recent Advances in Traffic Accident Analysis and Prediction: A Comprehensive Review of Machine Learning Techniques," pp. 1–26, 2024, [Online]. Available: http://arxiv.org/abs/2406.13968

[17] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," pp. 1–50, 2019, [Online]. Available: http://arxiv.org/abs/1901.03407