Editor's Comments:

[Improving Manuscript Novelty and Quality](https://chatgpt.com/c/686a6466-860c-800e-8893-b13bd3f942e9)2-

1[-Development of a Hybrid Deep Learning Model for Car Crash Prediction Using Driver’s Behavioral Pattern](https://link.springer.com/chapter/10.1007/978-3-031-66410-6_2?utm_source=chatgpt.com)

Here are several **recent, high-quality ML/DL studies (2023–2025)** that would strengthen your literature review and better position your contribution:

### **🧠 CrashFormer: A Multimodal Architecture to Predict Crash Risk (2024)**

A multi-modal Transformer-based model using accident history, weather, map imagery, and demographic data. Outperforms traditional LSTM/CNN models by ~1.8% in F1 across 10 major U.S. cities [arxiv.org+1mdpi.com+1](https://arxiv.org/abs/2402.05151?utm_source=chatgpt.com).

### **⚡ inTformer: Intersection Crash Prediction via Transformer (2023)**

Introduces a time-embedded attention-based Transformer for real-time crash likelihood at intersections using connected vehicle data. Achieves around 70–73% sensitivity [aimspress.com+2arxiv.org+2mdpi.com+2](https://arxiv.org/abs/2307.03854?utm_source=chatgpt.com).

### **🌐 STZITDGNN: Uncertainty‑Aware Graph Neural Network (2023)**

A novel probabilistic GNN that integrates Tweedie distributions to model crash frequency and severity with uncertainty quantification on London road segments [arxiv.org](https://arxiv.org/abs/2309.05072?utm_source=chatgpt.com).

### **🔄 Feature Group Tabular Transformer (FGTT) (Dec 2024)**

Employs a Transformer on grouped tabular data (weather, pavement, geometry) to predict crash types and extract causal insights—outperforms XGBoost, CatBoost, RF [arxiv.org+1arxiv.org+1](https://arxiv.org/abs/2412.06825?utm_source=chatgpt.com).

### **📊 Predictive Crash Analytics Using Deep Learning (Feb 2025)**

This framework merges ensemble learning and multi-modal data, achieving 0.893 mAP (+15% vs. baselines) and 92.4% accuracy in hotspot detection from ~60k high-quality samples [arxiv.org](https://arxiv.org/abs/2502.05777?utm_source=chatgpt.com).

### **🚦 RoadInTCP: DL-based Crash Prediction at Intersections (Jan 2025)**

A 3-phase model combining GCN for spatial features and 1D-CNN/RNN for temporal sequences to predict intersection crashes using big heterogenous data—outperforms existing methods [mdpi.com](https://www.mdpi.com/2076-3417/15/2/752?utm_source=chatgpt.com).

### **🤝 Development of Hybrid LSTM–SAE Model (Oct 2024)**

A hybrid SAE‑LSTM architecture trained on driver behavior data from Kaggle; LSTM achieves 98.6% accuracy in crash prediction [link.springer.com+1arxiv.org+1](https://link.springer.com/chapter/10.1007/978-3-031-66410-6_2?utm_source=chatgpt.com).

### **🧩 Two‑Layer Ensemble ML for Crash Prediction (Mar 2024)**

Stacking ensemble (k-NN, AdaBoost, Naive Bayes, Decision Trees + logistic regression meta) with SMOTE and PSO feature selection; achieves 88% accuracy and 83% F1 [mdpi.com](https://www.mdpi.com/2571-5577/7/2/25?utm_source=chatgpt.com).

### **📸 Dashcam + XGBoost for Urban Crash Severity (Jan 2024)**

Uses in-vehicle dashcam data with XGBoost, SVM, and logit models, along with SHAP/Boruta; finds that safety infrastructure reduces severity [aimspress.com](https://aimspress.com/article/id/6597ded0ba35de4e80d47e03?utm_source=chatgpt.com).

### **🌧️ DNN vs RF for Seattle/UK Crash Severity (Nov 2024)**

Compares Random Forest and DNN (91.1% accuracy), identifying environmental factors (snow, water, dusk lighting increases risk +10–13%) [ewadirect.com](https://www.ewadirect.com/proceedings/ace/article/view/17649?utm_source=chatgpt.com).

