# Distracted Driver Detection - With Encoder-only Transformer, ResNet, and LSTM Architectures

Sook hyun Lee, PhD

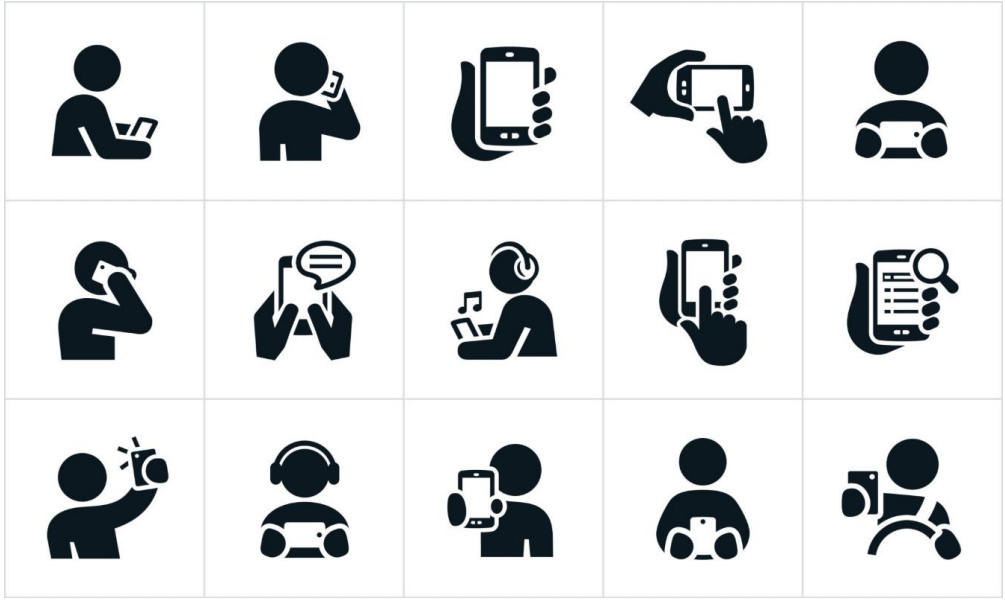
Data Science Career Track Capstone Project, Nov 2025

Springboard

# Problem statement

**Causes of Road Accidents:**
**Distracted, Fatigued Driving**

- **Distracted driving** diverts attention from the road
- **Fatigue and drowsiness** impair alertness and reaction time



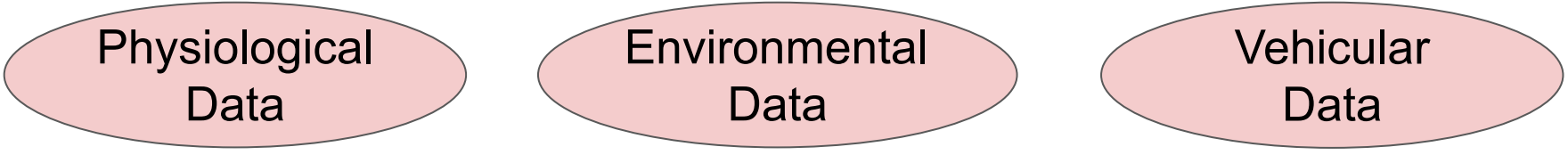
➢ How can machine learning be used to detect distracted or fatigued drivers and give them warnings?

➢ What types of information or features are effective for detecting distracted or fatigued drivers?

# Who Finds This Relevant

| | |
|---|---|
| Automotive Companies | Improve autonomous or semi-autonomous driving transitions |
| Tech Companies (sensors, software) | Apps or platforms that alert drivers to stay focused. |
| Insurance Companies | Incentivize safe driving. Determine liability in case of crashes. |
| Healthcare & Research institutions | Inform public health policies. Develop assistive technologies for impaired drivers. |

# Which Data Inputs Are Essential for Making Predictions?




- Heart rate (HR), heart rate variability (HRV)
- EEG
- EDA (skin conductance)
- Eye movement
- Facial EMG

- Road type
- Weather
- Lighting
- Traffic density
- Noise level
- GPS location

- Speed
- Acceleration
- Steering angle
- Brake pressure
- Lane position
- Throttle input

# Data Source and Summary

Data source: Kaggle

Ford Challenge Dataset

```
FordChallenge_X.csv
(Multi-modal Features),

FordChallenge_y.csv (Labels)

FordChallenge_subject_id.csv
(driver information, not used)
```

## Dataset Overview

- 600 real-time driving sessions,
- Each session duration: 2 minutes
- **Sampling rate:** Every **100 ms** (10 Hz)

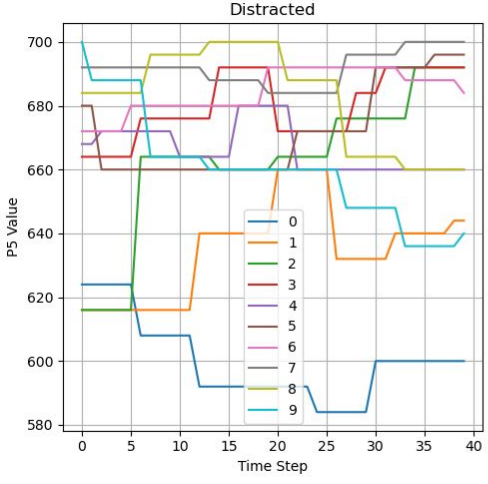
## Records contain:

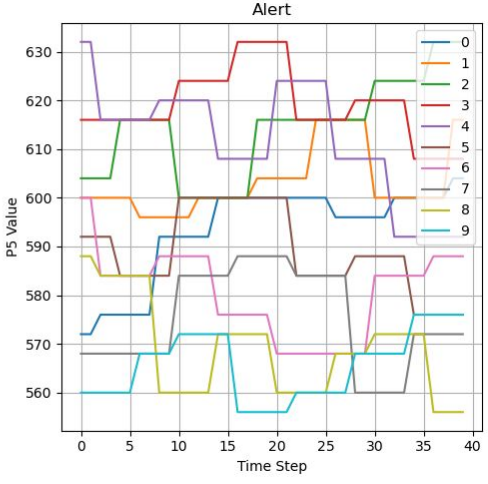
- **36257** multivariate time series
- In each time series:
  - **8** Physiological Features
  - **11** Environmental Features
  - **11** Vehicular Features
- Labels: 0 for **"distracted"** and 1 for **"alert"**

# Data Exploration - Physiological Data

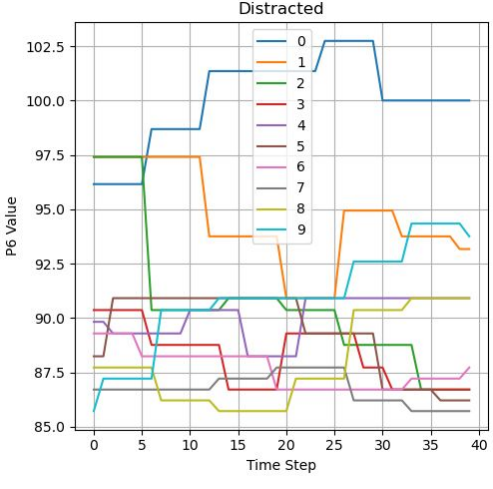
- Each panel shows 10 time series samples of a given feature for each category.

