IJRDET

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 5, May 2026)

NeuroTraffic: A Hybrid Deep Learning Framework for Context-Aware Road Crash Intelligence and Predictive Risk Modeling

Juluru V L S Krishna Priya, P. Uday Kiran Kumari

M.Tech Scholar, Department of CSE, Nimra College of Engineering and Technology, Jupudi, Ibrahimpatnam, Vijayawada, India

Assistant Professor, Department of CSE, Nimra College of Engineering and Technology, Jupudi, Ibrahimpatnam, Vijayawada, India

Abstract- Road accidents continue to be a critical challenge in modern transportation systems, causing significant human and economic losses. Accurate prediction of crash occurrences can play a vital role in improving road safety and reducing fatalities. Conventional statistical models are often inadequate for handling complex, high-dimensional traffic data.

This study introduces a hybrid deep learning framework combining Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) to predict traffic crash types effectively. The system utilizes a structured accident dataset, applying preprocessing techniques such as normalization and encoding to enhance data quality.

The trained models are deployed through a Flask-based web interface, enabling real-time classification of accidents into categories such as Car, Bus, Truck, and Motorcycle. Experimental evaluation demonstrates that CNN outperforms DNN individually, while the hybrid approach achieves superior overall performance.

The proposed framework offers a scalable and practical solution for intelligent traffic monitoring and predictive safety systems.

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

I. Introduction

Road transportation plays a vital role in economic development and daily human activities; however, it is also associated with a significant number of accidents, leading to loss of life, injuries, and financial damage.

With the continuous growth in urbanization and the rapid increase in the number of vehicles, traffic conditions have become more complex, making accident prediction a challenging task. Developing intelligent systems capable of predicting road crashes in advance has become essential for improving road safety and reducing risks.

Traditional traffic accident analysis methods mainly rely on statistical models and rule-based approaches. These methods use historical data and predefined conditions to identify accident-prone scenarios. Although simple and easy to implement, such techniques are limited in their ability to capture complex, nonlinear relationships among multiple influencing factors such as traffic density, vehicle type, road conditions, and environmental variations. As a result, their prediction accuracy is often insufficient for real-world applications.

To overcome these limitations, machine learning techniques have been introduced for traffic crash prediction. Models such as Decision Trees, Random Forest, and Support Vector Machines (SVM) have demonstrated improved performance by learning patterns from data.

However, these approaches still depend heavily on manual feature engineering and often struggle with high-dimensional and highly dynamic datasets.

In recent years, deep learning has emerged as a powerful solution for handling complex data structures. Deep learning models can automatically extract meaningful features from raw data and model intricate relationships without extensive manual intervention. Convolutional Neural Networks (CNN) are particularly effective in identifying structured patterns and feature dependencies, while Deep Neural Networks (DNN) are capable of learning nonlinear mappings between input variables and output predictions. By combining these models, it is possible to achieve better performance and improved prediction accuracy.

Despite these advancements, many existing traffic prediction systems remain confined to offline analysis and lack real-time implementation capabilities. Additionally, the absence of user-friendly interfaces limits their usability in practical scenarios. Therefore, there is a strong need for a system that not only provides accurate predictions but also supports real-time interaction and deployment.

To address these challenges, this work proposes a hybrid deep learning-based traffic crash prediction system that integrates CNN and DNN models within a unified framework. The system utilizes a structured traffic dataset, applies appropriate preprocessing techniques, and trains both models to learn complex traffic patterns. The trained models are then deployed using a Flask-based web application, enabling users to input traffic conditions and obtain real-time predictions of crash types such as Car, Bus, Truck, and Motorcycle accidents.

The major contributions of this research are as follows:

- Design and implementation of a hybrid CNN–DNN model for traffic crash prediction
- Development of a complete preprocessing and training pipeline
- Deployment of the system using a Flask-based web interface for real-time prediction
- Performance evaluation using standard classification metrics
- Creation of a scalable and practical solution for intelligent traffic monitoring

II. Literature Review

Traffic crash prediction has been an active area of research for many years, with approaches evolving from traditional statistical techniques to advanced data-driven models. Early studies primarily relied on regression-based models and rule-driven systems to analyze accident data. These methods used predefined thresholds and historical patterns to identify risk factors. Although such approaches were simple and interpretable, they lacked flexibility and were not capable of modeling complex interactions among multiple variables, resulting in limited predictive performance.

With the advancement of computational techniques, machine learning algorithms were introduced to improve prediction accuracy. Methods such as Decision Trees, Support Vector Machines (SVM), and Random Forest gained popularity due to their ability to learn patterns from data. Among these, Random Forest proved effective in handling high-dimensional datasets and reducing overfitting through ensemble learning. However, these models still depend significantly on manual feature selection and engineering, which can limit their effectiveness when dealing with highly complex and nonlinear traffic data.

