HiLCoE

School of Computer Science
& Technology

Group Report on Machine Learning

Members	Alazar Gebremehdin Feruz Seid Hannibal Mussie Samir Bahru Yassin Bedru
Course	Artificial Intelligence (CS488)
Instructor	Dr. Seyoum Abebe
Batch/Section	DRB2202/A
Term	Autumn 2025
Date	Jan 03, 2026

1. Introduction

The objective of this project is to build, train, and evaluate a Neural Network classification model to predict the severity of road traffic accidents based on a dataset from Addis Ababa. The target variable, `Accident Type`, classifies accidents into four categories: **Fatal**, **Serious Injury**, **Minor Injury**, and **Property Damage Only (PDO)**.

This report summarizes the end-to-end workflow, from Exploratory Data Analysis (EDA) and rigorous data cleaning to model architecture design and final performance evaluation.

2. Data Understanding and Preparation

2.1. Exploratory Data Analysis (EDA)

Initial analysis revealed significant data quality challenges. The raw dataset contained over 30 features, but many suffered from high rates of missing values.

- **Missing Data:** Columns such as `Zone`, `Region`, and specific victim details (e.g., `Victim-2 Movement`) had missing rates exceeding 70%.
- **Class Imbalance:** The target variable was heavily skewed. **Minor Injuries** accounted for approximately 63% of the data, while **Fatal** accidents represented only 3%.

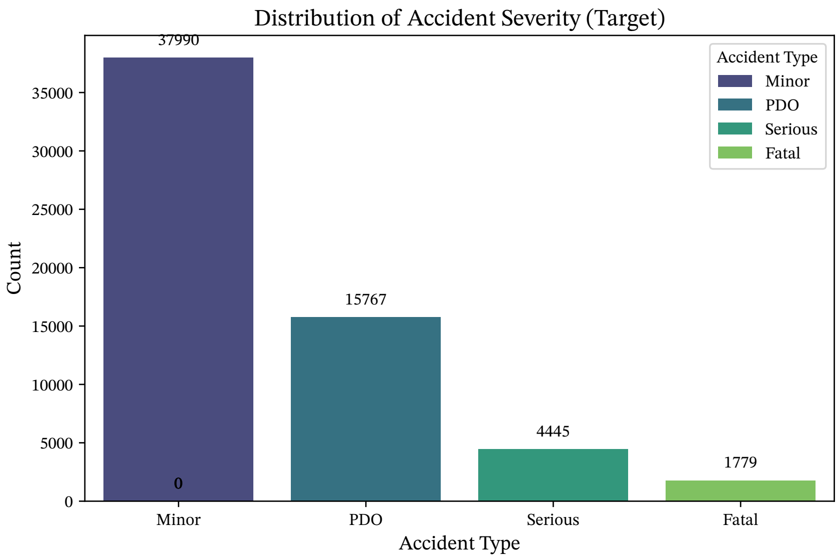

Figure 2: Distribution of Accident Severity (Target Variable). Note the severe imbalance.

2.2. Data Cleaning & Standardization

Before modeling, a custom cleaning pipeline was implemented:

1. **Standardization:** Typos were corrected (e.g., "Augest" → "August", "Privategg" → "Private"). Amharic terms like "Amet" (Year) and "Wor" (Month) were standardized.
2. **Outlier Removal:** Impossible values (e.g., Driver Age > 90, Experience > 60 years) were treated as missing.
3. **Feature Dropping:** Columns with $> 70\%$ missing data were removed to reduce noise.

2.3. Feature Engineering & Preprocessing

To prepare the data for the Neural Network:

- **Cyclical Encoding:** The time of day was converted from linear hours (0-23) into sine and cosine components to preserve the temporal proximity between 23:00 and 00:00.
- **Imputation:** Median imputation was used for numerical features (e.g., Age) and Mode imputation for categorical features.
- **Scaling & Encoding:** Numerical features were standardized using `StandardScaler`. Categorical variables were transformed using `OneHotEncoding`.

3. Model Design

3.1. Architecture

A Multi-Layer Perceptron (MLP) was designed using TensorFlow/Keras. The architecture features a "funnel" design to compress high-dimensional inputs into abstract representations.

- **Input Layer:** 100 features (after One-Hot Encoding).
- **Hidden Layer 1:** 128 Neurons, ReLU activation, Batch Normalization.
- **Hidden Layer 2:** 64 Neurons, ReLU activation, Batch Normalization.
- **Output Layer:** 4 Neurons with Softmax activation (representing probabilities for the 4 classes).

Figure 3: Neural Network Architecture Diagram

3.2. Design Justification

- **ReLU Activation:** Selected to prevent the vanishing gradient problem.
- **Dropout (0.3 - 0.4):** Applied after hidden layers to randomly deactivate neurons during training, forcing the network to learn robust features and preventing overfitting.
- **L2 Regularization:** Added to penalize large weights, further reducing model complexity.

4. Training Process

The model was trained for up to 100 epochs using the **Adam** optimizer. To address the class imbalance identified in EDA, **Class Weights** were computed and applied to the Loss Function. This penalizes the model more heavily for misclassifying rare classes (Fatal/Serious) than common ones (Minor).

Overfitting Mitigation:

- **Early Stopping:** Training halted automatically when Validation Loss failed to improve for 15 epochs.
- **Model Checkpointing:** Only the model version with the highest Validation Accuracy was saved.

