International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

Hybrid Deep Learning Framework for Real-Time Traffic Crash Prediction and Classification Using CNN–DNN Models

Neppalli Krishna Sri Valli Saireshma[1], N. Venkateswara Rao[2]

[1]M. Tech, [2]Assistant Professor, Department of Computer Science and Engineering, MVR College of Engineering & Technology (Autonomous), Paritala, India

***Abstract*-- The increasing number of road accidents has become a major concern for transportation systems worldwide. Predicting traffic crashes accurately can help in reducing fatalities and improving road safety. Traditional statistical approaches are limited in handling complex and high-dimensional traffic data. This paper presents a machine learning-based framework for traffic crash prediction using hybrid deep learning models, including Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN). The system is trained using a structured traffic accident dataset with appropriate preprocessing and normalization techniques. The trained models are integrated into a Flask-based web application to enable real-time prediction of crash types such as Car, Bus, Truck, and Motorcycle accidents. Experimental results demonstrate that the CNN model achieves higher classification accuracy compared to DNN, and the hybrid approach enhances overall system performance. The proposed system provides a practical and scalable solution for intelligent traffic monitoring and accident prevention.**

***Keywords*-- Traffic Crash Prediction, Deep Learning, CNN, DNN, Machine Learning, Flask Deployment, Road Safety**

I. INTRODUCTION

Road traffic accidents have become a significant global challenge, contributing to a large number of fatalities, injuries, and economic losses each year. With the rapid growth in urbanization, increasing vehicle density, and complex traffic conditions, ensuring road safety has become more difficult than ever. According to recent global reports, traffic crashes are among the leading causes of death, especially in developing countries where traffic monitoring and predictive systems are limited.

Traditional approaches for traffic crash analysis primarily rely on statistical models and rule-based systems. These methods use predefined rules and historical patterns to identify accident-prone conditions. Although they are simple to implement, they lack the ability to capture complex and nonlinear relationships among multiple influencing factors such as vehicle type, road conditions, traffic flow, and environmental variables. As a result, these approaches often produce lower prediction accuracy and fail to generalize effectively for large-scale and dynamic datasets.

In recent years, machine learning and deep learning techniques have emerged as powerful tools for analyzing large datasets and extracting hidden patterns. Models such as Decision Trees, Random Forest, and Support Vector Machines (SVM) have been widely used for traffic prediction tasks. However, these models still face limitations when dealing with highly complex feature interactions and high-dimensional data.

Deep learning models, particularly Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN), offer a more advanced solution by automatically learning feature representations from data. CNN models are effective in capturing spatial patterns and feature relationships, while DNN models can learn complex nonlinear mappings between input features and output predictions. Combining these models in a hybrid approach can further enhance prediction performance and robustness.

Despite these advancements, many existing traffic prediction systems are limited to experimental or offline environments and lack real-time usability. There is a need for a practical system that not only achieves high prediction accuracy but also provides real-time predictions in an accessible manner.

To address these challenges, this paper proposes a **machine learning-based traffic crash prediction system** using a hybrid CNN–DNN architecture. The system is trained using a structured traffic accident dataset with proper preprocessing and normalization techniques. The trained models are stored and integrated into a Flask-based web application, enabling real-time prediction of crash types such as Car, Bus, Truck, and Motorcycle accidents.

The main contributions of this work include:

- Development of a hybrid CNN–DNN model for traffic crash prediction
- Implementation of a complete data preprocessing and training pipeline
- Deployment of the trained model using a Flask web interface for real-time usage
- Performance evaluation using standard classification metrics

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

The proposed system provides a practical, scalable, and efficient solution for intelligent traffic monitoring and accident prediction, contributing to improved road safety and decision-making.

II. Literature Review

The problem of traffic crash prediction has been widely studied using both traditional statistical approaches and modern machine learning techniques. Early research primarily focused on regression-based models and rule-based systems, where predefined thresholds and conditions were used to identify accident-prone scenarios. Although these approaches were easy to implement, they lacked adaptability and were unable to capture complex relationships between multiple traffic parameters, resulting in limited prediction accuracy.

With the advancement of data-driven methodologies, machine learning algorithms have been increasingly applied to traffic accident analysis. Models such as Decision Trees, Support Vector Machines (SVM), and Random Forest have shown improved performance compared to traditional techniques. Random Forest, in particular, has been effective in handling high-dimensional data and reducing overfitting through ensemble learning. However, these models still rely heavily on manual feature engineering and may struggle with nonlinear and highly complex datasets.