### **🧬 Meta‑heuristic optimized RF with GIS (2024)**

Uses a Crown Porcupine-optimized RF and GIS to model crash severity, achieving >90% across metrics and revealing weather/spatial factors [cell.com](https://www.cell.com/heliyon/fulltext/S2405-8440%2824%2911626-X?utm_source=chatgpt.com).

## **🔹 How to Use These in Your Paper**

* **State-of-the-art overview**:  
   Transition your lit review from "classic ML only" to encompass **multi-modal DL**, **transformers**, **GNNs**, **and interpretability (e.g., SHAP)** techniques.
* **Identify gaps**:  
  + Are dataset types and modalities comparable?
  + Is uncertainty modeling (e.g., GNN Tweedie) used?
  + Do any leverage your specific country contexts (Canada or Pakistan)?
* **Position your contribution**:  
   Clearly define whether you're adding:  
  + Spatial–temporal modeling (GCN, Transformers),
  + Interpretability (rule mining + SHAP),
  + Cross-country comparative insights,
  + Or real-world deployment components.

Would you like me to draft how these citations could fit into your **literature review narrative**, or help reword the **introduction** to highlight where your work advances beyond these studies?

## **🔥 Top Recent ML/DL Models for Crash Prediction (2023–2025)**

### **1. XGBoost (eXtreme Gradient Boosting)**

* ✅ **Still one of the strongest ML baselines** for structured data.
* ✅ Works well for feature importance analysis, missing data, and class imbalance.
* 🔍 Can be enhanced with SHAP for interpretability.
* ✅ Used in multiple 2023–2024 papers for crash severity & hotspot prediction.

### **2. LightGBM with Feature Engineering + SHAP**

* 📈 Faster and more memory-efficient than XGBoost.
* ⚙️ Combine with SMOTE or ADASYN to handle class imbalance.
* 📊 Use **SHAP plots** to interpret crash factors (weather, behavior, location).

## **🔬 Advanced & Novel DL Architectures**

### **3. Transformer-based Models**

* 🧠 Models: CrashFormer, inTformer, TabTransformer, Informer
* 🛰️ Application: Spatiotemporal crash prediction, especially with time-series, map, or sensor data.
* ✅ Excellent for modeling **attention across time and location**.
* ⚙️ Option: Use TabTransformer for structured datasets with categorical and numerical variables.

### **4. Graph Neural Networks (GNN)**

* 📍 Best for road network and spatial crash modeling.
* 🔁 Models: GCN (Graph Convolutional Network), GAT (Graph Attention Network), ST-GCN, T-GCN
* ✨ Recent addition: STZITDGNN (2023) — adds uncertainty modeling.
* 🔄 Links road topology + traffic flow + crash data = **true smart city modeling**.

### **5. CNN-LSTM or 1D-CNN + RNN Hybrid**

* 🧪 Used in RoadInTCP (2025) and other hybrid frameworks.
* 🧠 CNN extracts features from crash trends or spatial images.
* ⏳ LSTM captures long-term dependencies (like rush hour patterns, seasonal variations).

## **💡 New & Emerging Approaches**

### **6. AutoML with Neural Architecture Search (NAS)**

* 🛠️ Use AutoML libraries (e.g., AutoGluon, AutoKeras, H2O AutoML).
* ⚙️ Quickly test optimized pipelines — useful for government or low-resource agencies.

### **7. Self-Supervised Learning (SSL)**

* 📡 Trains on unlabeled traffic videos or time series and fine-tunes with labeled crash data.
* 🧠 Great for **small labeled datasets** like Rawalpindi RTA or camera feeds.

### **8. Multi-Modal DL**

* 🔄 Integrates images (road scenes), structured data (weather, vehicle info), and sequences.
* 🖼️ Combine CNN for image, Transformer/RNN for time-series, and MLP for tabular inputs.

## **🏁 Best Choice for Your Study (Pakistan + Canada Crash Data)**

| **Model Type** | **Why Use It** | **Tools** |
| --- | --- | --- |
| XGBoost + SHAP | Strong baseline, interpretable, and tunable | xgboost, shap |
| TabTransformer | Works great on tabular traffic datasets | pytorch-tabnet, TabTransformer |
| ST-GCN or T-GCN | For spatial + temporal features like intersection, time, etc. | dgl, pyg, tensorflow\_gnn |
| CNN-LSTM Hybrid | Good for sequence data with road images or crash sequences | keras, torch |
| AutoML (AutoGluon) | Automatically finds optimal model and features | autogluon.tabular |

Great question! Benchmark experiments and ablation studies are **critical** to demonstrate the **robustness, novelty, and utility** of your ML/DL model — especially when you're being challenged on novelty and depth, like in your TRR review.

