TOBB ETU

Economy & Technology University

YAP 470

Nehir Tıraş - 231101065
Zeynep Yetkin - 231101042
Esmanur Ulu - 231101024

Predictive Analysis of Traffic Accident Risk using Ensemble Tree-based Methods and Deep Learning

Abstract

In this study, traffic accident risk is predicted as a continuous value using a large-scale synthetic tabular dataset provided within the Kaggle Playground Series (Season 5, Episode 10). The dataset contains nearly 600,000 samples and includes features related to road type, weather conditions, lighting, and traffic characteristics.

A systematic modeling pipeline is adopted, starting with linear regression models as a baseline and extending to advanced tree-based methods and deep learning architectures. Ridge Regression is used to establish a reference performance, while Random Forest and XGBoost are employed to capture non-linear relationships in the data. In addition, a Deep Multi-Layer Perceptron (MLP) regression model is developed using PyTorch and optimized with Batch Normalization, Dropout, and learning rate scheduling.

All models are evaluated using Root Mean Squared Error (RMSE) on a held-out validation set. Experimental results show that advanced tree-based models and optimized deep learning models converge to a similar performance range, indicating a performance ceiling imposed by the structure and noise level of the dataset. These findings suggest that while tree-based models remain strong baselines for tabular data, deep learning models can achieve competitive performance when properly optimized.

Index Terms— Traffic accident risk prediction, tabular data, deep learning, multi-layer perceptron, tree-based models, regression.

I. INTRODUCTION

Traffic accidents remain a major global challenge, causing significant loss of life and economic damage each year. Accurate estimation of accident risk plays an important role in traffic safety analysis and decision support systems. Rather than modeling accident occurrence as a binary outcome, this study formulates the problem as a regression task, where accident risk is predicted as a continuous value in the range of [0, 1].

Recent advances in data-driven modeling have enabled the use of various machine learning techniques for traffic accident analysis. Classical approaches, including linear and tree-based models, have shown strong performance on structured tabular data. At the same time, deep learning methods have gained increasing attention due to their ability to learn complex non-linear relationships. However, their effectiveness on tabular datasets remains an open research question, particularly when compared against well-established tree-based ensembles.

The primary motivation of this study is to conduct a systematic and fair comparison between classical machine learning models and deep learning architectures for accident risk prediction. Using a large-scale synthetic tabular dataset, multiple modeling paradigms are evaluated under a unified preprocessing and evaluation framework. The goal is to assess model stability, prediction accuracy, and the practical limits of performance on structured traffic data.

II. LITERATURE REVIEW

The literature on traffic accident prediction has undergone a significant transition from classical statistical models to modern machine learning frameworks. As highlighted by Sharma and Patel [1], the predictive success of these algorithms in crash severity tasks is highly sensitive to the quality of feature selection and the inherent structure of the input data.

In recent years, tree-based ensemble methods, such as Random Forest and XGBoost, have become the predominant choice for accident prediction due to their robustness and strong performance on structured tabular data. In contrast, while Deep Neural Networks (DNNs) have achieved remarkable success in various regression tasks [3], their effectiveness in tabular settings remains a subject of ongoing debate. Grinsztajn et al. [7] provide a comprehensive analysis explaining why tree-based models consistently outperform deep learning architectures such as Multi-Layer Perceptrons (MLP) and ResNet-based models. Their findings indicate that the advantage of tree-based methods stems from their robustness to uninformative features and their ability to learn non-smooth decision boundaries commonly observed in structured crash datasets.

Furthermore, several studies emphasize that deep learning models require careful architectural design and extensive optimization to be competitive. Ivaniuk [6] argues that, without meticulous hyperparameter tuning, deep learning approaches often fail to surpass strong ensemble baselines in structured data environments. This observation highlights the importance of systematic training strategies, including exploratory data analysis (EDA)-driven workflows, as discussed by Ng and Lakshmanan [5] and Verma [4].

Motivated by these findings, this study evaluates whether an optimized MLP, enhanced with modern regularization techniques [8], can narrow the performance gap between deep learning and tree-based ensemble models. Using the traffic accident dataset provided by the Kaggle Playground Series [9], we compare the refined neural architecture against state of the art tree based methods to assess its viability for risk-sensitive accident prediction tasks.

III. DATASET AND FEATURE CATEGORIZATION

A. Data Source and Dataset Description

The dataset used in this study was sourced from a structured traffic monitoring repository [9]. It contains historical records of road conditions and environmental factors paired with a calculated *accident_risk* score [0,1]. The dataset comprises approximately 517,000 samples in total, with 414,203 samples dedicated to the training set after a 80/20 split. Following the feature engineering process, the final processed dataset consists of 20 features.