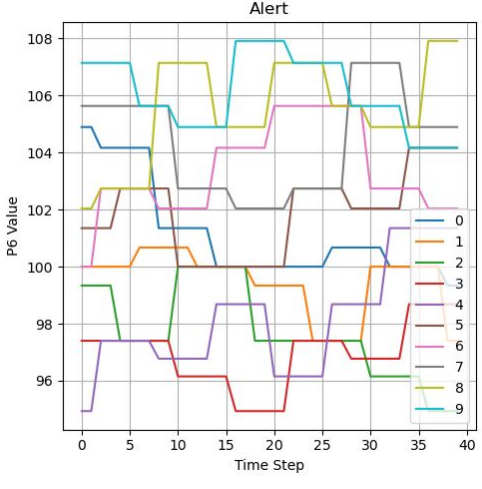
P5



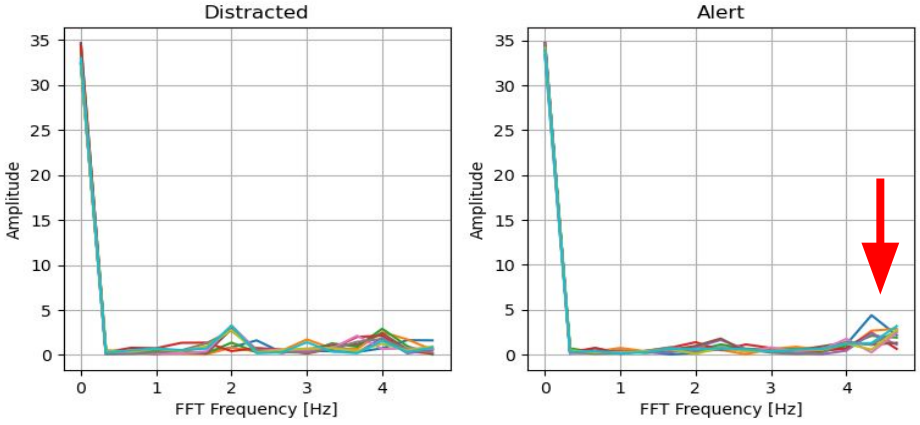

P6



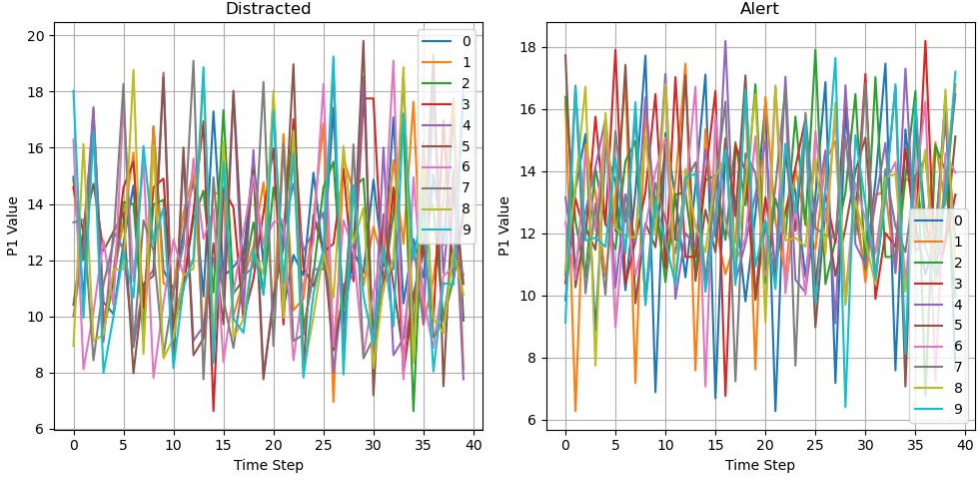

- Left panel: Distracted
- Right panel: Alert

# Physiological Data - ECG Signals




FFT

Represent heart's electrical activity

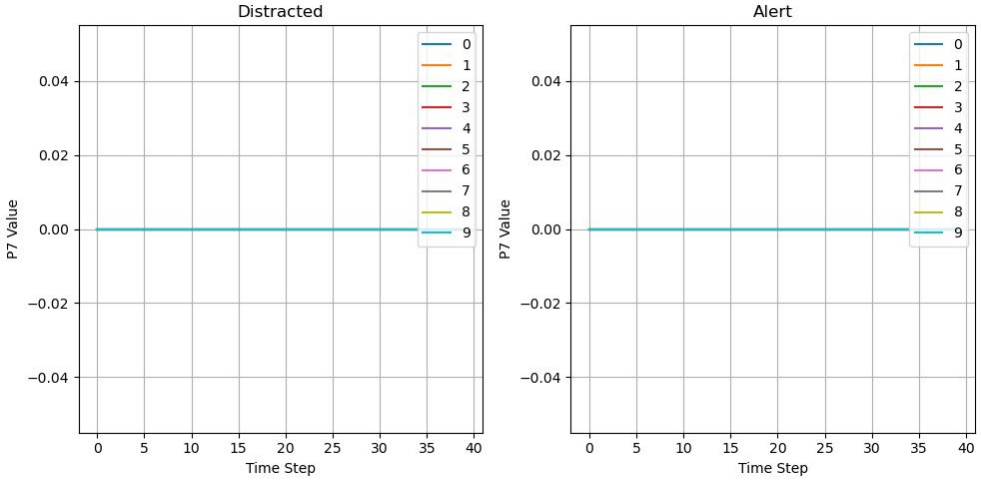



While deep learning algorithms are capable of learning complex feature interactions directly from raw data, model performance can be improved by

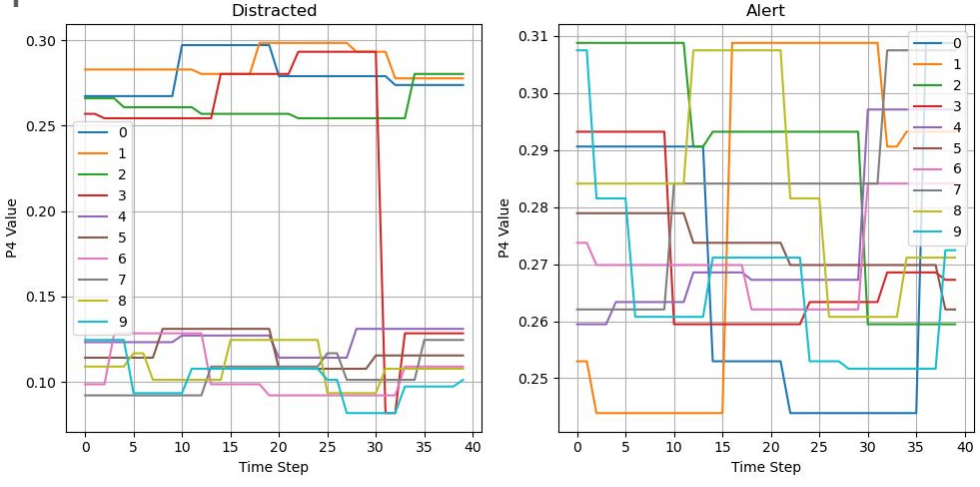
- Signal cleaning and filtering
- Normalization / Standardization
- Compute derived representations, e.g. FFT

# Physiological Data

P7




P4

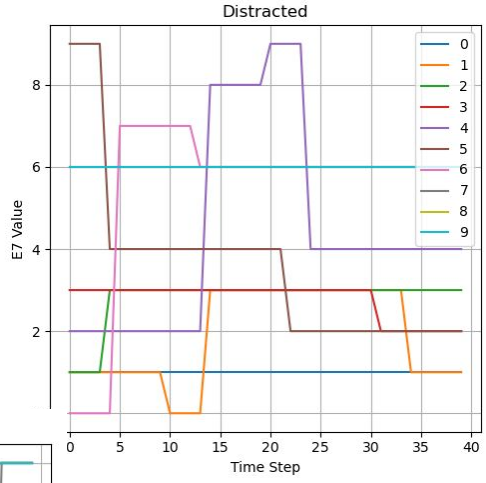


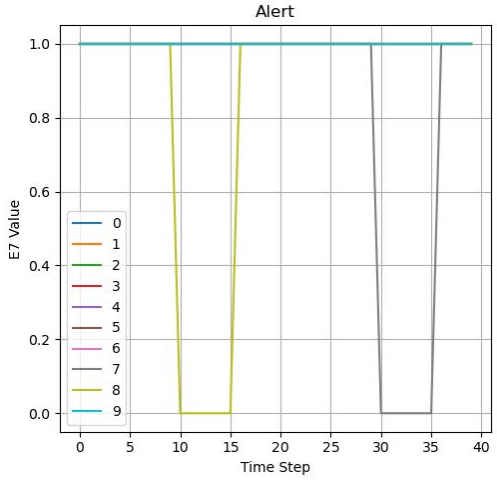

Feature P7: Constant Zero

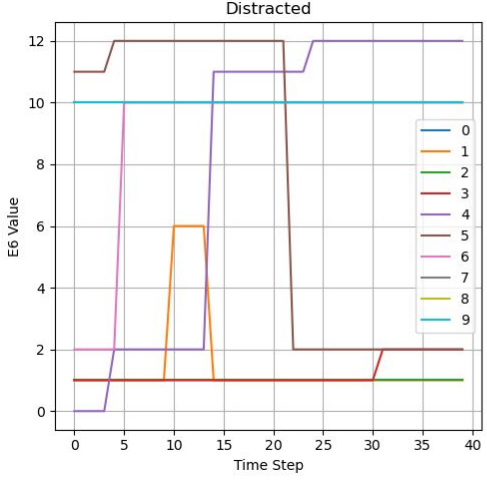
- ❏ Can contribute noise to dataset.
- ❏ Excluded from analysis.