## **✅ 1. Benchmark Experiments (to validate model performance)**

These test how well your model performs compared to established methods:

### **📌 A. Compare Against Traditional Models**

Use the **same dataset** and **evaluation metrics** (accuracy, F1-score, ROC-AUC) across models like:

| **Model** | **Purpose** |
| --- | --- |
| Logistic Regression | Baseline for binary crash severity classification |
| Decision Tree | Simple, interpretable model |
| Random Forest | Strong ensemble for structured data |
| SVM | Useful for linearly/non-linearly separable features |
| Naive Bayes | Fast, useful when features are categorical |

### **📌 B. Compare with Advanced Models**

| **Model** | **Why** |
| --- | --- |
| XGBoost | Strong ML benchmark |
| LightGBM | Faster gradient boosting |
| TabNet / TabTransformer | DL models for tabular data |
| CNN-LSTM | For time-series or sequence modeling |
| GCN/ST-GCN | If spatial or network topology is included (e.g., road segments) |

### **📊 Metrics to Report**

* Accuracy, Precision, Recall, F1-score
* ROC-AUC
* Confusion Matrix
* PR-AUC (if data is imbalanced)

**Bonus**: Do a 5-fold or 10-fold cross-validation and show the **variance in performance**.

## **🔍 2. Ablation Studies (to test model components)**

These help answer: **"What part of my model or pipeline is most critical?"**

### **🔬 A. Feature Ablation**

Remove one group of features at a time:

| **Feature Group** | **Examples** |
| --- | --- |
| Driver behavior | speeding, inattentiveness |
| Environmental | time of day, weather |
| Road type | highway, urban, rural |
| Location | spatial info, traffic light presence |

📉 Show performance drop when each group is removed.

### **🔬 B. Preprocessing Ablation**

Test your model **with and without** specific preprocessing:

| **Step** | **Test** |
| --- | --- |
| Feature scaling | Try without standardization or normalization |
| SMOTE / class balancing | Try with and without SMOTE |
| Encoding | Compare Label vs. One-Hot vs. Target Encoding |

📊 Include a table showing how each step improves/hurts accuracy or F1.

### **🔬 C. Model Architecture Ablation (for DL)**

If you're using CNN, LSTM, Transformer, etc.:

| **Architecture Component** | **Test** |
| --- | --- |
| LSTM layers | Try 1-layer vs. 2-layer |
| Attention module | With and without self-attention |
| Dropout/Bias layer | With and without dropout |
| Feature fusion | With/without concatenation of additional input (e.g., road images) |

## **🧪 3. Generalization Experiment**

Test your trained model on:

* **Pakistan data → test on Canada**
* **Canada data → test on Pakistan**

✅ This shows whether your model generalizes across regions/countries.

## **🗂️ Suggested Table Format for Paper**

**Benchmark Comparison Table:**

| **Model** | **Accuracy** | **F1** | **ROC-AUC** | **Notes** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.61 | 0.58 | 0.65 | Baseline |
| XGBoost | 0.75 | 0.72 | 0.81 | Proposed |
| TabTransformer | 0.78 | 0.75 | 0.85 | Best |

**Ablation Study Table:**

| **Component Removed** | **Accuracy** | **F1** | **Δ from Full Model** |
| --- | --- | --- | --- |
| Driver Behavior Features | 0.69 | 0.65 | -6% |
| SMOTE Balancing | 0.71 | 0.66 | -4% |
| Attention Layer | 0.72 | 0.68 | -3% |

## **💡 Final Tips**

* Visualize results using **SHAP plots**, **Feature Importance**, or **Heatmaps** of confusion matrix.
* If using DL: plot **training curves**, **loss vs. epochs**, and **attention maps** (if Transformer).
* Add a “What-if” section: How will performance change if data is limited?

Would you like:

* A Python/Jupyter template for ablation studies?
* Sample code to generate comparative tables?
* Visuals like SHAP summary plots or attention maps?

I can also help you integrate these into your paper’s **Results** or **Discussion** section.