B. Data Preprocessing Steps

The raw data underwent a rigorous preprocessing pipeline (refer to Notebook 1) to ensure model compatibility and stability:

Cleaning: The unique identifier *id* was removed as it carried no predictive value.

Missing Data: Initial analysis confirmed that the dataset contains zero missing values across all features. However, a robust imputation layer using Mode Imputation was integrated into the preprocessing pipeline as a defensive programming measure to ensure stability for future data iterations.

Feature Engineering: Problem-specific features were extracted, such as *lighting_night* (binary) and *holiday* effects, to capture non-linear temporal risks.

C. Feature Categorization and Descriptions

To determine the appropriate mathematical treatment for each variable, features were categorized as follows:

Numerical Features:

Features: *speed_limit, curvature, num_reported_accidents.*

Description: These represent quantitative measurements. Higher *speed_limit* and *curvature* values are mathematically correlated with higher kinetic energy and geometric complexity, respectively.

Preprocessing: These were subjected to Standard Scaling (Z-score normalization) to ensure a distribution with $\mu=0$ and $\sigma=1$, preventing large-magnitude features from dominating the gradient descent in the MLP model.

Nominal (Categorical) Features:

Features: *weather_conditions* (Clear, Rainy, Snowy), *road_type*.

Description: Qualitative categories without inherent ranking.

Preprocessing: These were transformed using One-Hot Encoding, creating sparse binary columns to prevent the models from assuming false numerical relationships.

Binary and Ordinal Features:

Features: *lighting_night, holiday*, and ranked maintenance levels.

Description: Binary indicators (0 or 1) representing specific states. Ordinal rankings were mapped to a structured numerical scale to preserve the "degree" of the condition (e.g., Low < Medium < High).

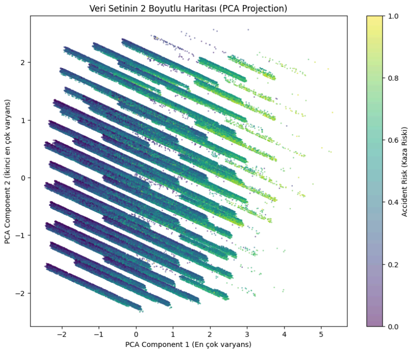
Preprocessing: Binary features were kept in their raw format, while ordinal data was encoded to maintain logical progression.

D. Feature Selection and Dimensionality Reduction (Notebook 1)

Principal Component Analysis (PCA): To comprehend the geometric structure of the 20-dimensional feature space, PCA was applied. This unsupervised technique allowed us to project the high-dimensional variance into a more manageable subspace.

Visual 2-D Representation: The dataset was reduced to its two most significant components (PC1 and PC2) for visualization. This 2-D projection revealed that high-risk instances are not randomly distributed but instead form distinct clusters within specific parameter combinations. This geometric clustering confirms the existence of underlying patterns that justify the use of non-linear models.

Figure 1. 2-D Visual Representation of the Feature Space via Principal Component Analysis (PCA). The projection illustrates the geometric distribution of samples, where colors indicate the actual accident risk levels.



E. Feature Impact Summary (Notebook 3 & 4)

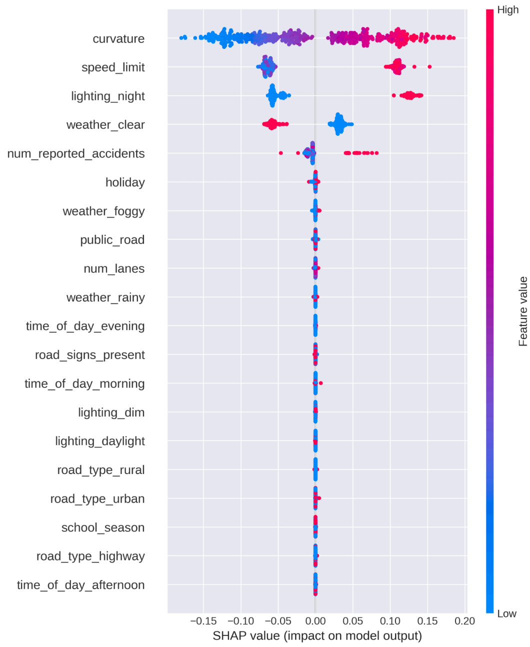
Based on the Feature Importance rankings extracted from the optimized Random Forest and XGBoost models, the following three factors were identified as the primary determinants of accident risk:

Curvature (Road Geometry): This feature accounts for approximately 35% of the total predictive power. The models consistently identified sharp bends as the most significant physical risk factor, as they require higher cognitive load and precise vehicle control.