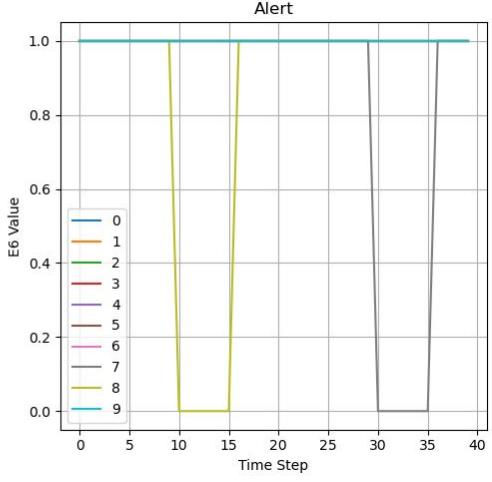
# Environmental Data

E7



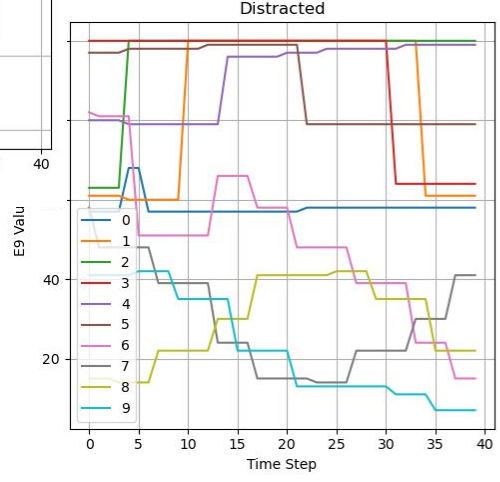



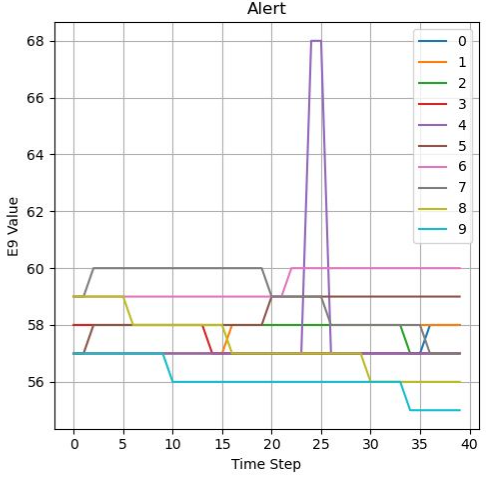



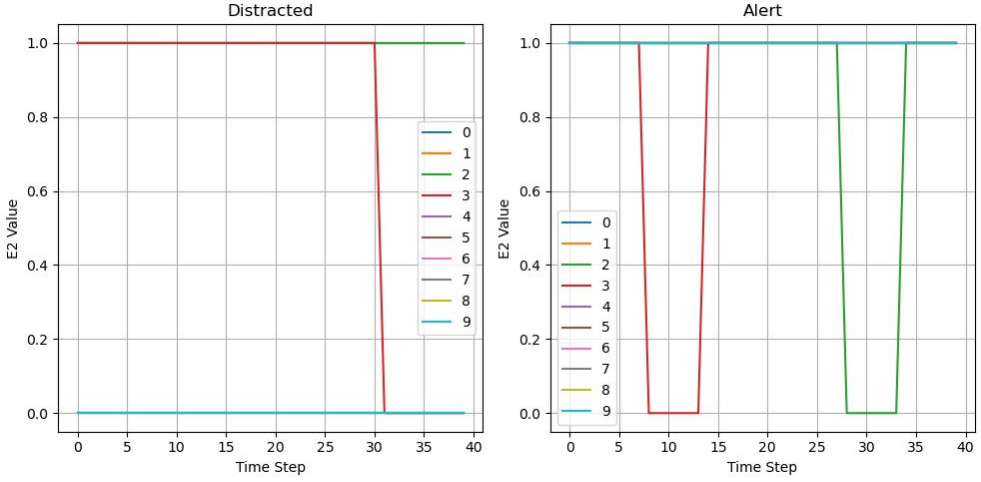

E6

E9



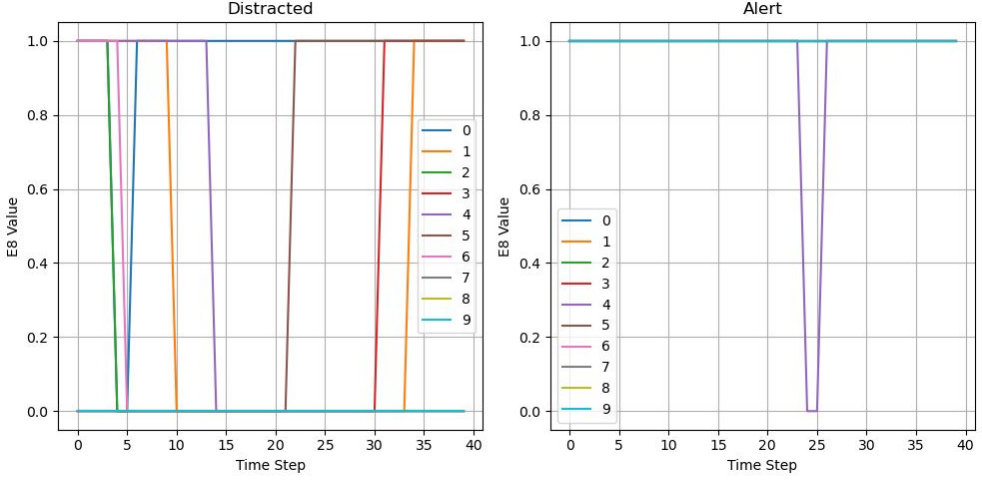



# Environmental Data
# E2 and E8 : Binary Data

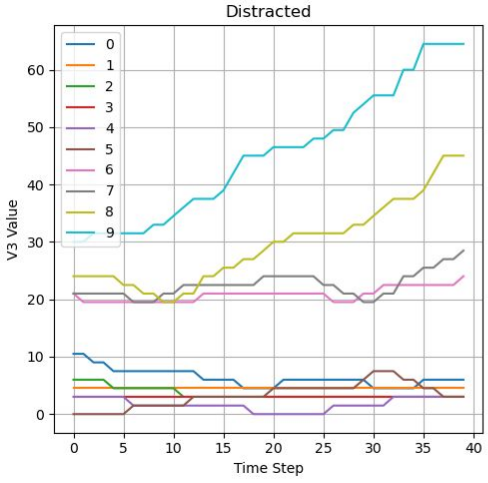


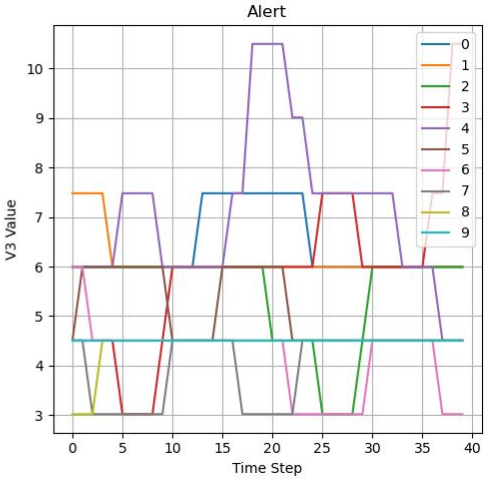

E2

E8



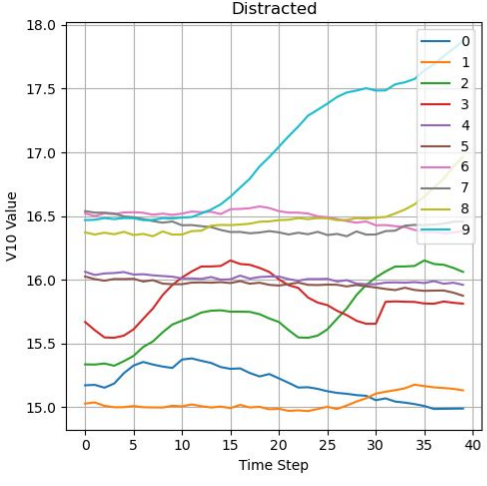
# Vehicular Data

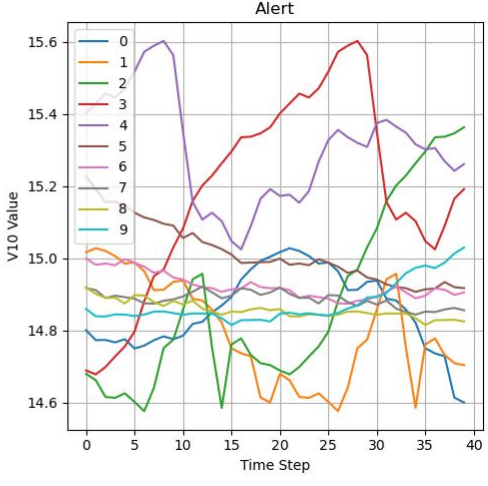



V3

V10



# Data Summary

- Physiological dataset
  - **5** *step-wise* and **1** (P0) *continuous* variables, 1 (P1) a high-frequency electrical signal, and **1** (P7) constant zero.

- Environmental dataset
  - **8** *step-wise* variables, **2** (E2, E8) *binary categorical* variables, and **1** *continuous* (E10) variable.

- Vehicular dataset
  - **5** *continuous* variables, **2** (V3, V9) *step-wise* variables, **1** (V2) *discrete* variable, **1** *binary* (V4) variable, and **2** (V6, V8) constant zero.

# Feature Engineering Steps

- Data reshaping and removal of zero-only features

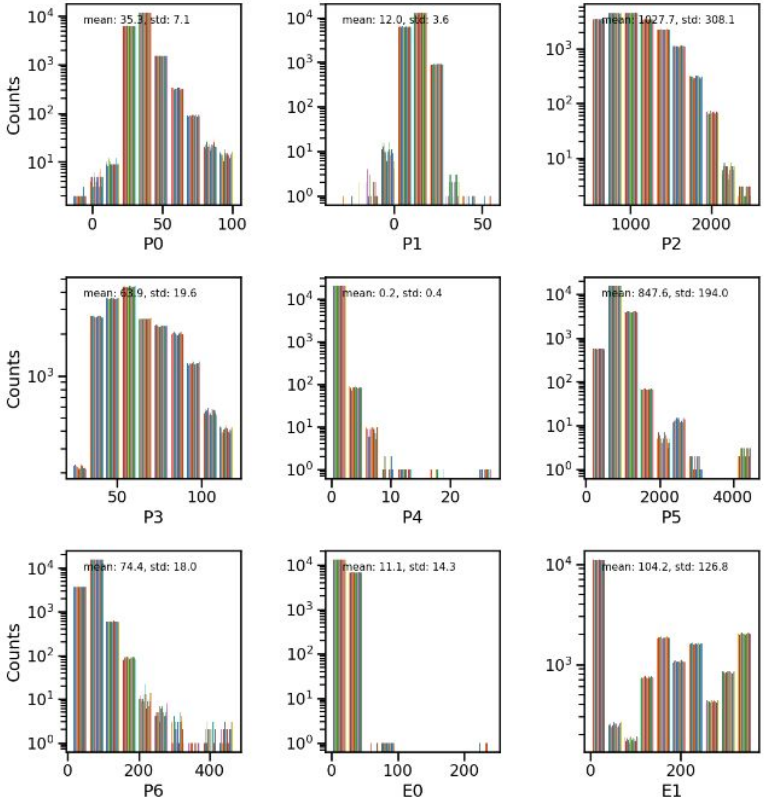
  `[num_samples, num_features, num_timesteps]`

- Identification of outliers

  **P5** contains outliers, which are removed.

- Normalization of data

  All features are normalized to the range **[0, 1]** before being used for model training.