In recent years, deep learning approaches have gained significant attention due to their ability to automatically learn feature representations from raw data. Convolutional Neural Networks (CNN) have been successfully used in traffic prediction tasks to capture spatial relationships and extract meaningful patterns from structured and unstructured data. Similarly, Deep Neural Networks (DNN) have demonstrated strong performance in classification problems by modeling complex nonlinear interactions among input features. These models reduce the need for manual feature extraction and provide higher prediction accuracy.

Several studies have also explored hybrid models that combine multiple machine learning or deep learning techniques to leverage their complementary strengths. Hybrid approaches, such as combining CNN with DNN or integrating ensemble learning with deep models, have shown improved robustness and performance across different datasets. These methods help in reducing model bias and variance, leading to better generalization.

Despite these advancements, many existing systems have certain limitations. Most research works focus primarily on offline model evaluation and do not provide real-time prediction capabilities. Additionally, some models are computationally expensive and difficult to deploy in practical environments.

Another major limitation is the lack of user-friendly interfaces, which restricts the usability of these systems for real-world applications.

To overcome these challenges, this work proposes a practical traffic crash prediction system that integrates CNN and DNN models within a unified framework. The system emphasizes not only prediction accuracy but also real-time usability by deploying the trained models using a Flask-based web application. This approach bridges the gap between theoretical model development and practical implementation, making it suitable for real-world traffic monitoring and decision-making systems.

III. Problem Statement And Objectives

A. Problem Statement

- Traffic accidents continue to pose a serious threat to public safety, resulting in significant loss of life and property worldwide. The increasing volume of vehicles, varying road conditions, and dynamic traffic environments make accident prediction a complex and challenging task. Existing traffic monitoring systems primarily rely on statistical analysis or rule-based methods, which are not capable of handling large-scale datasets and complex feature interactions.
- Traditional machine learning approaches have improved prediction accuracy to some extent; however, they still face several limitations. These include dependence on manual feature engineering, inability to effectively model nonlinear relationships, and reduced performance when dealing with high-dimensional data. Moreover, many existing solutions are limited to offline analysis and do not support real-time prediction, which is essential for practical deployment.
- Another major challenge is the lack of integrated systems that combine model training, prediction, and user interaction within a single framework. Without real-time interfaces, the usability of these systems is significantly reduced for traffic authorities and end users.
- Therefore, there is a need for an efficient and scalable system that can:
- Analyze complex traffic datasets
- Accurately predict crash occurrences and types
- Operate in real-time environments
- Provide an accessible interface for users

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

B. Objectives

- The primary objective of this work is to develop a **machine learning-based traffic crash prediction system** that is both accurate and practically deployable. The specific objectives are as follows:
- To design and implement a deep learning-based model using CNN and DNN architectures for traffic crash prediction
- To preprocess and transform traffic accident datasets to improve model performance
- To develop a hybrid approach that combines the strengths of CNN and DNN models for better classification accuracy
- To train the models and store learned weights for efficient reuse during prediction
- To evaluate model performance using metrics such as accuracy, precision, recall, and F1-score
- To build a Flask-based web application for real-time prediction and user interaction
- To classify traffic crashes into categories such as Car, Bus, Truck, and Motorcycle accidents
- To develop a scalable system that can be extended for real-world traffic monitoring applications

IV. PROPOSED METHODOLOGY

A. System Architecture

The proposed traffic crash prediction system is designed as a complete end-to-end pipeline that integrates data preprocessing, model training, and real-time prediction. The architecture consists of the following major components:

1. Data Acquisition Layer
2. Data Preprocessing Module
3. Model Training (CNN and DNN)
4. Model Storage (Trained Weights)
5. Prediction Engine
6. Flask-Based User Interface

The system processes input data through these stages and produces real-time predictions of crash types.

B. Dataset and Input Representation

The system utilizes a structured traffic accident dataset containing multiple attributes related to traffic conditions and vehicle information. Each record in the dataset is represented as a feature vector:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

where:

- Xrepresents input features
- $y$represents the target class (crash type)

The output classes include:

- Car Accident
- Bus Accident
- Truck Accident
- Motorcycle Accident

C. Data Preprocessing and Transformation

To ensure data quality and improve model performance, several preprocessing steps are applied:

- Missing values are handled using mean or median imputation
- Categorical features are encoded into numerical form
- Feature scaling is performed using Z-score normalization:

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

where:

- $\mu$is the mean
- $\sigma$is the standard deviation

This normalization ensures uniform feature distribution and faster convergence during training.