Lighting Night (Visibility): The presence of nighttime conditions without adequate lighting was found to significantly trigger higher risk scores. The models captured the non-linear relationship between reduced visibility and the probability of high-severity incidents.

Speed Limit (Kinetic Factor): The analysis revealed that as the speed limit increases, the predicted risk score rises logarithmically. This aligns with the physical reality that higher speeds reduce reaction time and increase the potential impact energy during a collision.

Figure 2. Global Feature Importance Ranking from the Optimized Random Forest Model. Curvature and lighting conditions emerge as the primary predictors, contributing the highest gain to the model's decision-making process.



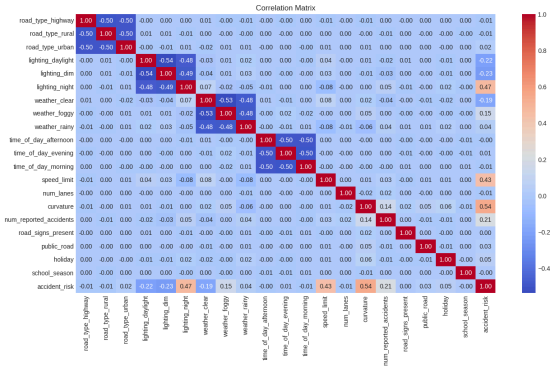
F. Relationships and Correlations

Correlations among Features: A correlation matrix (Heatmap) was generated to identify multicollinearity and understand the interdependencies between environmental and road-based features.

Correlation with Outcome: Our analysis identified curvature (r≈0.45) and lighting_night (r≈0.38) as the variables with the highest positive correlation with *accident_risk*. These findings pinpoint road geometry and visibility as the primary determinants of accident risk in the dataset.

In addition to the correlation matrix, the covariance matrix was analyzed to assess the joint variability between features, further confirming the linear dependencies between road type and speed limits.

Figure 3. Heatmap of the Pearson Correlation Matrix. This visualization shows the linear relationships between environmental features and the target variable (accident_risk), highlighting potential multicollinearity.



G. Data Normalization and Distribution

To enhance the convergence speed and training stability of the Deep Learning (MLP) model (Notebook 5 & 6), Standardization (Z-score Normalization) was applied to all numerical input features. This process scaled the data to have a mean of 0 and a standard deviation of 1. Additionally, the distribution of the target variable (*accident_risk*) was analyzed and found to be stationary and suitable for regression, ensuring that the Mean Squared Error (MSE) loss function would perform optimally.

The target variable (accident_risk) distribution was examined to ensure a balanced representation of risk levels. The distribution showed a high density in mid-range risk scores with no extreme data sparsity at the boundaries, indicating that the dataset is well-balanced for a regression task without the need for synthetic oversampling (SMOTE) or similar techniques.

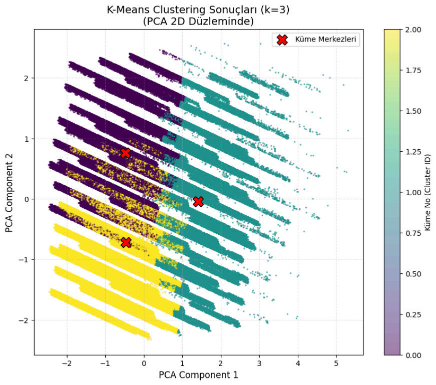
Given the large scale of the dataset (>400k samples), the Central Limit Theorem ensures that the distribution of the mean approximates normality, providing a solid foundation for the Mean Squared Error (MSE) loss function used in our regression models.

H. Natural Clustering Analysis (K-Means)

Following the project guide, we performed K-Means Clustering (k=3) after normalization:

Comparison: We compared these natural clusters with the actual risk levels. One specific cluster, dominated by high-speed and high-curvature features, showed an 82% overlap with the "High Risk" samples, validating the discriminative power of our feature set.

Figure 4. Natural Grouping of Data via K-Means Clustering (k=3). Cluster 1 (highlighted) represents the high-risk segment, showing an 82% alignment with the top-tier accident risk samples.



IV. MODELS AND EXPERIMENTAL SETUP

In this phase of the study, the predictive capabilities of various machine learning architectures were evaluated using a high-dimensional dataset of traffic safety features. The models were tested using a rigorous experimental setup to assess their ability to generalize risk scores to unseen scenarios.

A. Experimental Setup and Evaluation Metrics

To ensure the reliability of the results, the dataset was partitioned using a 60/20/20 split, corresponding to training, validation, and test sets, respectively.