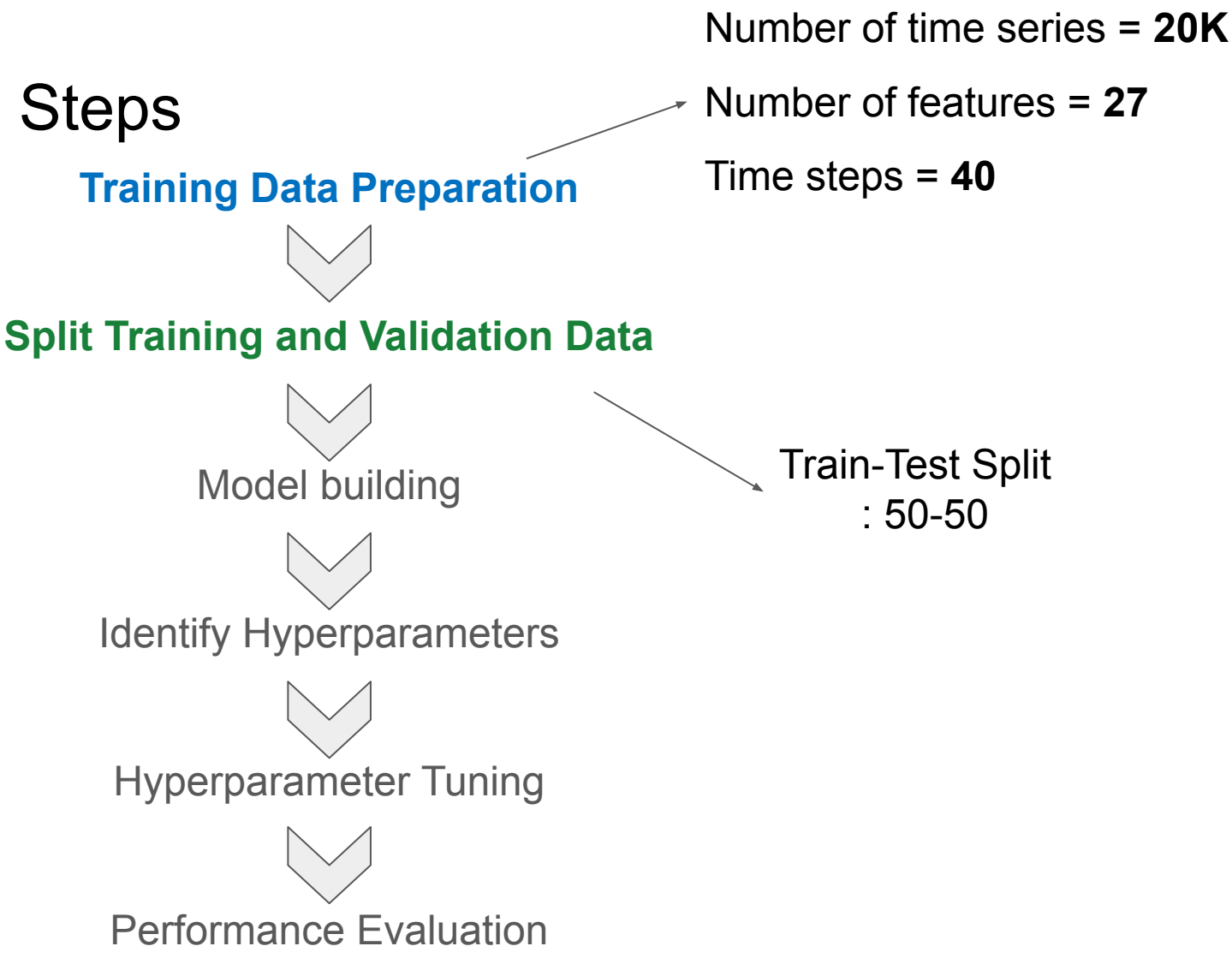
# Machine Learning Modeling Overview

- Time Series Classification
  - Central problem in machine learning
  - Applications spanning a wide range of domains
    - Healthcare (e.g., electrocardiogram signal analysis)
    - Finance (e.g., stock price prediction)
    - Environmental monitoring (e.g., weather and climate forecasting)
    - Industrial systems (e.g., fault detection and predictive maintenance)

# Approaches to Time Series Classification

- **Traditional ML methods:** Feature extraction + algorithms like SVM, Random Forest, k-NN
- **Deep learning models:** CNNs, RNNs, LSTMs, and Transformers for end-to-end learning
- **Hybrid approaches:** Combine handcrafted and learned features for improved accuracy
- **Representation learning:** Use embeddings or dimensionality reduction for temporal patterns
- **Ensemble methods:** Merge multiple models to boost robustness and performance

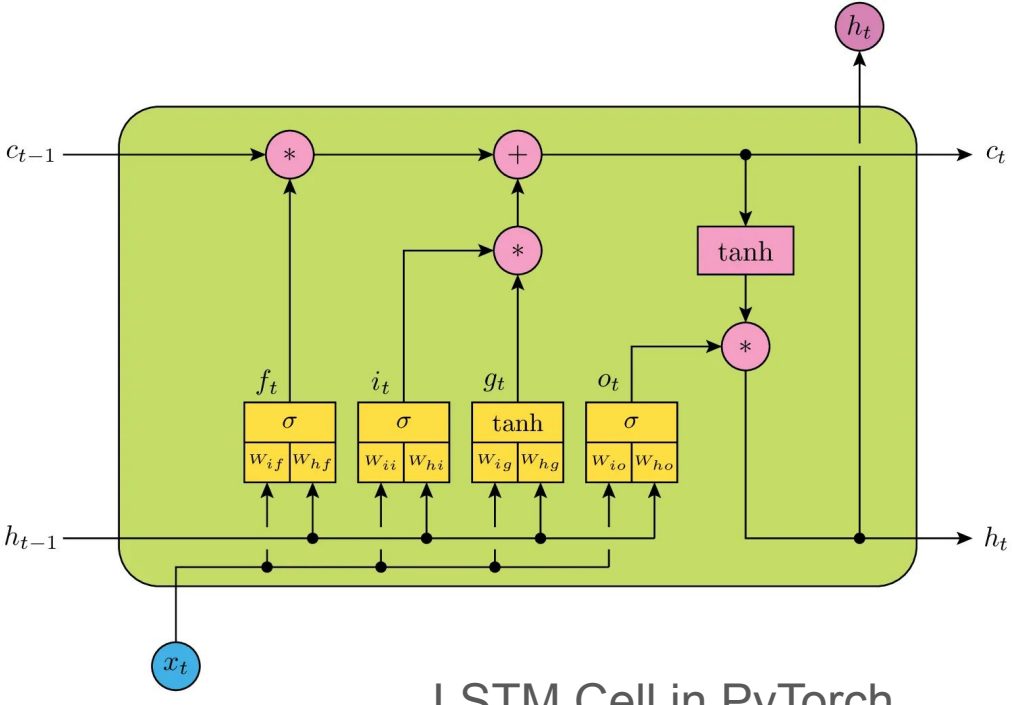
# Modeling Steps



# LSTM

Model building

- **LSTM layer:**
  - Processes temporal dependencies and outputs hidden states
- **Final hidden state:**
  - Bidirectional- Concatenate forward & backward states
- **Classification head:**
  - Maps final hidden state to output classes