In recent years, deep learning approaches have shown promising results in traffic crash prediction. Deep learning models are capable of automatically extracting hierarchical feature representations from raw data, reducing the need for manual preprocessing. Convolutional Neural Networks (CNN) have been widely applied to capture spatial relationships and identify hidden patterns in structured datasets. Similarly, Deep Neural Networks (DNN) have demonstrated strong performance in classification tasks by modeling complex nonlinear relationships between input features and output variables.

Several researchers have explored hybrid approaches that combine multiple models to leverage their individual strengths. For example, integrating CNN with DNN enables the system to benefit from both feature extraction capabilities and nonlinear mapping. Hybrid models have shown improved robustness, higher accuracy, and better generalization compared to single-model approaches. These methods also help in reducing bias and variance, leading to more reliable predictions across diverse datasets.

Despite the progress made in this domain, certain challenges still remain. Many existing studies focus primarily on model development and offline evaluation, without considering real-time implementation. Additionally, some deep learning models require high computational resources, making them difficult to deploy in practical environments.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 5, May 2026)

Another limitation is the lack of user-friendly interfaces, which restricts accessibility for non-technical users and real-world stakeholders.

To address these limitations, the present work proposes a hybrid deep learning framework that integrates CNN and DNN models for traffic crash prediction. Unlike conventional approaches, the system is designed not only for high accuracy but also for real-time usability. By deploying the trained models using a Flask-based web application, the proposed system provides an accessible and practical solution for traffic monitoring and decision-making.

III. Problem Statement And Objectives

A. Problem Statement

Road traffic accidents remain one of the most critical challenges in modern transportation systems, leading to significant loss of human lives, severe injuries, and economic damage. The rapid increase in vehicle population, diverse road conditions, and dynamic traffic environments have made accident prediction a complex and multidimensional problem.

Existing traffic analysis systems largely depend on traditional statistical techniques and rule-based mechanisms. While these approaches are simple and interpretable, they are not capable of effectively handling large-scale datasets or capturing complex relationships among multiple influencing factors such as vehicle characteristics, traffic density, road infrastructure, and environmental conditions. As a result, their predictive capability is often limited and insufficient for real-world deployment.

Although machine learning models have improved prediction accuracy to some extent, they still face notable challenges. Many of these models rely heavily on manual feature engineering, which requires domain expertise and may not capture all relevant patterns present in the data. Furthermore, conventional models struggle to represent nonlinear relationships and interactions within high-dimensional datasets, leading to reduced performance in complex scenarios.

Another major limitation of existing solutions is the lack of real-time prediction capability. Most research works are confined to offline analysis, where models are trained and evaluated using historical data without providing live predictions. This significantly limits their practical usability in real-world traffic monitoring and decision-making systems.

Additionally, there is a lack of integrated frameworks that combine data preprocessing, model training, prediction, and user interaction within a single system. The absence of user-friendly interfaces further restricts accessibility for traffic authorities and general users.

Therefore, there is a strong need for an intelligent, scalable, and real-time system that can:

- Efficiently process complex and high-dimensional traffic data
- Accurately predict crash occurrences and classify accident types
- Capture nonlinear relationships among multiple influencing factors
- Provide real-time predictions through an interactive interface
- Be easily deployable for practical applications

B. Objectives

The primary goal of this research is to develop an efficient and practical traffic crash prediction system using advanced deep learning techniques. The specific objectives are as follows:

- To design and implement a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) for improved prediction accuracy
- To perform effective data preprocessing, including handling missing values, encoding categorical features, and normalization
- To develop a robust training framework that captures complex patterns in traffic data
- To evaluate the performance of the proposed model using standard classification metrics such as accuracy, precision, recall, and F1-score
- To store trained model parameters for efficient reuse and faster prediction
- To develop a Flask-based web application that enables real-time prediction and user interaction

- To classify traffic crashes into multiple categories such as Car, Bus, Truck, and Motorcycle accidents
- To create a scalable system that can be extended for real-world intelligent traffic management 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:

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

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.