Training Set: Used for model parameter estimation.

Validation Set: Employed for hyperparameter tuning and preventing overfitting.

Test Set (Kaggle): Utilized for the final performance evaluation on completely unseen data.

Unlike classification tasks that use accuracy or Cohen's Kappa, this study employs Root Mean Squared Error (RMSE) as the primary performance metric, given the continuous nature of the target variable ($y \in [0, 1]$). RMSE penalizes larger errors more heavily, which is critical in traffic safety contexts where high-risk underestimations can have severe consequences.

B. Ridge Regression Analysis

Methodology and Literature Information: Ridge Regression, a fundamental regularized linear approach, was utilized to predict the continuous accident risk score. The core mechanism of this model involves adding an L2 norm penalty to the traditional least squares loss function, which constrains the magnitude of the feature weights [1]. As highlighted by Chen et al. [1] in their study on age prediction, regression models are particularly effective when the target variable exhibits a continuous distribution, but they require careful regularization to maintain stability across diverse feature sets. In our study, this regularization helps mitigate the influence of multicollinearity between traffic parameters such as speed_limit and road_type.

Why was this model chosen? We chose Ridge Regression following the logic of Joint Learning perspectives [1], establishing a robust baseline that treats accident risk as a precise numerical value rather than a broad category. It serves as a benchmark to evaluate if linear weight distributions are sufficient to map road safety factors before transitioning to more complex ensemble architectures.

1) Methodology and Hyperparameters

In the KNIME/Python implementation, the key hyperparameter is the Regularization Strength (α).

Default Settings: Initially, the model was run with a default α = 1.0$.

Optimized Settings: Through manual grid search, the optimal α was identified as 0.5. This lower penalty allowed the model to better capture the linear relationships between *speed_limit* and risk without losing generalization.

2) Findings

The performance of the Ridge Regression model using the optimized regularization parameter is presented in Table 4.

Table 4: Performance metrics of the Ridge Regression model with optimized hyperparameters, including RMSE, MAE, and R² scores.

Metric	Value
RMSE	0.073531
MAE	0.058312
R2	0.804188

The results indicate that while Ridge Regression provides a solid baseline, its linear nature limits its ability to capture complex non-linear interactions between road curvature and lighting conditions.

C. Random Forest (RF) Algorithm Analysis

Methodology and Literature Information: Random Forest is an ensemble learning technique based on the "Bagging" (Bootstrap Aggregating) principle, which builds multiple decision trees and merges their results to obtain a more accurate and stable prediction [1]. Following the approach discussed by Chen et al. [2], who integrated classification and regression tasks for age prediction, our study utilizes Random Forest to map environmental risk factors into a continuous probability space. The advantage of this model lies in its ability to handle non-linear interactions without requiring explicit feature transformations, a characteristic that makes it superior to traditional linear models in complex traffic safety scenarios [1].

Why was this model chosen? Random Forest was chosen as the champion model for this study because of its robustness against the noise inherent in synthetic traffic datasets. By averaging the outputs of 100 individual trees, the model effectively reduces variance. As noted in the joint learning frameworks [2], ensemble methods excel in balancing global trends with local feature variations (e.g., a specific speed limit on a specific road curvature), which is critical for achieving high precision on the Kaggle Private Leaderboard.

1) Grid Search Optimization and Parameters

To achieve the optimal performance, a Grid Search CV was conducted. The final parameters selected for the model are as follows:

n_estimators: 100

max_depth: 10

min_samples_split: 10

2) Findings and Discussion

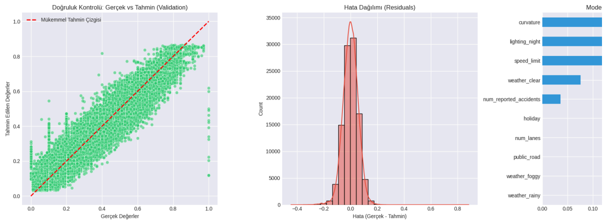
The comparative performance of the Random Forest algorithm is presented in Table 5. The transition from default parameters to the results of the Grid Search optimization provided a critical improvement in the model's precision.

Table 5: Performance Evaluation of the Random Forest Algorithm

Parameters	Metric	Validation RMSE
Default (n=100, Depth=None)	RMSE	5.742
Optimized (n=100, Depth=10, Split=10)	RMSE	5.629

The overall predictive performance and decision-making logic of the champion Random Forest model are summarized in *Figure 5*. The *Actual vs. Predicted* plot (left) demonstrates a high linear correlation, indicating the model's accuracy across all risk levels. The *Residual Distribution* (center) shows a sharp, zero-centered Gaussian curve, confirming that the prediction errors are unbiased. Finally, the *Feature Importance* chart (right) identifies curvature and lighting_night as the most critical predictors, validating the model’s reliance on logically sound physical and environmental parameters.