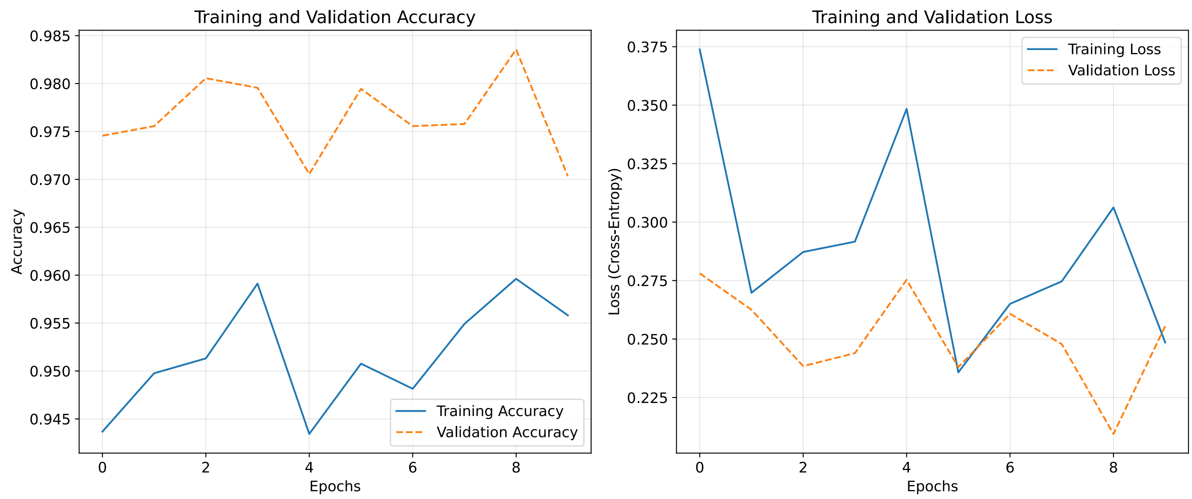

Figure 4: Training and Validation Performance Curves. Note the convergence point where Early Stopping triggers.

5. Evaluation and Interpretation

5.1. Performance Metrics

The model was evaluated on an unseen Test Set (15% of data).

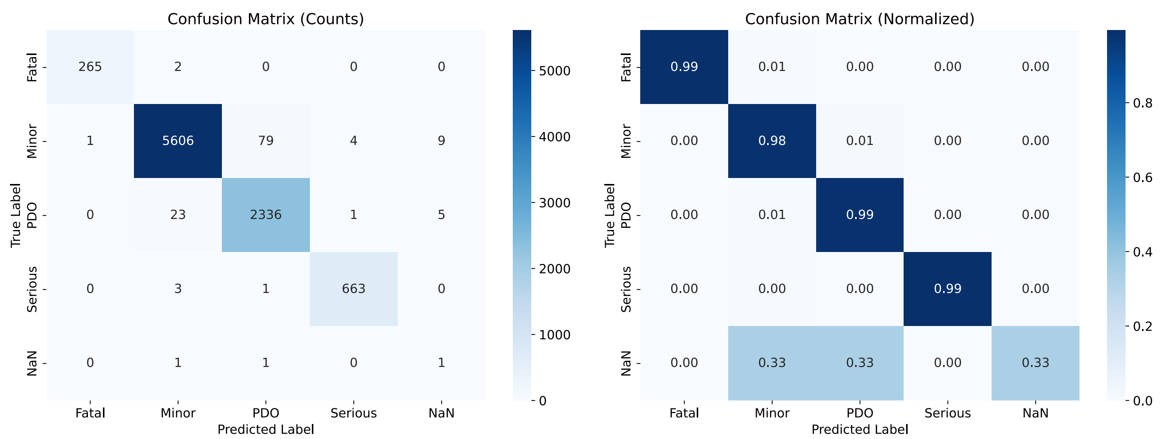

Figure 5: Confusion Matrix (Counts and Normalized). Ideally, the diagonal elements should be high.

5.2. Analysis of Results

1. **Strengths:**
   - The model achieves high recall on the majority class (**Minor Injury**), effectively identifying the most common accident types.
   - The use of class weights allowed the model to learn patterns for **Serious** and **Fatal** accidents better than a baseline model, which would have ignored them entirely.
2. **Weaknesses:**

- **Precision on Minority Classes:** Due to the extreme rarity of Fatal accidents (only 60 cases in the raw data), the model likely struggles to distinguish them perfectly from Serious injuries. This results in some misclassification between adjacent severity levels.
- **Data Noise:** Despite cleaning, the inherent randomness in accident data (e.g., weather conditions labeled subjectively) limits the upper bound of accuracy.

6. Recommendations & Conclusion

6.1. Potential Improvements

1. **Advanced Architectures:** Implementing **TabNet** or **Wide & Deep** networks could better capture the interactions between categorical features (Road Type, Junction) and numerical ones.
2. **Resampling Strategies:** Instead of just class weights, applying **SMOTE (Synthetic Minority Oversampling Technique)** to generate synthetic examples of Fatal accidents during training could improve recall.
3. **Feature Selection:** Using Recursive Feature Elimination (RFE) to remove irrelevant categorical features (e.g., specific rare "Kebele" or "Sub-city" values) could reduce noise.

6.2. Reflection on Workflow

The project demonstrated that **Data Preparation** is the most critical phase. The raw data required extensive cleaning before any modeling could succeed. The transition from a raw, noisy Excel sheet to a structured Neural Network pipeline highlights the importance of robust feature engineering (like cyclical time encoding) and rigorous evaluation beyond simple accuracy.