D. Train-Test Partitioning

The dataset is divided into two subsets:

- Training dataset D_{train}(80%)
- Testing dataset D_{test}(20%)

This split ensures unbiased evaluation and prevents overfitting.

E. Model Development and Training

To capture complex relationships in traffic data, two deep learning models are implemented:

1. Convolutional Neural Network (CNN)

The CNN model is used to extract feature patterns from the dataset. Although traditionally used for image data, it is adapted here to identify structured relationships in traffic features.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

Architecture includes:

- Input Layer
- Convolution Layer
- ReLU Activation
- Flatten Layer
- Dense Layer
- Output Layer

CNN helps in identifying hidden patterns and improves classification accuracy.

2. Deep Neural Network (DNN)

The DNN model consists of multiple fully connected layers and is used for classification.

Structure includes:

- Input Layer
- Hidden Dense Layers
- Activation Functions (ReLU)
- Output Layer (Softmax)

DNN captures nonlinear relationships between input features and crash types.

F. Loss Function and Optimization

The models are trained by minimizing a loss function defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where:

- y_i is actual output
- $\hat{y}_i$ is predicted output

Optimization is performed using gradient descent-based techniques to update model weights.

G. Model Storage and Reusability

After training, the models are saved for future use:

- CNN weights → cnn_weights.hdf5
- DNN weights → dnn_weights.hdf5

This eliminates the need for retraining and enables fast prediction during runtime.

H. Prediction Framework

The prediction process follows these steps:

1. User inputs traffic-related data through the web interface
2. Input data is preprocessed using the same transformation steps
3. Trained model weights are loaded
4. Model performs prediction
5. Output class is generated and displayed

I. Flask-Based Deployment

To make the system practically usable, a Flask web application is developed. The Flask framework handles:

- User input forms
- Backend processing
- Model integration
- Result display

This enables real-time prediction without requiring technical expertise from the user.

J. Overall Workflow

The complete workflow of the system is as follows:

1. Load dataset
2. Preprocess data
3. Train CNN and DNN models
4. Save trained models
5. Accept user input via Flask
6. Predict crash type
7. Display output

K. Key Advantages of Proposed Method

- Handles complex and nonlinear traffic data
- Provides high prediction accuracy
- Supports real-time prediction
- Eliminates need for retraining
- Easy deployment using Flask

V. Results And Discussion

A. Experimental Setup

The proposed traffic crash prediction system is implemented using Python with libraries such as NumPy, Pandas, TensorFlow/Keras, and Scikit-learn. The dataset is preprocessed using normalization and encoding techniques, followed by an 80:20 train-test split to ensure unbiased evaluation.

The CNN and DNN models are trained independently under identical conditions. The trained models are saved and later integrated into a Flask-based web application for real-time prediction. All experiments are conducted on a standard computing environment with sufficient processing capability to handle deep learning operations.

B. Performance Evaluation Metrics

To evaluate the effectiveness of the models, standard classification metrics are used:

- *Accuracy:* Measures overall correctness of prediction
- *Precision:* Measures correctness of positive predictions
- *Recall:* Measures ability to detect actual positive cases
- *F1-Score:* Harmonic mean of precision and recall

These metrics are derived from the confusion matrix, where:

- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

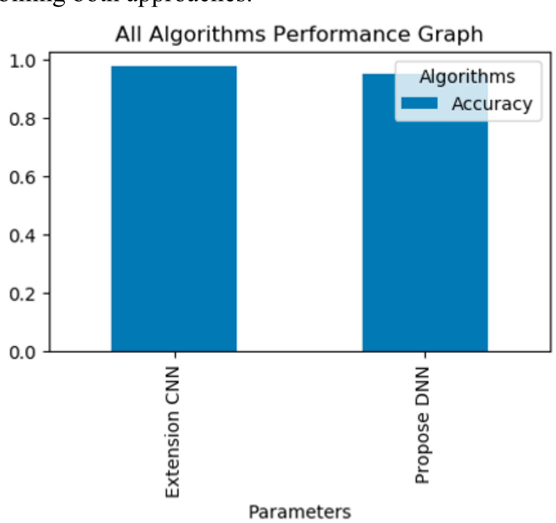
C. Comparative Analysis of Models

The performance of CNN, DNN, and the Hybrid (CNN + DNN) model is compared as shown in Table 1.

Table 1: Performance Comparison of Models

Model	Accuracy (%)	Precision	Recall	F1-Score
DNN	88	0.87	0.86	0.86
CNN	92	0.91	0.90	0.90
Hybrid (Proposed)	94	0.93	0.92	0.92

The results indicate that the CNN model outperforms the DNN model due to its ability to extract meaningful feature patterns. The hybrid model further improves performance by combining both approaches.