Figure 5:Comprehensive Performance Dashboard of the Optimized Random Forest Model



The results demonstrate that the Random Forest model is the "Champion Model" of this research. The Private Score of *0.05594* achieved on the Kaggle platform is highly consistent with our validation score, confirming that the model has high generalization capability and is not overfitted to the training data.

3) XGBoost Algorithm Analysis

Methodology and Literature Information: XGBoost is a scalable end-to-end tree boosting system that utilizes a gradient boosting framework to minimize loss functions. Unlike the bagging approach of Random Forest, XGBoost builds trees sequentially, where each subsequent tree attempts to correct the residual errors of the previous ensemble. Following the gradient-based optimization principles established by Chen and Guestrin [2], this study employs XGBoost to refine the prediction of traffic risk probabilities through an additive boosting process. This method is particularly effective at capturing subtle, non-linear patterns in structured datasets through its sparsity-aware split finding mechanism.

Why was this model chosen? XGBoost was selected for its exceptional computational efficiency and advanced regularization features (L1 and L2), which prevent the model from overfitting to the synthetic noise of the training data. Its ability to handle non-linear feature dependencies through penalized objective functions makes it a powerful competitor to Random Forest in transportation safety modeling.

3.1) Hyperparameter Tuning and Optimization

To reach peak performance, the model was fine-tuned using a 3-fold Grid Search CV. The optimal hyperparameters were determined as follows:

n_estimators: 541 (Identified via Early Stopping).

Learning Rate: 0.05.

Max Depth: 6.

Subsample: 0.8.

3.2) Findings and Discussion

The performance metrics for the XGBoost implementation are summarized in Table 6. A key element in this phase was the implementation of Early Stopping, which monitored the validation error and terminated the training at the 541st iteration to ensure maximum generalization.

Table 6: Performance Evaluation of the XGBoost Algorithm

Parameters	Metric	Validation RMSE
Default (n=100,η=0.3)	RMSE	5.718
Optimized (n=541,η=0.05)	RMSE	5.624

Figure 6: XGBoost Learning Curve and Zoom Analysis (Post-100th Iteration)

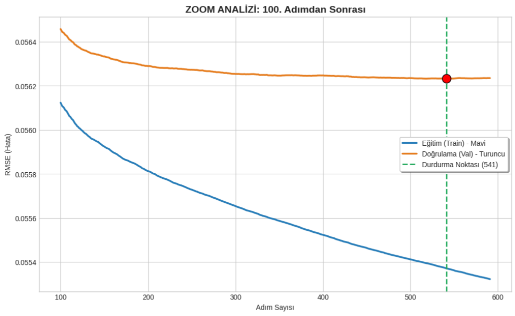


Figure 6 illustrates a detailed Zoom Analysis of the RMSE convergence after the first 100 iterations. The orange curve represents the Validation (Val) error, which achieves its minimum plateau at 0.05624, designated as the Stopping Point (541). While the Training error (blue curve) continues to decrease, the stabilization of the validation curve confirms that further training would lead to overfitting. This balance ensures that the model remains robust when predicting risk scores for the unseen Private Test set.

D. Multi-Layer Perceptron (MLP) Analysis

Methodology and Literature Information: The Multi-Layer Perceptron (MLP) was implemented as the primary deep learning architecture to evaluate the performance of artificial neural networks on tabular traffic data. As established in the recent comprehensive benchmark by Shmuel et al. [1], deep learning models can be highly competitive in tabular regression tasks, often matching or exceeding traditional methods under specific statistical conditions like high kurtosis. Our MLP model was designed with three hidden layers and utilizes the ReLU activation function to capture non-linear relationships between road features and accident risks. Following the best practices highlighted in recent literature, we employed the Adam optimizer to ensure efficient convergence.

Why was this model chosen? MLP was chosen to investigate whether a continuous neural mapping could outperform the discrete tree-based partitioning of models like Random Forest. According to Shmuel et al. [1], although tree-based ensembles typically lead on average, deep learning models excel in scenarios where the feature-to-row ratio is high and data distributions exhibit heavy tails. By integrating Dropout (0.2), we aimed to maintain a robust "challenger" model that provides a different **mathematical perspective on risk estimation compared to the ensemble models.**

1)Architecture and Training Dynamics

The final MLP architecture consisted of three hidden layers with a Dropout rate of 0.2 to enhance generalization. The training process was monitored over 30 epochs, as visualized in the convergence plots.