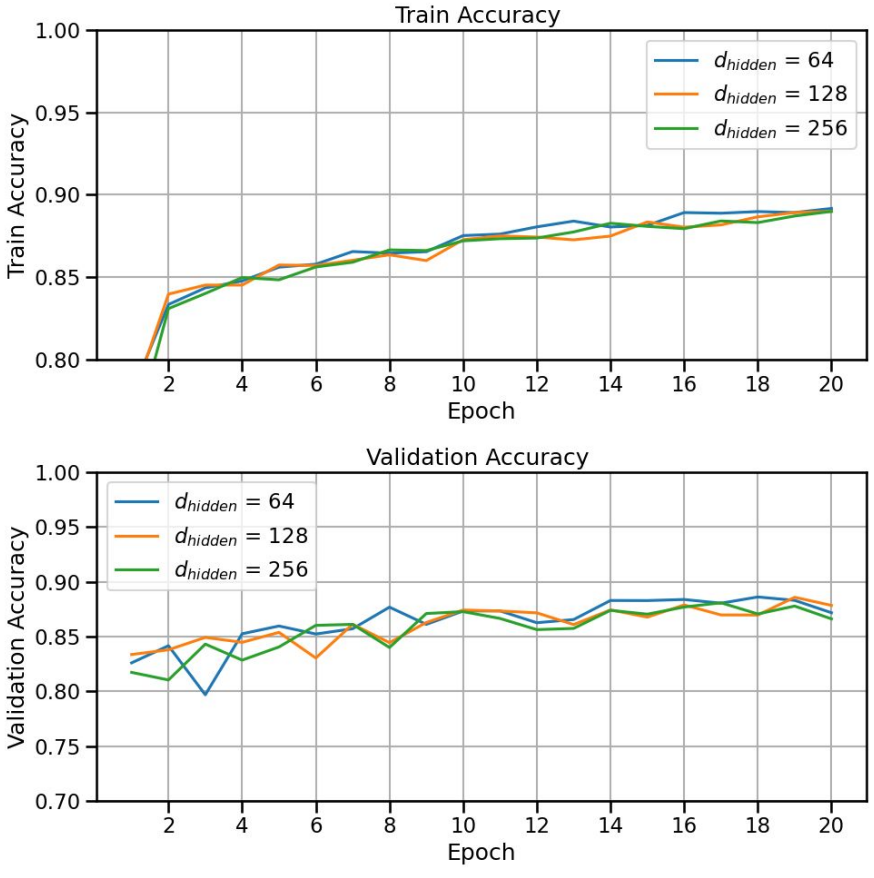



LSTM Cell in PyTorch

# Hyperparameter tuning for LSTM model

- Hidden layer dimension

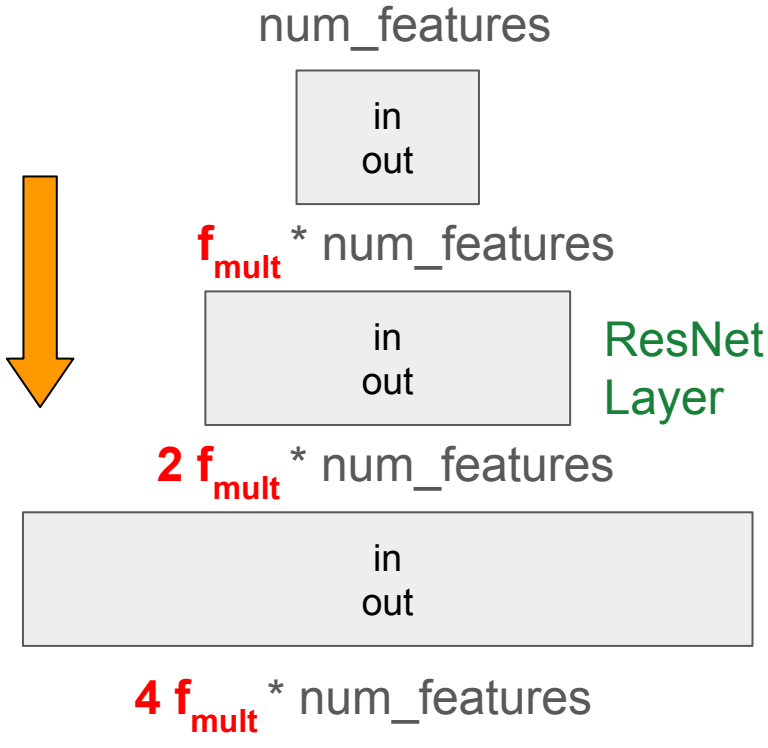
  $d_{hiddent}$ **= 128**



# ResNet

Model building

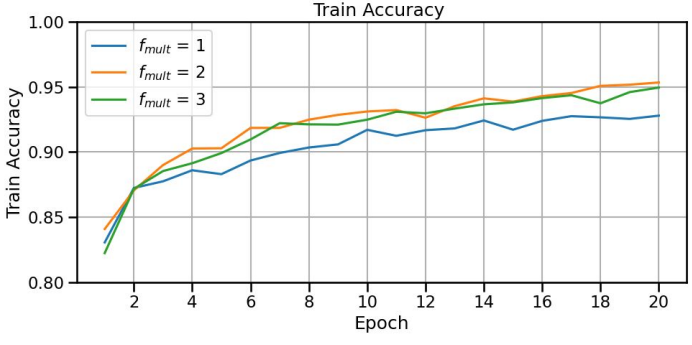
- Extract **local** temporal features and **hierarchical** representations through *one-dimensional* convolutions.
- **Residual connections** improves gradient flow and facilitates the training of deep networks.

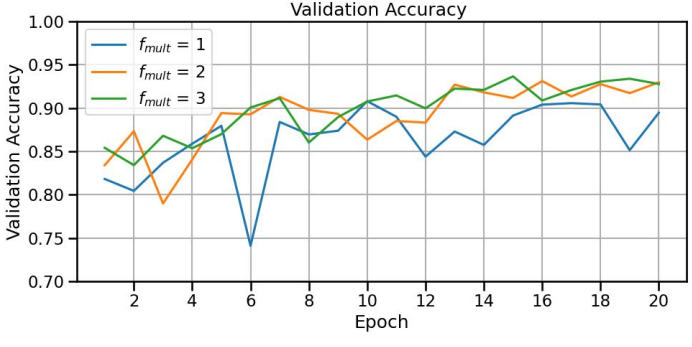


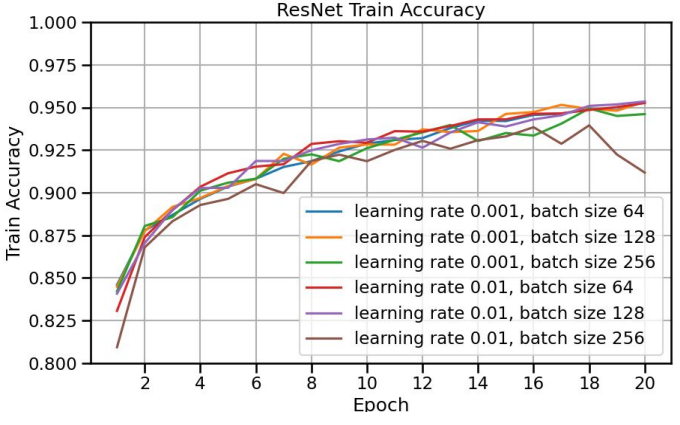

➔ This design increases the model's capacity to compensate for the loss of spatial or temporal resolution.

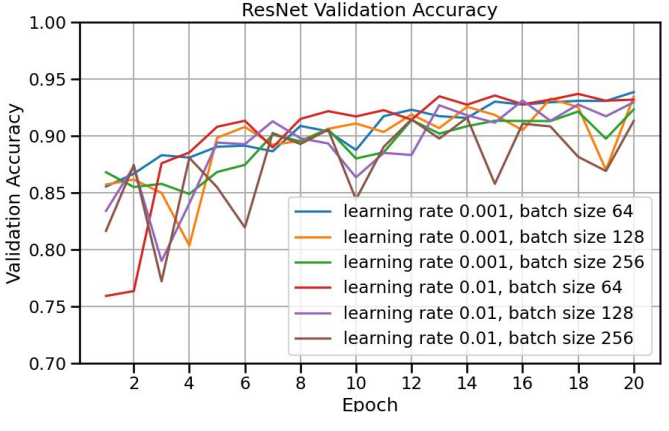
# Hyperparameter tuning for ResNet model

- Network size factor $f_{mult}$ = **2**
- Learning Rate = **64**, Batch Size = **0.01**



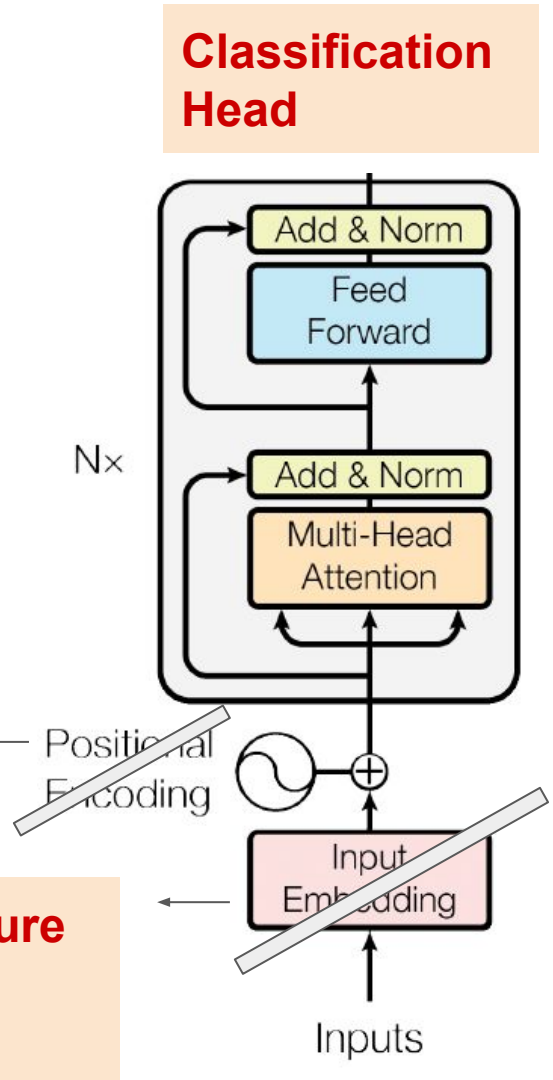



# Encoder-only Transformer

Model Building

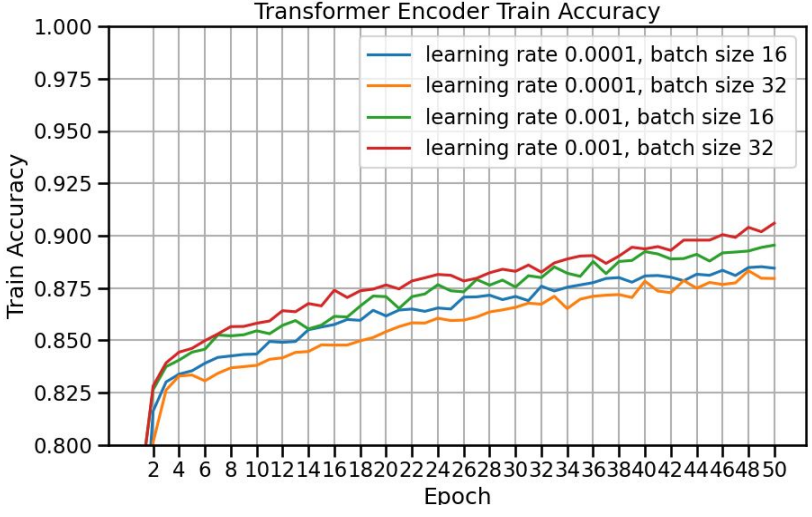
- Transformer encoder layer
  - Embedding
  - Multi-head self-attention
  - Position-wise feed-forward sublayers
  - Residual connections and layer normalization.