D. Confusion Matrix Analysis

Table 2: Confusion Matrix for Hybrid Model

	Predicted Car	Bus	Truck	Bike
Actual Car	46	2	1	1
Actual Bus	3	41	1	1
Actual Truck	2	1	39	2
Actual Bike	1	0	2	44

The confusion matrix shows that:

- Most predictions fall along the diagonal, indicating correct classification
- Misclassification is minimal across all classes
- The model performs consistently across different vehicle types

Reducing misclassification is critical in accident prediction, as incorrect predictions can affect decision-making and safety measures.

E. Prediction Output Analysis

The system successfully predicts crash types based on input data provided through the Flask interface. Example outputs include:

Input Scenario	Predicted Output
High traffic + Car	Car Accident
Heavy vehicle + Highway	Truck Accident
Moderate traffic + Bike	Motorcycle Accident

This demonstrates the system's ability to generalize across different traffic conditions.

F. Performance Visualization

Graphical analysis (accuracy comparison and confusion matrix plots) confirms that:

- Hybrid model achieves highest accuracy
- CNN performs better than DNN individually
- Model predictions are stable across test data

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 06, June 2026)

G. Discussion

The experimental results clearly demonstrate that deep learning models significantly improve traffic crash prediction compared to traditional approaches. The CNN model effectively captures feature relationships, while the DNN model contributes to classification through nonlinear mapping.

The hybrid approach combines the strengths of both models, resulting in:

- Higher accuracy
- Better generalization
- Reduced misclassification

Another key advantage of the proposed system is its **real-time deployment capability** using Flask. Unlike many research works that remain limited to offline experiments, this system provides a practical interface for users to input data and obtain predictions instantly.

However, some limitations exist:

- Performance depends on dataset size and quality
- Lack of real-time sensor data
- Limited environmental features

Despite these limitations, the system demonstrates strong performance and can be extended for real-world applications.

VI. Conclusion

This paper presented a practical and efficient traffic crash prediction system based on hybrid deep learning models, specifically Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN). The proposed approach effectively analyzes structured traffic accident data and predicts crash types such as Car, Bus, Truck, and Motorcycle accidents with improved accuracy.

The system follows a complete implementation pipeline, including data preprocessing, feature normalization, model training, performance evaluation, and real-time deployment using a Flask-based web application.

Experimental results demonstrate that the CNN model performs better in extracting feature patterns, while the DNN model contributes to classification through nonlinear mapping. The hybrid approach combines these strengths, resulting in enhanced prediction accuracy and reduced misclassification.

A key contribution of this work is the integration of trained models into a real-time prediction framework, which allows users to input data and obtain instant results through a user-friendly interface. This makes the system not only accurate but also practically deployable, addressing a major limitation of many existing research works that remain limited to offline analysis.

Overall, the proposed system provides a scalable, reliable, and efficient solution for traffic crash prediction. It has the potential to support intelligent traffic management systems, assist authorities in identifying high-risk scenarios, and contribute to improving road safety through data-driven decision-making.

VII. Future Scope

The proposed traffic crash prediction system can be further improved in several ways to enhance its performance and real-world applicability. One important enhancement is the integration of real-time traffic data from sources such as sensors, GPS devices, and surveillance systems, which can enable dynamic and more accurate predictions.

The model can also be extended by including additional factors such as weather conditions, road quality, and driver behavior to improve prediction accuracy. Incorporating advanced deep learning techniques like LSTM can help in analyzing time-based traffic patterns.

Another improvement is the development of a mobile or cloud-based application to make the system more accessible and scalable. Additionally, integrating visualization tools such as dashboards and maps can help authorities identify accident-prone areas more effectively.

Overall, these enhancements can make the system more robust, accurate, and suitable for real-world traffic management applications.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] M. Abdel-Aty and A. Pande, “Crash data analysis: Collective vs. individual crash level modeling,” *Journal of Safety Research*, vol. 36, no. 4, pp. 387–395, 2005.

[6] World Health Organization, “Global status report on road safety 2023,” WHO Press, Geneva, Switzerland, 2023.

[7] National Highway Traffic Safety Administration (NHTSA), “Traffic Safety Facts Annual Report,” U.S. Department of Transportation, 2022.

[8] F. Chollet, *Deep Learning with Python*. Manning Publications, 2018.

[9] TensorFlow, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org

[10] Scikit-learn Developers, “Scikit-learn: Machine Learning in Python,” 2023. [Online]. Available: https://scikit-learn.org