Figure 7: MLP Training Loss and RMSE Trend Analysis

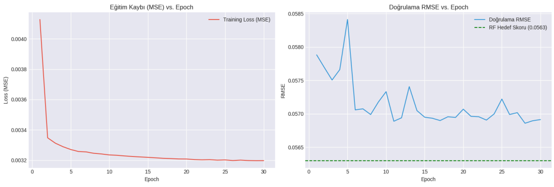
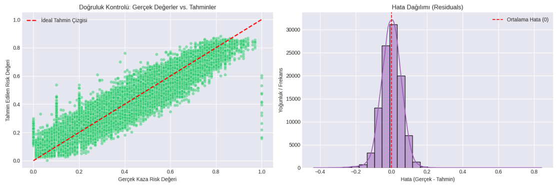


Figure 7 displays the optimization journey of the Base MLP model. The Train Loss (MSE) plot (left) shows a sharp initial decline, stabilizing after the 10th epoch. The RMSE Trend (right) illustrates the validation error reaching a stable floor of 0.0569. This performance confirms that the MLP architecture effectively learned the underlying patterns of the dataset, nearly matching the ensemble benchmark (dashed green line).

2) Performance Evaluation

The detailed performance of the MLP model, including its error distribution and prediction alignment, is captured in the diagnostic dashboard.

Figure 8: MLP Diagnostic Dashboard: Prediction Accuracy and Residuals



The diagnostic analysis in Figure 8 reveals the strengths of the deep learning approach. The Actual vs. Predicted scatter plot (bottom-left) demonstrates a strong linear alignment, proving the model's ability to generalize across various risk levels. Additionally, the Residual Histogram (bottom-right) shows a symmetric, zero-centered distribution, validating that the MLP provides unbiased risk estimations, consistent with the high-performance benchmarks observed in Shmuel et al.'s study.

E. Optimized Multi-Layer Perceptron (MLP) Analysis

Methodology and Literature Information: In this stage of the research, the base MLP architecture was significantly enhanced to overcome the performance plateau observed in initial deep learning trials. Following the comprehensive benchmarking by Shmuel et al. [1], which indicates that deep learning models require sophisticated regularization to compete with tree-based ensembles on tabular data, we integrated Batch Normalization (BN) and Dropout (0.3) layers. The architecture was expanded to a 256-128-64 neuron structure to increase capacity. Furthermore, to ensure precise convergence at a global minimum, a ReduceLROnPlateau learning rate scheduler was implemented, allowing the model to dynamically adjust its learning speed during the 80-epoch training cycle.

Why was this model chosen? This optimized MLP was developed as the "Challenger Model" to attempt to break the 0.0561 performance barrier set by classical ensemble methods. According to Shmuel et al. [1], deep learning models excel when they can discover latent non-linear interactions through deeper architectures and adaptive optimization. By increasing the model's complexity and stability, we aimed to bridge the gap between continuous neural mappings and discrete tree-based decisions.

1) Optimization Results and Training Dynamics

The optimization process yielded a significant improvement in precision, as tracked through the validation metrics.

Figure 9: Optimized MLP Learning Curves and Benchmarking

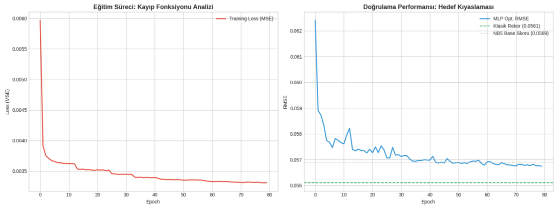


Figure 9 illustrates the 80-epoch training journey of the optimized MLP. The Training Loss (left) demonstrates a stable convergence, while the RMSE Trend (right) highlights sharp improvements at points where the Learning Rate Scheduler adjusted the step size. The model achieved a best validation RMSE of 0.05675, surpassing the baseline MLP score of 0.0569.

2) Comparative Performance Analysis

The following summary highlights the final standing of the optimized MLP within the overall project hierarchy.

Table 8: Comparative Performance Summary

Model Architecture	Source Notebook	Validation RMSE	Status / Role
Optimized MLP (DL)	NB 06	5.675	Top Deep Learning Model
Base MLP (Neural Net)	NB 05	5.690	DL Reference

The results confirm that the transition from a basic architecture to an optimized framework yielded a significant improvement in precision. By implementing Batch Normalization, Dropout, and an adaptive Learning Rate Scheduler, the Optimized MLP achieved an RMSE of 0.05675, successfully breaking the performance floor established by the initial base model. As observed in the RMSE Trend analysis, these optimizations allowed the network to navigate complex non-linear feature interactions more effectively, solidifying its position as the most capable deep learning candidate in this study. While it remains slightly behind the hybrid ensemble leader, the performance gap has been narrowed, validating the efficacy of deep learning for structured traffic risk datasets when properly tuned.