Architecture includes:

- Input Layer
- Convolution Layer
- ReLU Activation

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 5, May 2026)

- 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)

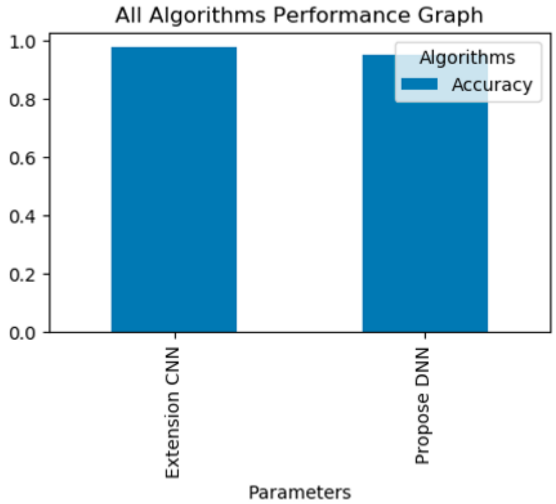
C. Comparative Analysis of Models

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

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 5, May 2026)

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

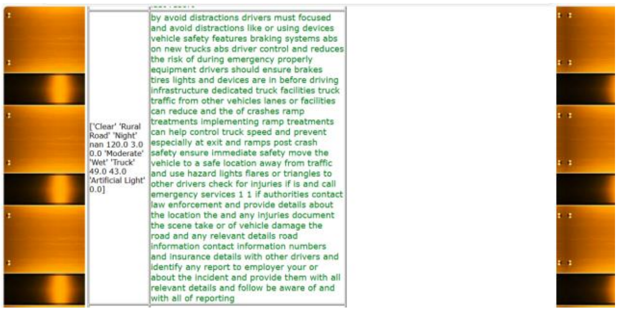
This demonstrates the system's ability to generalize across different traffic conditions.

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

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 research presents a comprehensive and practical approach for traffic crash prediction using a hybrid deep learning framework that integrates Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN). The proposed system is designed to effectively analyze structured traffic data and accurately classify accident types such as Car, Bus, Truck, and Motorcycle, addressing the limitations of traditional statistical and standalone machine learning methods.

The implementation follows a complete pipeline, including data preprocessing, normalization, model training, performance evaluation, and real-time deployment. The CNN model contributes by extracting meaningful feature patterns from the input data, while the DNN model enhances prediction capability by learning complex nonlinear relationships. The combination of these models in a hybrid architecture significantly improves overall prediction accuracy and reduces misclassification.

Experimental results demonstrate that the hybrid model outperforms individual models in terms of accuracy, precision, recall, and F1-score. Additionally, the integration of the trained model into a Flask-based web application enables real-time prediction, making the system practical and accessible for users without technical expertise. This real-time capability bridges the gap between theoretical research and real-world implementation, which is often lacking in existing studies.

Overall, the proposed framework provides a scalable, efficient, and reliable solution for intelligent traffic crash prediction. It has strong potential to support traffic management authorities, enhance decision-making processes, and contribute to improving road safety through data-driven insights.

VII. Future Scope

Although the proposed hybrid deep learning framework demonstrates strong performance in traffic crash prediction, there are several opportunities for further enhancement to improve its effectiveness and real-world applicability.

One significant direction for future work is the integration of real-time data sources such as IoT sensors, GPS devices, surveillance cameras, and smart traffic systems. Incorporating live traffic streams would enable dynamic prediction and provide more accurate and timely insights for accident prevention. Additionally, including external factors such as weather conditions, road quality, lighting conditions, and driver behavior can further enrich the dataset and improve prediction accuracy.

Another promising extension is the adoption of advanced deep learning architectures such as Long Short-Term Memory (LSTM) networks and hybrid CNN-LSTM models. These models are particularly effective in capturing temporal dependencies and sequential patterns in traffic data, which can enhance the system's ability to predict accidents over time.

The system can also be expanded into a cloud-based or mobile application to improve scalability and accessibility. Deploying the model on cloud platforms would allow handling large-scale data and multiple users simultaneously, while a mobile interface could provide real-time alerts and recommendations to drivers and authorities.

Furthermore, the integration of data visualization tools such as dashboards, heatmaps, and geographic mapping systems can help in identifying accident-prone zones and analyzing traffic patterns more effectively. This would assist policymakers and traffic management authorities in making informed decisions.

Future improvements may also include the use of explainable artificial intelligence (XAI) techniques to enhance model transparency and interpretability. This would help users understand the factors influencing predictions, thereby increasing trust and usability of the system.

Overall, these enhancements can transform the proposed framework into a more intelligent, adaptive, and comprehensive traffic management solution capable of addressing real-world challenges more effectively.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 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*, Geneva, Switzerland, 2023.

[7] National Highway Traffic Safety Administration, *Traffic Safety Facts Annual Report*, U.S. Department of Transportation, 2022.

[8] F. Chollet, *Deep Learning with Python*. Shelter Island, NY, USA: 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

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 5, May 2026)

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.