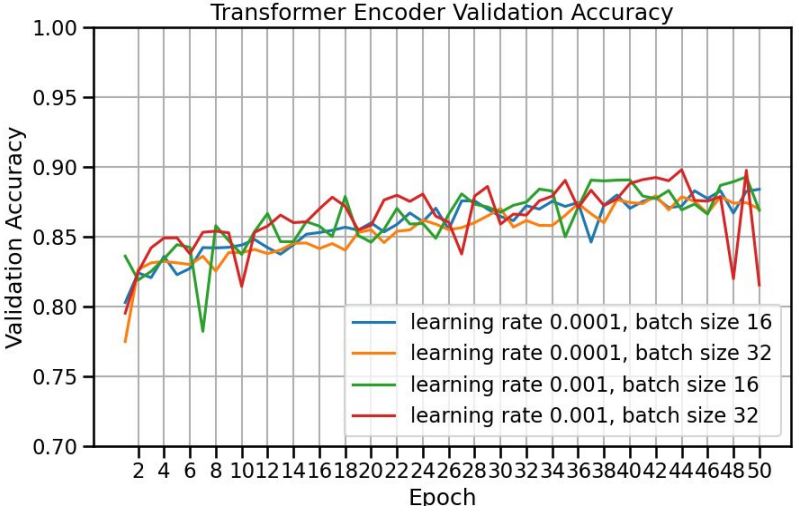
- Classification head
  - Predicts the target class based on the aggregated embeddings.



# Hyperparameter tuning for Self-attention Encoder model

- Learning Rate = **32**
- Batch Size = **0.001**

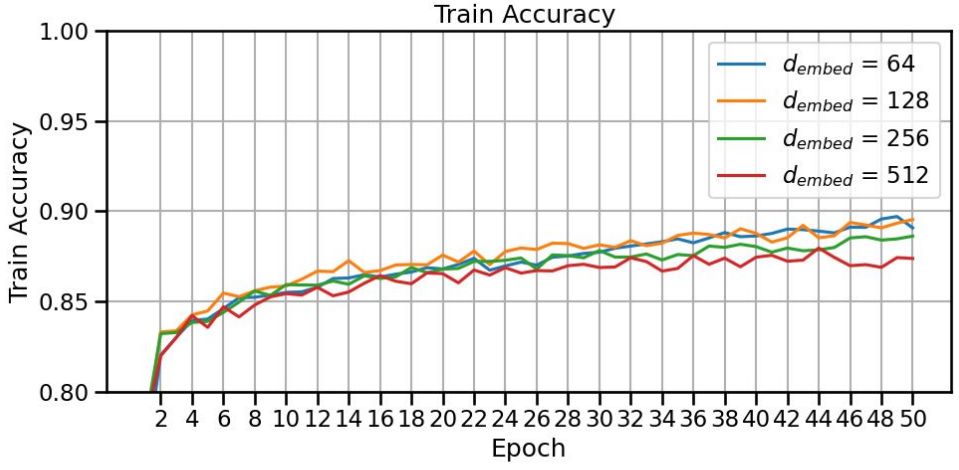


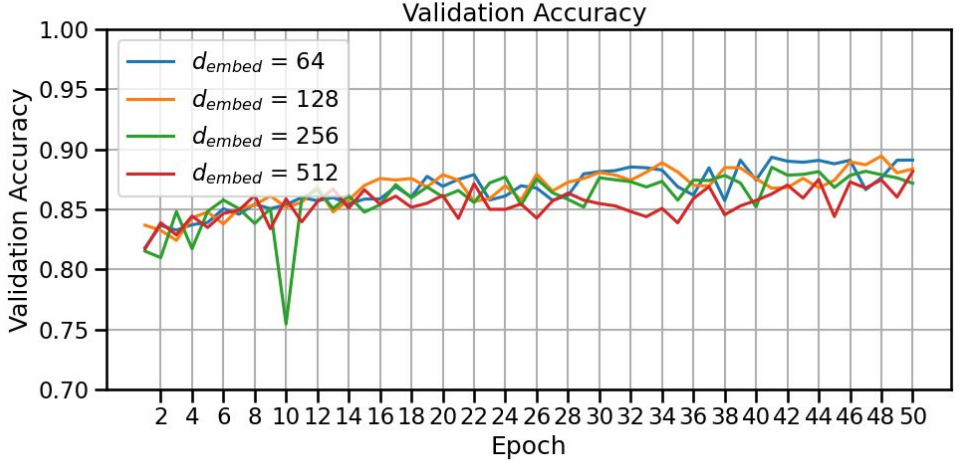



- Embedding dimension $d_{embed}$ **= 128**

## Embedding :

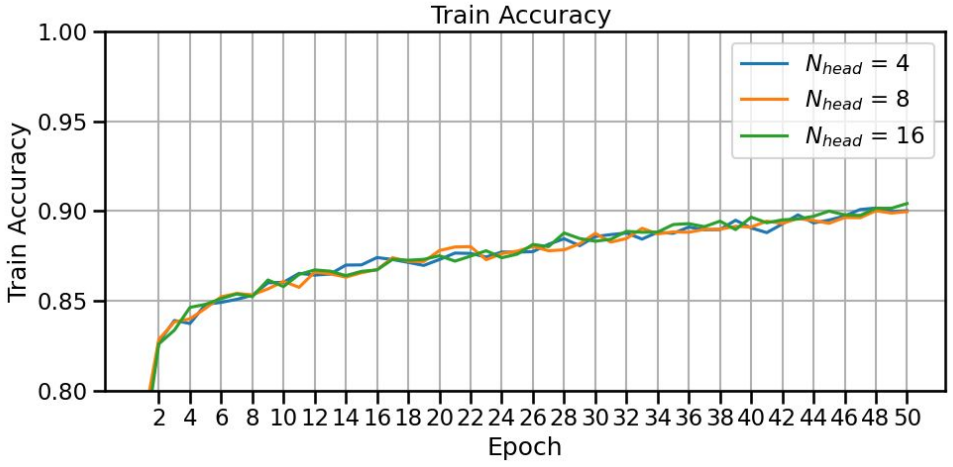
**Applied linear projection** to feature vectors and incorporated only **temporal embedding components**

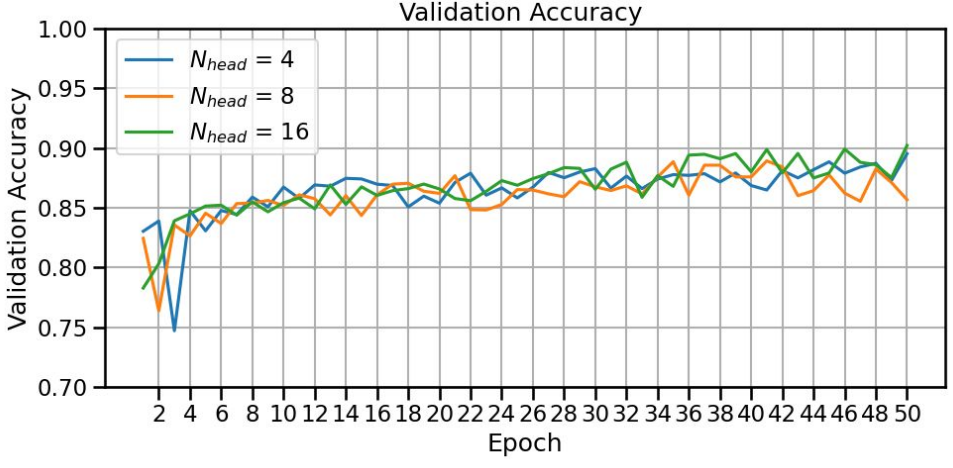


- Number of heads $N_{head}$ **= 4**

- **Self-attention mechanisms**
  - Capture complex temporal and contextual relationships within input sequences.
- **Number of attention heads**
  - Determines how the model distributes its representational capacity across different subspaces of the embedding.