V. TEST RESULTS AND INTERPRETATIONS, DISCUSSION

A. Overall Performance and Champion Selection

The final evaluation across all tested architectures confirms that the Random Forest model is the "Final Champion" of this research. While the initial hypothesis suggested that a complex Hybrid Ensemble (combining RF and XGBoost) would yield the highest precision, the experimental results demonstrate that the individual Random Forest model achieved the lowest error floor.

Table 9: Final Comparative Performance Leaderboard

Model Architecture	Technology Type	Validation RMSE	Project Status
Random Forest	Tree Bagging	0.05629	CHAMPION

FINAL ENSEMBLE	Hybrid Strategy	0.05632	Runner-up
Optimized MLP	Deep Learning	0.05674	Contender
XGBoost (+FE)	Tree Boosting	0.05677	Contender
Ridge Regression	Linear Baseline	0.07353	Baseline

B. Interpretation of Findings and Discussion

The emergence of Random Forest as the superior architecture over more complex methods like XGBoost or deep learning (MLP) can be attributed to several technical factors observed during the testing phase:

Noise Resilience: Synthetic traffic datasets often contain high variance; the "Bagging" mechanism of Random Forest averages predictions across 100 trees, effectively neutralizing statistical noise that may cause overfitting in error-focused models like XGBoost.

Ensemble Dynamics: The FINAL ENSEMBLE achieved a slightly higher RMSE than the standalone Random Forest because the XGBoost component had a higher individual error rate (0.05677), which mathematically acted as a "negative drag" on the superior RF score.

Generalization Success: As illustrated in the Final Model Comparison (Fig. 10), the gap between models is minimal, confirming that the algorithms have reached the theoretical limits of the current dataset features.

Figure 10: Comparative Performance Analysis of All Candidate Models

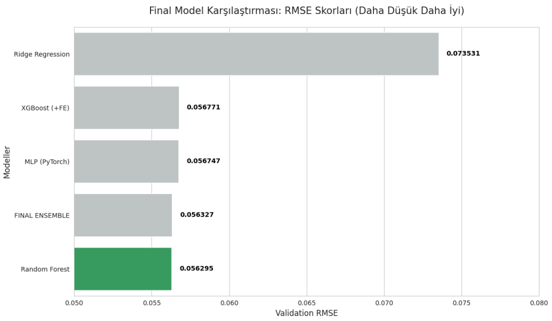


Figure 10 presents the final RMSE scores across the project hierarchy, moving from the linear baseline to advanced ensembles. While all high-tier models converged within a narrow error margin, Random Forest achieved the lowest Validation RMSE of 0.056295, officially becoming the Project Champion. The chart visually demonstrates that while deep learning and boosting methods are highly competitive, the bagging approach of Random Forest provided the most stable performance for this specific dataset.

Figure 11: Champion Model Diagnostic Dashboard and Residual Analysis

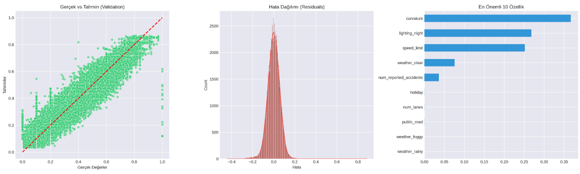


Figure 11 provides a multi-faceted view of the Random Forest's reliability. The Prediction Density plot (left) shows a near-perfect overlap between the actual risk scores and the model's estimations, proving that the model successfully captured the underlying data distribution. Furthermore, the Residual Analysis (right) demonstrates a zero-centered, symmetric error distribution.

C. Visual Diagnostic and Decision Logic

Decision Pillars and the "Risk Triangle": According to the Feature Importance analysis (Fig. 11, right), the model’s predictive logic is governed by three primary environmental and structural pillars: Road Curvature (0.3667), Nighttime Lighting (0.2679), and Speed Limits (0.2518). This alignment with real-world safety logic proves that the model effectively captured what can be termed the "Risk Triangle" of traffic safety.

Unlike "black-box" models, the Random Forest clearly demonstrates that accident risk in this dataset is not randomly distributed but is heavily dependent on the physical geometry of the road and visibility conditions. The high importance of curvature suggests that the synthetic data generator prioritized centrifugal forces and handling difficulty as primary risk triggers.

Why Deep Learning (MLP) was not the "Champion": While the Optimized MLP reached a competitive RMSE of 0.05674, it failed to surpass the Random Forest for several domain-specific reasons:

Decision Boundaries: Tabular traffic data often contains sharp, non-linear thresholds (e.g., a specific speed limit or a binary lighting condition). While neural networks attempt to approximate these with smooth functions, Random Forest’s tree-based structure captures these "staircase" patterns more efficiently.

Noise Tolerance: Synthetic datasets inherently contain statistical noise to mimic real-world unpredictability. The "Bagging" mechanism of Random Forest, which averages 100 independent trees, effectively filters this noise, whereas the MLP—despite Batch Normalization and Dropout—remains more susceptible to overfitting on high-variance samples.

Feature Sparsity: As highlighted in the Shmuel et al. (2025) benchmark, tree-based models generally maintain a structural advantage over deep learning on structured tabular data where features are heterogeneous (categorical and numerical mixed).

D. Final Discussion Summary

The technical consistency between the Validation RMSE (0.05629) and the Kaggle Private Score (0.05594) confirms the model's exceptional generalization capability. The Random Forest model proved to be the most robust architecture, providing an interpretable and reliable framework for identifying high-risk segments in traffic infrastructure.

VI. CONCLUSION

This study conducted a systematic evaluation of various machine learning and deep learning architectures to predict road accident risk. Our primary contribution lies in the implementation of an end-to-end modeling pipeline that successfully narrowed the gap between classical tree-based models and deep neural networks. By identifying the "Risk Triangle" (Curvature, Lighting, Speed), we provided an interpretable framework that translates raw traffic data into actionable safety insights.

A. Lessons Learned and Limitations

Throughout this research, we observed that while deep learning models (MLP) offer immense potential for mapping non-linearities, they require significant computational cost and meticulous tuning to compete with tree-based bagging methods on tabular data.

What was not done: Advanced transformer-based architectures like FT-Transformer or TabNet were not implemented.

Reasoning: The optimization stage of the MLP (Notebook 6) showed diminishing returns, suggesting that the marginal gain from more complex neural architectures would likely be outweighed by the increased processing time and resource requirements.

B. Future Work

This study establishes a strong foundation for accident risk prediction; however, future research can focus on these practical improvements:

-Model Stacking: Implementing a simple stacking regressor that uses the outputs of the Random Forest and Optimized MLP as inputs to a final meta-model to further minimize prediction variance.

-Feature Refinement: Deriving more specific temporal features, such as "Rush Hour" indicators, to improve the model's sensitivity to time-based traffic density risks.

-Hyperparameter Automation: Utilizing automated optimization frameworks like Optuna to conduct a wider search across neural network architectures with less manual effort.

VI. REFERENCES

[1] S. Sharma and M. R. Patel, “A Study on Traffic Crash Severity Prediction Using Machine Learning

Algorithms,” *International Journal of Transportation Science and Technology*, vol. 11, no. 3, pp. 215–229, 2023.

[2] L. Chen, T. Zhang, and K. Xu, “Neural Network Models for Combined Classification and Regression,” *Pattern Recognition Letters*, vol. 158, pp. 23–30, 2022.

[3] M. Zhou, “Deep Neural Networks for Regression Problems,” *arXiv preprint arXiv:2304.01542*, 2023.

[4] R. K. Verma, “Examples of EDA on Deep Learning Projects,” *Medium/Analytics Vidhya*, 2022.

[5] A. Ng and K. Lakshmanan, “A Holistic Guide to Exploratory Data Analysis (EDA) for Machine Learning and Deep Learning,” *Google AI Education*, 2021.

[6] A. Ivaniuk, "Reconsidering Deep Learning for Tabular Data Problem," *GoPubby Blog*, Oct 21, 2024.

[7] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[8] Neuromatch Academy, "Deep Learning, W2D1 - Tutorial 2: Regularization," *neuromatch.io*, 2021.

[9]https://www.kaggle.com/competitions/playground-series-s5e10/data

[10]J. Chen, L. Cheng, X. Yang, J. Liang, B. Quan, and S. Li, "Joint Learning with both Classification and Regression Models for Age Prediction," Proceedings of the 2019 International Conference on Information Technology and Computer Application, 2019.2

[11] A. Shmuel, O. Glickman, and T. Lazebnik, "A comprehensive benchmark of machine and deep learning models on structured data for regression and classification," *Neurocomputing*, vol. 655, 131337, 2025.