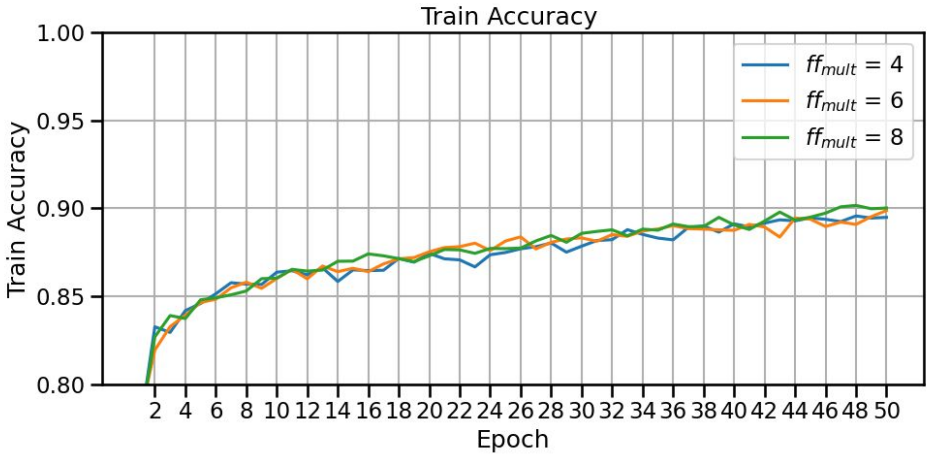



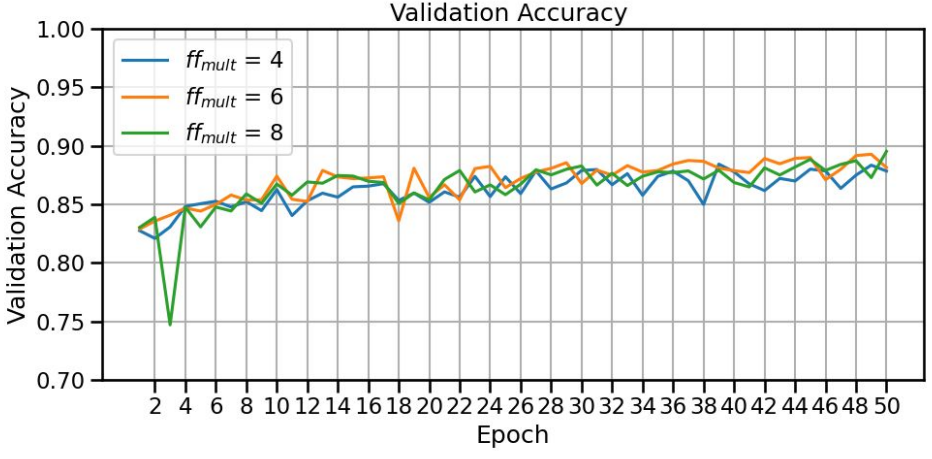
- Feed-forward network size

  $ff_{mult}$ = **6**

- **Self-attention layer**
  - Mixes information across time steps.

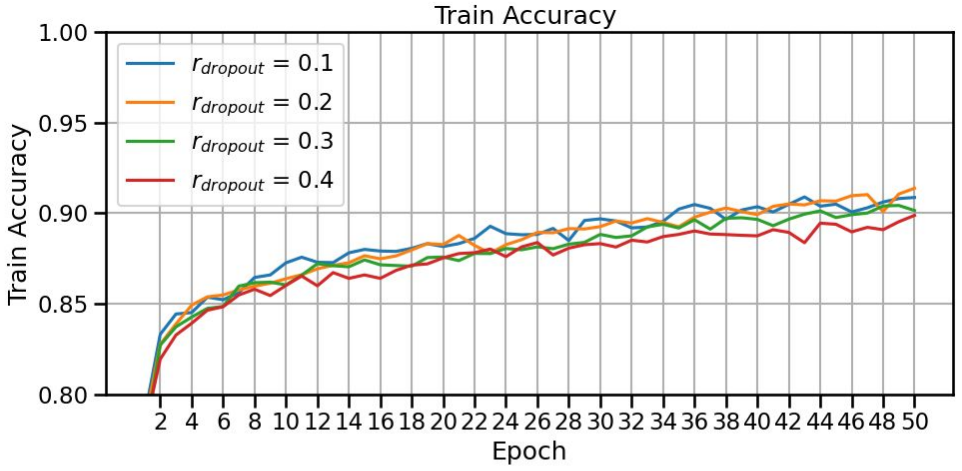
- **Feed-forward network**
  - Expands this information within each time step
  - Gives the model non-linear capacity to represent more complex transformations.

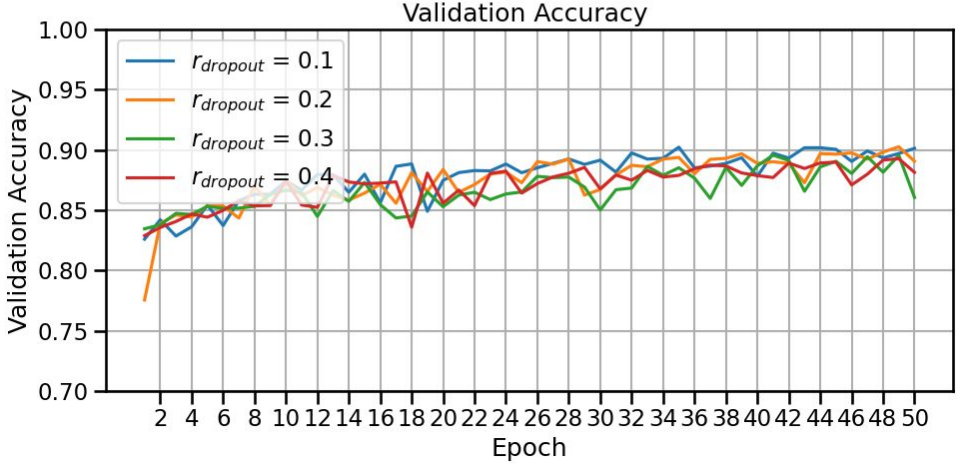


- Dropout rate $r_{dropoutt}$ = **0.1**

- **Dropout**
  - Prevent overfitting by randomly deactivating a subset of neurons during training.
  - Encourages model to learn robust and generalizable features.
  - Most influential hyperparameter for training quality for this work.



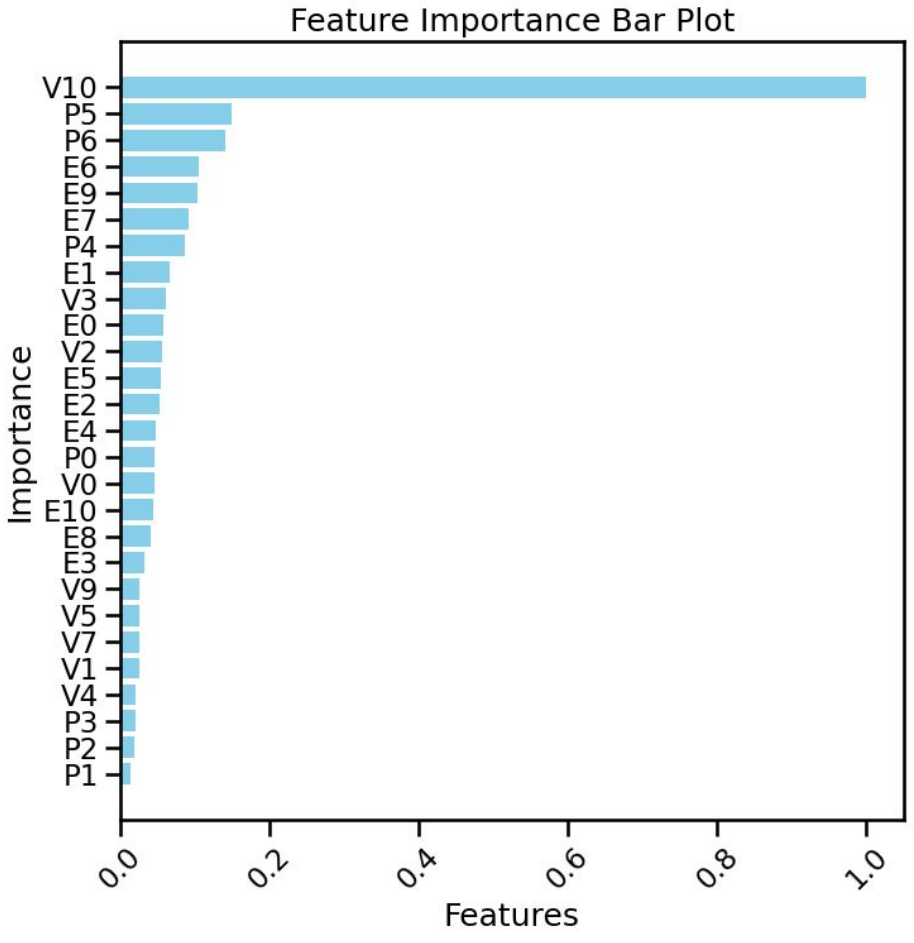



# Feature Importance

Saliency Scores :

- ❏ Computed using probability-based Gradients
- ❏ Normalized relative to maximum value

➔ **Physiological features (P5 and P6) rank among the top three** — reasonable, as distraction affects both mental and physical states.

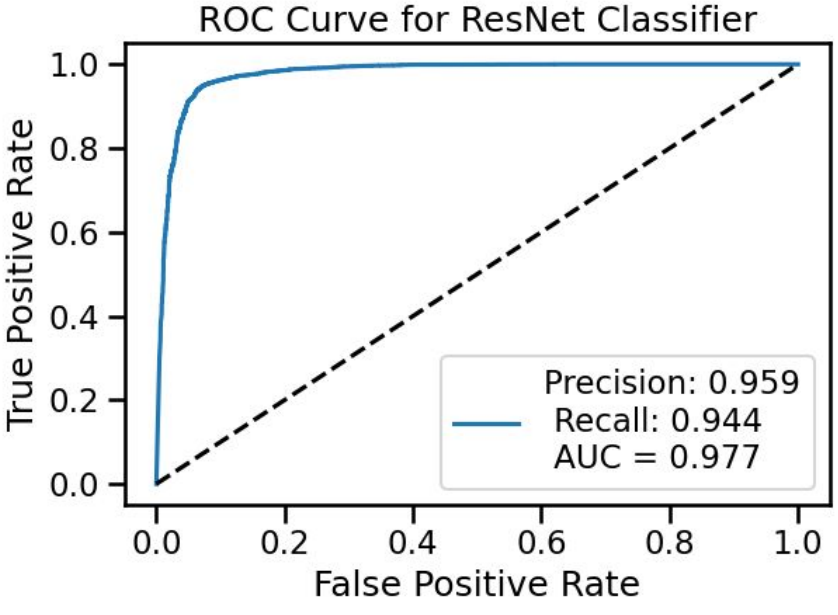


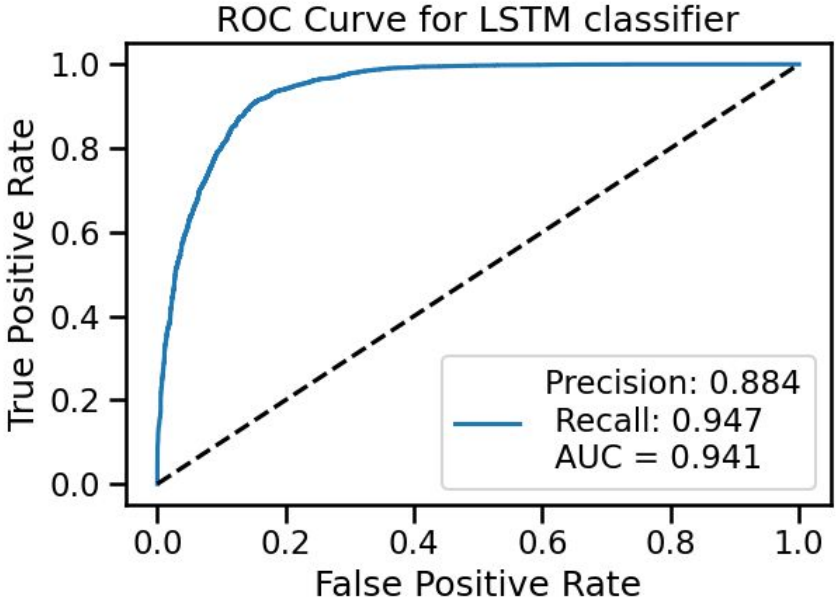
# Evaluation Metrics

- Recall
  - *Safety-critical* metric since missing a distracted driver is costly.
- Precision
  - Higher scores can help reduce *over-alerting*.
- ROC (Receiver Operating Characteristic ) Curve
  - Threshold *dependent* True positive rate vs. False positive rate
- AUC (Area Under the Curve)
  - Threshold *independent* overall performance measure

# Results

## ResNet

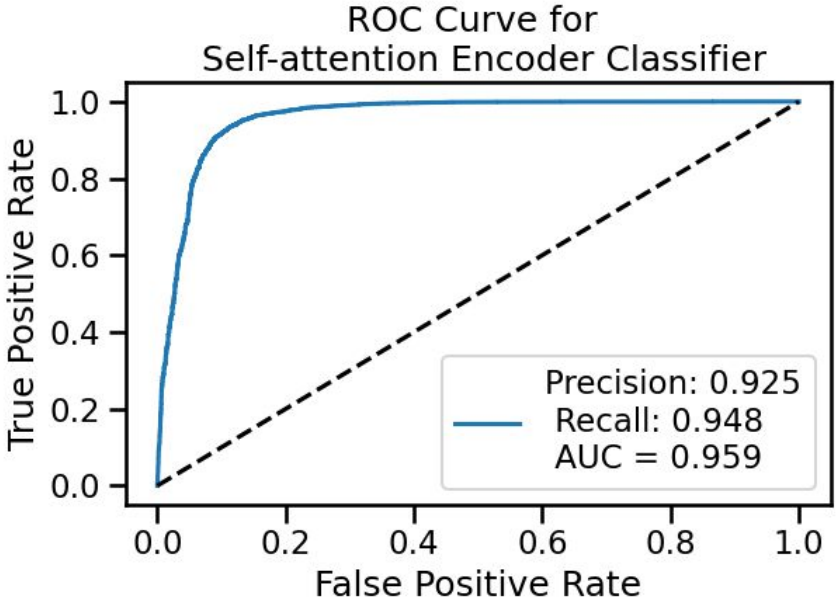


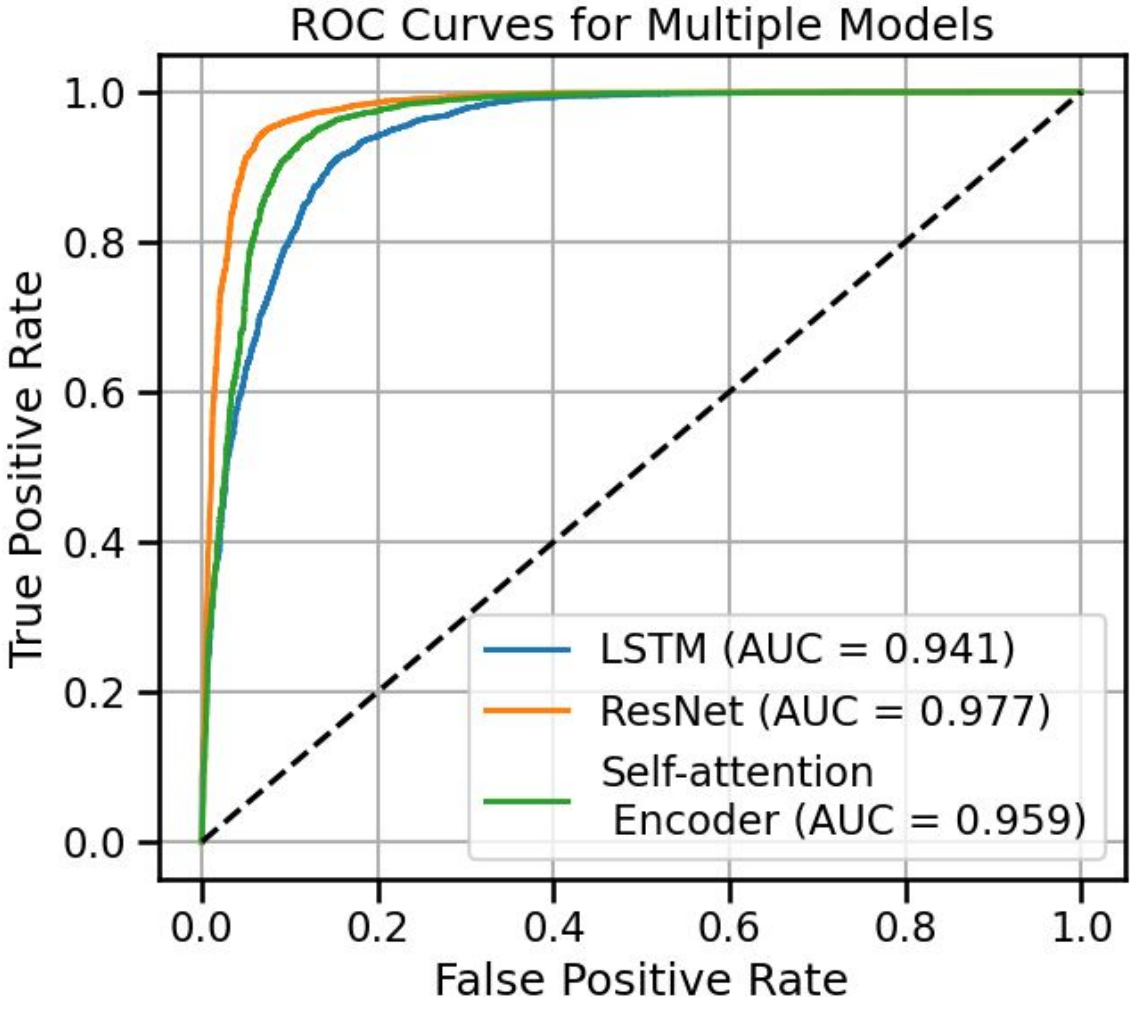

## LSTM

| | Precision | Recall | AUC |
|---|---|---|---|
| **LSTM** | 0.884 | 0.947 | 0.941 |
| **ResNet** | **0.959** | 0.944 | **0.977** |
| **Transformer** | 0.925 | **0.948** | 0.959 |

## Self-attention Encoder



# Comparison of ROC Curve and AUC Across Models



# Interpretation of Results

**Recall** :

**Self-attention encoder** performs best, while other models trail closely.

**Precision :**

**ResNe**t model achieves the highest.

**AUC :**

**ResNet > Self-attention encoder > LSTM**

- **ResNe**t excels at hierarchical temporal feature learning, enabling it to capture detailed temporal patterns.
- **Self-attention encoder** effectively captures long-range dependencies and contextual relationships.

# Summary

- Investigated **distracted driver detection** using **multimodal time series data** combining physiological, environmental, and vehicular signals
- Explored three **deep learning architectures:**
  - **LSTM:** Captures sequential and temporal dependencies in driving behavior
  - **ResNet:** Learns deep hierarchical feature representations from complex sensor patterns
  - **Transformer Encoder:** Utilizes self-attention to model long-range dependencies and inter-feature relationships
- **Performance evaluated** using Precision, Recall, and AUC metrics with ROC curve comparisons
- **Results:** Multimodal deep learning significantly enhances **real-time distracted driver detection** by capturing complementary behavioral cues

# Future Work

- ➢ Incorporate **non-temporal features** (e.g., demographics, driver history, environmental context)
- ➢ Enhance **accuracy, robustness, and generalization** in real-world driving scenarios